Sentiment Analysis for Pain Detection Using Multimodal Data

Thesis submitted by

Anay Ghosh

Doctor of Philosophy (Engineering)

Department of Information Technology

Faculty Council of Engineering & Technology

Jadavpur University

Kolkata, India

2025

JADAVPUR UNIVERSITY

FACULTY OF ENGINEERING AND TECHNOLOGY

INDEX NO. **94/22/5**

1. **Title of the Thesis**: Sentiment Analysis for Pain Detection Using Multimodal Data

2. **Name, Designation and Institution of the Supervisor/s:**

   (a) **Prof. Dr. Bibhas Chandra Dhara**
   Professor, Department of Information Technology
   Jadavpur University, Kolkata –700032

   (b) **Dr. Saiyed Umer**
   Assistant Professor, Department of Computer Science and Engineering
   Aliah University, Kolkata –700156

3. **List of Publications:**

   (a) **Journals**

   J1: Anay Ghosh, Bibhas Chandra Dhara, Chiara Pero and Saiyed Umer, "A multimodal sentiment analysis system for recognizing person aggressiveness in pain based on textual and visual information", *Springer Journal of Ambient Intelligence and Humanized Computing* (SCOPUS, IF: 3.662) [Published].

   J2 Anay Ghosh, Saiyed Umer, Muhammad Khurram Khan, Ranjeet Kumar Rout, Bibhas Chandra Dhara, "Smart sentiment analysis system

for pain detection using cutting edge techniques in a smart healthcare framework", *Springer, Cluster Computing* (SCI, IF: 3.6) [Published].

J3 Anay Ghosh, Saiyed Umer, Bibhas Chandra Dhara, Ranjeet Kumar Rout, "Analyzing deep textual facial patterns for human pain sentiment recognition system in smart healthcare framework", *Intelligent Decision Technologies, IOS Press (Sage Journals)* (SCOPUS, IF: 0.6) [Published].

J4 Anay Ghosh, Saiyed Umer, Bibhas Chandra Dhara, G G Md Nawaz Ali, "A Multimodal Pain Sentiment Analysis System Using Ensembled Deep Learning Approaches for IoT-Enabled Healthcare Framework", *MDPI, Sensors* (SCI, IF: 3.4) [Published].

J5 Anay Ghosh, Saiyed Umer, Bibhas Chandra Dhara, Deepak Kumar Jain, Ranjeet Kumar Rout, and Amir Hussain, "A Novel Pain Sentiment Detection System Utilizing a PainCapsule Model and Textual Facial Patterns", *Neurocomputing, ELSEVIER* (SCI, IF: 6.5) [Published].

J6 Anay Ghosh, Saiyed Umer, and Bibhas Chandra Dhara, "A Multimodal Pain Sentiment Analysis Framework Integrating Text, Audio, Image, and Video Using Advanced Fusion Strategies for Real-Time Healthcare Applications", [Communicated].

4. **List of Presentations in International Conferences and Workshops:**

C1 Anay Ghosh, Saiyed Umer, Bibhas Chandra Dhara, "Pain Sentiment Analysis System Utilizing Deep Learning Frameworks on Multimodal Data for an Effective Healthcare System", ICISS 2024 (SCOPUS), IIEST Shibpur, India, (Presented).

C2 Anay Ghosh, Saiyed Umer, Bibhas Chandra Dhara, "A Method For Diverse Sentiment Analysis Using Textual Datasets", WREC'25 (SCOPUS), NIT Jalandhar, Punjab, India, (Presented).

JADAVPUR UNIVERSITY

FACULTY OF ENGINEERING AND TECHNOLOGY

STATEMENT OF ORIGINALITY

I, **Shri Anay Ghosh** registered on **22nd May, 2022**, do hereby declare that this thesis entitled **"Sentiment Analysis for Pain Detection Using Multimodal Data"** contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All information in this thesis have been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred all materials and results that are not original to this work.

I also declare that I have checked this thesis as per the Policy on Anti Plagiarism, Jadavpur University, 2019, and the level of similarity as checked by iThenticate software is 4%.

Anay Ghosh

Signature of the Candidate :

Date : 14.01.2026

Certified by Supervisors :

(Signature with date, seal)

1. ---------- 14.01.26

PROFESSOR
Deptt. of Information Technology
JADAVPUR UNIVERSITY
Block -LB, Plot-8, Sector-3
Salt Lake, Kolkata-700106, India

(Prof. Dr. Bibhas Chandra Dhara)

2. Saiyed Umer 13.01.2026

Dr. Saiyed Umer
Assistant Professor
Dept. of CSE, Aliah University
Kolkata-700156

(Dr. Saiyed Umer)

JADAVPUR UNIVERSITY

FACULTY OF ENGINEERING AND TECHNOLOGY

CERTIFICATE FROM THE SUPERVISOR/S

This is to certify that the thesis entitled **"Sentiment Analysis for Pain Detection Using Multimodal Data"** submitted by **Shri Anay Ghosh**, who got his name registered on 22nd **May, 2022** for the award of **Ph.D. (Engg.)** degree of Jadavpur University is absolutely based upon his own work under the supervision of **Prof. Dr. Bibhas Chandra Dhara** and **Dr. Saiyed Umer** and that neither his thesis nor any part of the thesis has been submitted for any degree or any other academic award anywhere before.

1. ---------- 14.01.26 ----------

(Prof. Dr. Bibhas Chandra Dhara)
Signature of the Supervisor
and date with Official seal

PROFESSOR
Deptt. of Information Technology
JADAVPUR UNIVERSITY
Block -LB, Plot-8, Sector-3
Salt Lake, Kolkata-700106, India

2. ---------- Saiyed Umer 13.01.2026 ----------

(Prof. Dr. Saiyed Umer)
Signature of the Supervisor
and date with Official seal

Dr. Saiyed Umer
Assistant Professor
Dept. of CSE, Aliah University
Kolkata-700156

Acknowledgements

In the completion of my research, I have been fortunate to have help, support, and encouragement from many people. I would like to acknowledge them for their support. First, I would like to thank my supervisor, Prof. Dr. Bibhas Chandra Dhara, Department of Information Technology, Jadavpur University, and Dr. Saiyed Umer, Department of Computer Science and Engineering, Aliah University, for giving me great intellectual freedom to pursue my topic of interest, for his immense patience and understanding, and for guiding me through every step of the process with knowledge and wisdom. Finally, I would like to thank my family, especially my parents, and other research members. They have encouraged me in every step of my life to be a strong, independent thinker, to work hard, and to keep trying at difficult things. I am eternally grateful to them for all the opportunities they made available to me, and for the support they have given me along the way.

I am grateful to all my co-authors for their contribution in my work.

Anay Ghosh

Anay Ghosh

PhD Fellow, Jadavpur University

Abstract

Sentiment analysis is a vital area of research with applications in various fields such as security, modern healthcare, and business strategy. It plays a key role in addressing serious issues, for example, in helping to identify people with suicidal tendencies, enabling timely intervention and support. It also helps to detect and mitigate the effects of cyberbullying, where individuals can lose social standing due to negative online interaction. In addition, sentiment analysis contributes to improving the quality of products, food, and services by analyzing customer feedback and insights. In today's world, its influence is especially significant in the modern healthcare system, where it supports better patient care and good service.

This thesis explores different innovative approaches to analyze pain sentiments among patients. This study incorporates linguistic and behavioral signals captured from textual, audio, and visual datasets. Here, we have developed text-based pain analysis systems using machine learning approaches as well as deep learning approaches. The machine learning approaches are pillared on feature extraction methods like Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) and classification methods. Deep learning models include Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) architectures to gain deeper insight into the emotional context conveyed in the text.

Text-based systems suffer from some inherent limitations. To overcome these issues, we next design audio-based pain sentiment analysis systems. Vocal signals are rich sources of emotion, picking up subtle changes in pitch, speech rhythm, and instability in the voice that are omitted or hard to detect when written. The feature extraction module of the audio-based system computes statistical features, Mel-frequency cepstral coefficients (MFCCs), and spectral features. Each feature set is initially

classified using traditional machine learning models; subsequently, the features are concatenated, and a deep learning architecture with a fully connected network is implemented for classification.

Despite the significant benefits provided by audio-based pain analysis, it also poses serious challenges, especially in terms of ambiguity in data from various aspects such as due to tone, language barriers, and cultural differences. This fundamental issue prompted us to incorporate facial expression analysis as a complementary modality because facial movements are governed by consistent neuromuscular patterns and are difficult to consciously suppress, offering a more stable reflection of true pain labels. To resolve the issue, we developed image-based systems. From the facial image, features like the Local Binary Pattern (LBP) and the Histogram of Oriented Gradients (HOG) are derived and then classified. On the other hand, we designed four CNN deep architectures.

Image-based analysis has its own limitations, primarily its static nature, which prevents the assessment of temporal dynamics such as the duration and progression of pain expressions. To overcome this limitation, we extend the framework by incorporating video, a dynamic signal. The image-based system using the deep learning approach has much superior performance in comparison to the machine learning approach, so a video-based pain analysis system is designed following the deep architecture only. Here, features are derived using the two well-performing CNN models along with five well-known pretrained models. For classification, we propose two architectures, *PainCapsule* and *AttentionPainCapsule*.

The individual system based on different types of data cannot achieve sufficient accuracy. Finally, we design multimodal systems to fuse different systems. In this work, we apply the post-classification fusion technique and achieve significant improvement.

Contents

Abstract i

Table of Contents iii

List of Figures vi

List of Tables ix

Abbreviations xi

List of Publications xv

1 Introduction 1
- 1.1 Characteristics of Sentiment Analysis . . . 4
- 1.2 Types of Sentiment Analysis . . . 6
- 1.3 Pain Sentiment Analysis . . . 12
  - 1.3.1 Text-Based Pain Analysis . . . 14
  - 1.3.2 Audio-Based Pain Analysis . . . 15
  - 1.3.3 Image-Based Pain Analysis . . . 16
  - 1.3.4 Video-Based Pain Analysis . . . 17
  - 1.3.5 Multimodal-Based Pain Analysis . . . 18
  - 1.3.6 Hand-Crafted Features and Classifiers . . . 19
    - 1.3.6.1 Hand-Crafted Feature Representations . . . 19
    - 1.3.6.2 Machine Learning Classifiers . . . 23
  - 1.3.7 Deep Learning Features and Classifiers . . . 26
- 1.4 Challenges of the Thesis . . . 31
- 1.5 Objectives of the Thesis . . . 34
- 1.6 Contributions of the Thesis . . . 35
- 1.7 Experimental Setup . . . 37
  - 1.7.1 Performance Measure . . . 38
- 1.8 Organization of the Thesis . . . 40

2 Text-Based Pain Sentiment Analysis **42**
2.1 Literature Review . . . 44
2.2 Proposed PSA_{text} Systems . . . 47
2.2.1 Text Preprocessing . . . 47
2.2.2 $HANDPSA_{text}$ System . . . 49
2.2.2.1 Feature Extraction . . . 50
2.2.2.2 Classification . . . 54
2.2.3 $DLPSA_{text}$ Systems . . . 54
2.3 Experiments and Results . . . 58
2.4 Conclusions . . . 67

3 Audio-Based Pain Sentiment Analysis **69**
3.1 Literature Review . . . 70
3.2 Proposed PSA_{audio} Systems . . . 73
3.2.1 Audio Preprocessing . . . 74
3.2.2 $HANDPSA_{audio}$ System . . . 76
3.2.3 Classification . . . 79
3.2.4 $DLPSA_{audio}$ System . . . 79
3.3 Experiments and Results . . . 81
3.4 Conclusions . . . 88

4 Image-Based Pain Sentiment Analysis **90**
4.1 Literature Review . . . 92
4.2 Proposed PSA_{image} Systems . . . 94
4.2.1 Image Preprocessing . . . 94
4.2.2 $HANDPSA_{image}$ System . . . 97
4.2.3 Classification . . . 102
4.2.4 $DLPSA_{image}$ System . . . 102
4.2.5 Proposed CNN_1 System . . . 103
4.2.6 Proposed CNN_2 System . . . 105
4.2.7 Proposed CNN_3 System . . . 107
4.2.8 Proposed CNN_4 System . . . 110
4.3 Experiments and Results . . . 111
4.4 Conclusions . . . 124

5 Video-Based Pain Sentiment Analysis **126**
5.1 Literature Review . . . 127
5.2 Proposed PSA_{video} Systems . . . 130
5.2.1 Video preprocessing . . . 131
5.2.2 $DLPSA_{video}$ System . . . 135
5.2.2.1 Proposed *PainCapsule* model . . . 138
5.2.2.2 Proposed *PainAttentionCapsule* model . . . 139

5.3 Experiments and Results . . . 141
5.4 Conclusions . . . 152

6 Multimodal-Based Pain Sentiment Analysis **153**
6.1 Literature Review . . . 156
6.2 MPSA Systems . . . 158
6.2.1 Data Synchronization . . . 159
6.2.2 Modalities Fusion . . . 161
6.3 Experiments and Results . . . 166
6.3.1 MPSA$_{TA}$ System . . . 166
6.3.2 MPSA$_{AV}$ system . . . 167
6.3.3 MPSA$_{TAV}$ System . . . 170
6.4 Conclusions . . . 172

7 Conclusions and Future Scope **174**
7.1 Future Research Directions . . . 176

Bibliography **177**

List of Figures

1.1 Polarity of the Sentiment Analysis System. 1
1.2 Flow diagram of a Sentiment Analysis System. 2
1.3 Characteristics and components of SAS. 5
1.4 Block diagram of a Pain Sentiment Analysis System. 13

2.1 Workflow diagram of proposed PSA_{text} System. 43
2.2 Steps with result of the text preprocessing of PSA_{text} System. 50
2.3 Block diagram of BoW based feature extraction method. 51
2.4 Block diagram to compute TF-IDF features. 53
2.5 Block diagram of LSTM based $DLPSA_{text}$ System ($DLPSA_{textlstm}$ System). 55
2.6 Block diagram of BiLSTM based $DLPSA_{text}$ System ($DLPSA_{textbilstm}$ System). 57
2.7 Some samples of TD_{aggr} (3-Class Problem). 57
2.8 Some samples of TD_{amazon} (5-Class Problem). 59
2.9 Some samples of TD_{llm} (5-Class Problem). 60
2.10 The confusion matrix in percentage using $HANDPSA_{text}$ System and $DLPSA_{text}$ System with TD_{aggr} dataset for 3-class classification. . . . 64
2.11 The confusion matrix in percentage using $HANDPSA_{text}$ System and $DLPSA_{text}$ System with TD_{amazon} for 5-class classification. 65
2.12 The confusion matrix in percentage using $HANDPSA_{text}$ System and $DLPSA_{text}$ System with TD_{llm} for 5-class classification. 66
2.13 Effect of batch vs. epoch for the proposed BiLSTM Sentiment Analysis system for 3-class (with TD_{aggr}) classification. 67

3.1 Demonstration of proposed PSA_{audio} Systems. 74
3.2 Steps of audio preprocessing. 76
3.3 Proposed $DLPSA_{audio}$ System. 80
3.4 Some sample audio signals of AD_{VIVAE} (3-Class). 81
3.5 Some samples of $AD_{RAVDESS}$ (5-Class). 83
3.6 The performance of the proposed $DLPSA_{audio}$ System with (a) 50-50% training-testing, and (b) 75-25% training-testing sets for 3-class classification problem. 85

3.7 Comparison of accuracy of the proposed $\text{HANDPSA}_{\text{audio}}$ System and $\text{DLPSA}_{\text{audio}}$ System on AD_{VIVAE} dataset. . . . 86
3.8 Comparison of accuracy of the proposed $\text{HANDPSA}_{\text{audio}}$ System and $\text{DLPSA}_{\text{audio}}$ System on $\text{AD}_{\text{RAVDESS}}$ dataset. . . . 86
3.9 Confusion matrices in percentage for $\text{HANDPSA}_{\text{audio}}$ System and $\text{DLPSA}_{\text{audio}}$ System on 3-class classification problem for AD_{VIVAE} dataset. . . . 87
3.10 Confusion matrices in percentage for $\text{HANDPSA}_{\text{audio}}$ System and $\text{DLPSA}_{\text{audio}}$ System on 5-class classification problem for $\text{AD}_{\text{RAVDESS}}$ dataset. . . . 87

4.1 Block diagram of the proposed $\text{PSA}_{\text{image}}$ Systems. . . . 92
4.2 Block diagram of TSPM. . . . 95
4.3 Results of image preprocessing of $\text{PSA}_{\text{image}}$ System. . . . 98
4.4 An example of feature extraction using the LBP technique. . . . 99
4.5 An example of feature extraction using the HOG technique. . . . 100
4.6 Statistical-based approach for feature representation from the facial region F. . . . 101
4.7 Proposed CNN_1 architecture. . . . 103
4.8 Proposed CNN_2 architecture. . . . 105
4.9 Proposed CNN_3 architecture. . . . 107
4.10 Proposed CNN_4 architecture. . . . 108
4.11 Some sample images of ID_{UNBC} (3-Class). . . . 113
4.12 Sample video frames from $\text{VD}_{\text{BioVid}}$ (5-Class). . . . 114
4.13 Performance of $\text{HANDPSA}_{\text{image}}$ System with SVM as classifier on ID_{UNBC} 2-class pain detection. . . . 116
4.14 Effectiveness of batch vs epochs for the proposed $\text{DLPSA}_{\text{image}}$ System on ID_{UNBC}. . . . 116
4.15 Confusion matrices in percentage for the proposed $\text{HANDPSA}_{\text{image}}$ System and $\text{DLPSA}_{\text{image}}$ System on ID_{UNBC}. . . . 122
4.16 Confusion matrices in percentage for the proposed $\text{HANDPSA}_{\text{image}}$ System and $\text{DLPSA}_{\text{image}}$ System on $\text{VD}_{\text{BioVid}}$. . . . 122

5.1 Block Diagram of the proposed $\text{PSA}_{\text{video}}$ system. . . . 127
5.2 Block Diagram of the proposed $\text{PSA}_{\text{video}}$ System using *PainCapsule* method. . . . 132
5.3 Detailed analysis of face global motion before facial expression detection for $subject_1$ of $\text{VD}_{\text{BioVid}}$. . . . 133
5.4 Detailed analysis of face global motion after facial expression detection for $subject_1$ of $\text{VD}_{\text{BioVid}}$. . . . 133
5.5 Detailed analysis of face global motion before facial expression detection for $subject_2$ of $\text{VD}_{\text{BioVid}}$. . . . 134

5.6 Detailed analysis of face global motion after facial expression detection for $subject_2$ of VD_{BioVid}. 135
5.7 Results of frame preprocessing of PSA_{video} System. 136
5.8 Proposed *PainCapsule* architecture for $DLPSA_{video}$ System. 139
5.9 Proposed *PainAttentionCapsule* architecture for $DLPSA_{video}$ System. 140
5.10 Sample video of $VD_{MIntPAIN}$ (5-Class). 142
5.11 Sample video frames from VD_{BioVid} (3-Class). 143
5.12 Sample video frames from $VD_{MIntPAIN}$ (3-Class). 143
5.13 Performance for $DLPSA_{video}$ system using pretrained models for VD_{BioVid} (Approach-1). 144
5.14 Performance for $DLPSA_{video}$ System using end to end CNN models for VD_{BioVid} (Approach-2). 145
5.15 Performance for $DLPSA_{video}$ System using end to end CNN models for $VD_{MIntPAIN}$ (Approach-2). 146
5.16 Demonstration of confusion matrix in percentage as the performance measure for the $DLPSA_{video}$ system for 3-Class problem. 148
5.17 Demonstration of confusion matrix in percentage as the performance measure for the $DLPSA_{video}$ system for 5-Class problem. 150

6.1 A block diagram of the proposed MPSA system. 155
6.2 Data synchronization model employed in this work. 160
6.3 Block Diagram of the proposed $MPSA_{TA}$ system. 164
6.4 Block Diagram of the proposed $MPSA_{AV}$ system. 165
6.5 Block Diagram of the proposed $MPSA_{TAV}$ system. 166
6.6 Illustration of 3-class and 5-class confusion matrices in percentage for the $MPSA_{TA}$ system using WSSLF technique. 168
6.7 Illustration of 3-class and 5-class confusion matrices in percentage for the $MPSA_{AV}$ system using WSSLF technique. 170
6.8 Illustration of 3-class and 5-class confusion matrices in percentage for the $MPSA_{TAV}$ system using WSSLF technique. 172

List of Tables

2.1 Parameter details of the used LSTM architecture. . . . 56
2.2 Parameter details of the used BiLSTM architecture. . . . 56
2.3 Description of TD_{aggr} Samples. . . . 58
2.4 Detailed Description of TD_{amazon}. . . . 59
2.5 Detailed Description of TD_{llm}. . . . 60
2.6 Feature size of the datasets TD_{aggr}, TD_{amazon}, and TD_{llm}. . . . 60
2.7 Performance of the proposed $HANDPSA_{text}$ System with f_{bow} and $f_{tf\text{-}idf}$ features. . . . 62
2.8 Performance of the proposed system using $DLPSA_{text}$ systems along with training and testing times in *Sec.* . . . 63
2.9 Comparison of Performance of the proposed system with the other competing methods for the 3-class (with TD_{aggr}) and 5-class (with TD_{amazon}) classification problems. . . . 67

3.1 List of parameters for $DLPSA_{audio}$ System. . . . 81
3.2 Number of sample in each class of AD_{VIVAE} (3-Class). . . . 82
3.3 Number of sample in each class of $AD_{RAVDESS}$ (5-Class). . . . 83
3.4 The performance the proposed $HANDPSA_{audio}$ System on AD_{VIVAE} using different classifiers. . . . 84
3.5 The performance of $HANDPSA_{audio}$ System on $AD_{RAVDESS}$ using different classifiers. . . . 84
3.6 Performance of the proposed $DLPSA_{audio}$ System on AD_{VIVAE} and $AD_{RAVDESS}$. . . . 85
3.7 Comparison of the performance of the proposed systems with the SoA methods. . . . 88

4.1 Description of the CNN_1 architecture. . . . 104
4.2 Description of the CNN_2 architecture. . . . 106
4.3 Parameters list of CNN_3 architecture. . . . 109
4.4 Parameters list of CNN_4 architecture. . . . 110
4.5 Comparison of the CNN architectures as models for the $DLPSA_{image}$ system. . . . 112
4.6 Description of the employed ID_{UNBC}. . . . 112

4.7 Dataset description of both VD_{BioVid} for Approach-1, Approach-2 and Approach-3 . . . 113
4.8 Feature dimension using different feature partitioning setup for $HANDPSA_{image}$ system. . . . 114
4.9 Performance of $HANDPSA_{image}$ System for 2-class pain detection on $ID_{\mathbf{UNBC}}$ using P_1. . . . 115
4.10 Performance of the proposed $HANDPSA_{image}$ System on ID_{UNBC} (3-class) and VD_{BioVid} (5-class). . . . 116
4.11 Performance of the proposed $DLPSA_{image}$ System using CNN_1. . . . 117
4.12 Performance of the proposed $DLPSA_{image}$ System using CNN_2. . . . 118
4.13 Performance of the proposed $DLPSA_{image}$ System using CNN_3. . . . 118
4.14 Performance of the proposed $DLPSA_{image}$ System using CNN_4. . . . 119
4.15 Summary of the performance of different schemes of the proposed $DLPSA_{image}$ System. . . . 120
4.16 Performance comparison for the proposed PSA_{image} System of this Chapter. . . . 123

5.1 List of parameters required of the Proposed *PainCapsule* framework. 139
5.2 Dataset description of both VD_{BioVid} and $VD_{MIntPAIN}$ for Approach-1, Approach-2 and Approach-3. . . . 141
5.3 Performance of the $DLPSA_{video}$ system using VD_{BioVid} with different approaches. . . . 147
5.4 Performance of the $DLPSA_{video}$ system using $VD_{MIntPAIN}$ with different approaches. . . . 147
5.5 The LSTM architecture summarization and its parameters for the $DLPSA_{video}$ System. . . . 148
5.6 Performance of the LSTM model using VD_{BioVid} and $VD_{MIntPAIN}$ for 3-class problem. . . . 149
5.7 Performance of the LSTM model using VD_{BioVid} and $VD_{MIntPAIN}$ for 5-class problem. . . . 150
5.8 Performance comparison of the proposed $DLPSA_{video}$ system of this chapter. . . . 151

6.1 List of best prediction model considered for the MPSA system. . . . 161
6.2 Types of MPSA systems. . . . 163
6.3 Performance of $MPSA_{TA}$ System. . . . 167
6.4 Performance of $MPSA_{AV}$ System.. . . . 169
6.5 Performance of $MPSA_{TAV}$ System. . . . 171

Abbreviations

AAM	Active Appearance Models
AC	Audio Class
AD	Audio Dataset
AFEW	Acted Facial Expressions In The Wild
AI	Artificial Intelligence
AM	Attention Model
ANN	Artificial Neural Network
ASM	Active Shape Models
AV	Audio-Video
AUs	Action Units
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory
BioVid	Biopotential and Video Heat Pain Database
CAG	Covertly Aggressive
CK+	Extended Cohn-Kanade
CLIP	Contrastive Language–Image Pre-training
CNN	Convolutional Neural Network
ComParE	Computational Paralinguistics ChallengE
CRFs	Conditional Random Fields

DANs	Deep Attention Networks
DCT	Discrete Cosine Transform
DenseNet	Densely Connected Convolutional Networks
DFT	Discrete Fourier Transform
DL	Deep Learning
DLF	Decision-level fusion
DLPSA	Deep Learning based Pain Sentiment Analysis
DNNs	Deep Neural Networks
DT	Decision Tree
DTW	Dynamic Time Warping
EHRs	Electronic Health Records
FACS	Facial Action Coding System
FCNs	Fully Connected Networks
FFT	Fast Fourier transformation
FN	False Negative
FP	False Positive
GNNs	Graph neural networks
GPT	Generative Pre-trained Transformers
GRUs	Gated Recurrent Units
HANDPSA	Hand Crafted Pain Sentiment Analysis
HOG	Histogram of Oriented Gradients
IC	Image Class
ID	Image Dataset
IEMOCAP	Interactive emotional dyadic motion capture
kNN	k-Nearest Neighbors
LBP	Local Binary Pattern
LLM	Large Language Model

LR	Logistic Regression
LSTM	Long Short-Term Memory
MFCCs	Mel-Frequency Cepstral Coefficients
MIntPAIN	Multimodal Intensity Pain
ML	Machine Learning
MLP	Multi-Layer Perceptron
MPSA	Multimodal-Based Pain Sentiment Analysis
MVC	Modified Video Class
NAG	Non-Aggressive
NLP	Natural Language Processing
NMS	Non-Maximum Suppression
NN	Neural Network
OAG	Overtly Aggressive
OpenCV	Open Source Computer Vision Library
ORS	Observer Rating Scale
PCA	Principal Component Analysis
PI	Pain Intensity
PSAS	Pain Sentiment Analysis System
PSLF	Product Score Level Fusion
RAVDESS	Ryerson Audio-Visual Database of Emotional Speech and Song
ReLU	Rectified Linear Unit
ResNet	Residual Neural Network
RF	Random Forest
RMS	Root-Mean-Square
RNN	Recurrent Neural Network
SA	Sentiment Analysis
SAS	Sentiment Analysis System

SIFT	Scale-Invariant Feature Transform
SoA	State-of-the-Art
SS	Sensory Scale
SSLF	Sum Score Level Fusion
SVM	Support Vector Machine
TA	Text-Audio
TAV	Text-Audio-Video
TC	Text Class
TD	Text Dataset
TF-IDF	Term Frequency-Inverse Document Frequency
TN	True Negative
TP	True Positive
TSPM	Tree Structured Part Model
UNBC	University of Northern British Columbia
VAS	Visual Analog Scale
VC	Video Class
VD	Video Dataset
VGG	Visual Geometry Group
VIVAE	Variably Intense Vocalizations of Affect and Emotion Corpus
WSSLF	Weighted Sum Product Score Level Fusion
X-ITE	Experimentally Induced Thermal and Electrical Pain Database

List of Publications

List of Journal Contributing to This Thesis

1. **J1** Anay Ghosh, Bibhas Chandra Dhara, Chiara Pero and Saiyed Umer, "A multimodal sentiment analysis system for recognizing person aggressiveness in pain based on textual and visual information", Springer Journal of Ambient Intelligence and Humanized Computing (SCOPUS, IF: 3.662) [Published].

2. **J2** Anay Ghosh, Saiyed Umer, Muhammad Khurram Khan, Ranjeet Kumar Rout, Bibhas Chandra Dhara, "Smart sentiment analysis system for pain detection using cutting edge techniques in a smart healthcare framework", Springer, Cluster Computing (SCI, IF: 3.6) [Published].

3. **J3** Anay Ghosh, Saiyed Umer, Bibhas Chandra Dhara, Ranjeet Kumar Rout, "Analyzing deep textual facial patterns for human pain sentiment recognition system in smart healthcare framework", Intelligent Decision Technologies, IOS Press (Sage Journals) (SCOPUS, IF: 0.6) [Published].

4. **J4** Anay Ghosh, Saiyed Umer, Bibhas Chandra Dhara, G G Md Nawaz Ali, "A Multimodal Pain Sentiment Analysis System Using Ensembled Deep Learning Approaches for IoT-Enabled Healthcare Framework", MDPI, Sensors, (SCI, IF: 3.4) [Published].

5. **J5** Anay Ghosh, Saiyed Umer, Bibhas Chandra Dhara, Deepak Kumar Jain, Ranjeet Kumar Rout, "Exploring Pain Sentiment Detection System Through Implementation of PainCapsule Model By Analyzing Textual Facial Patterns", Expert Systems, (Under review).

6. **J6** Anay Ghosh, Saiyed Umer, and Bibhas Chandra Dhara, "A Multimodal Pain Sentiment Analysis Framework Integrating Text, Audio, Image, and Video Using Advanced Fusion Strategies for Real-Time Healthcare Applications", [Communicated].

List of Conference Contributing to This Thesis

1. **C1** Anay Ghosh, Saiyed Umer, Bibhas Chandra Dhara, "Pain Sentiment Analysis System Utilizing Deep Learning Frameworks on Multimodal Data for an Effective Healthcare System", ICISS 2024 (SCOPUS), IIEST Shibpur, India, (Presented).

2. **C2** Anay Ghosh, Saiyed Umer, Bibhas Chandra Dhara, "A Method For Diverse Sentiment Analysis Using Textual Datasets", WREC'25 (SCOPUS), NIT Jalandhar, Punjab, India, (Presented).

Chapter 1

Introduction

Sentiment refers to the emotional tone or attitude conveyed through text, speech, behavior, or other forms of expression [1]. Sentiment analysis (SA) is an application of the interdisciplinary domains [2], drawing from linguistics, computer science, and machine learning to detect, extract, quantify, and analyze emotions and subjective content (negative, neutral, positive) from various sources [3]. The primary objective of sentiment analysis systems [4] is to determine the polarity of a given input, specifically whether the expressed sentiment is positive, negative, or neutral. More sophisticated systems can identify specific emotions, levels of urgency, and underlying intentions. The polarity of sentiment analysis is illustrated in Fig. 1.1.

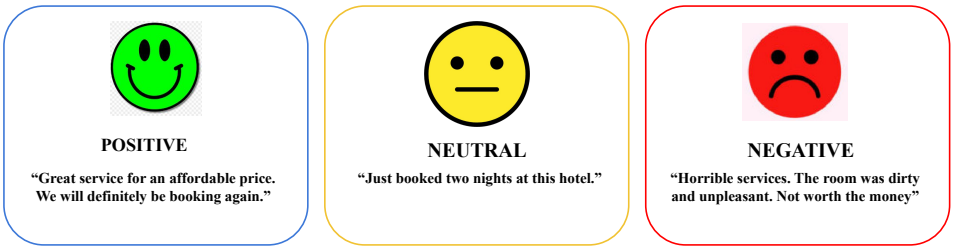


FIGURE 1.1: Polarity of the Sentiment Analysis System.

A sentiment analysis system [5] is a computerized system that discovers, examines, and comprehends feelings and opinions from various types of data (such as text, audio, or video). A sentiment analysis system uses computational methods [6] to automatically identify and interpret subjective data from text [3], audio, image, and video. The system examines sentiments using different techniques that categorize

sentiments as positive, negative, or neutral and further categorize them into more complex emotional states, such as joy, pain, anger, or surprise [4]. The process starts with data preprocessing and feature extraction to determine the elements influencing the sentiment. These comprehensive feature extraction strategies support real-time emotion monitoring in domains such as healthcare, customer sentiment analysis, and mental health assessment while addressing challenges related to contextual understanding, multimodal data integration, and bias mitigation. Advanced systems of modern days use deep learning models that can understand the complex relationships within different types of data. This helps them to make better sense of complex patterns and subtle signals, leading to more accurate and meaningful results [7]. Fig. 1.2 presents the working principle of a digital sentiment analysis system.

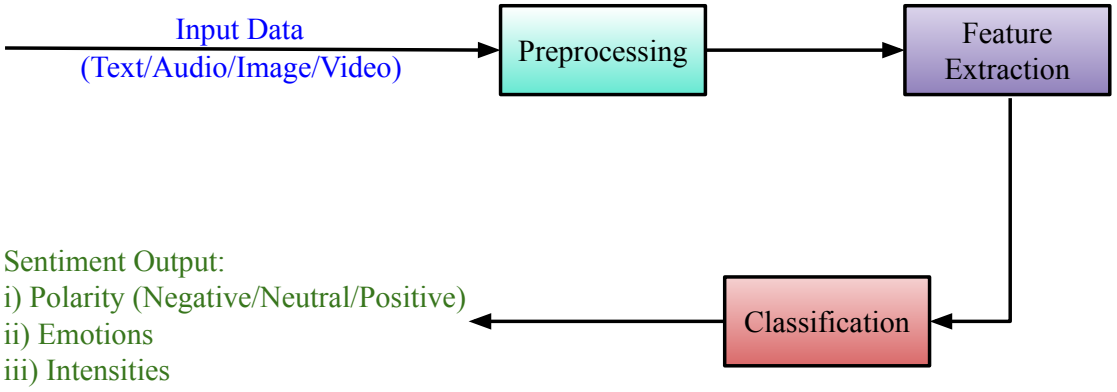


FIGURE 1.2: Flow diagram of a Sentiment Analysis System.

The system has four different parts: data acquisition (or data input), preprocessing data (to remove noise and unwanted data), feature extraction, and classification. In this thesis, for experimental purposes, we have used some standard datasets. So, there is no model for data acquisition. We briefly discuss the other modulus of the system in the following.

- **Data preprocessing:** Four types of data (as shown in Fig. 1.2) are commonly used in sentiment analysis. The preprocessing techniques for each of these modalities differ due to the distinct structural and contextual characteristics inherent to each data type. The text data preprocessing involves removing non-informative words such as stop words, articles, prepositions, pronouns,

and be-verbs. This is followed by tokenization, where the text is broken down into individual tokens or words, which are considered preprocessed text data. In audio-based data preprocessing, signal processing techniques are employed to digitize the audio signals and remove noise from these digitized signals using methods such as the Kalman Filter and wavelet denoising. The image preprocessing techniques involve filtering, noise reduction, and region-of-interest segmentation using image processing methods such as Gaussian filtering, median filtering, histogram equalization, edge detection, and thresholding to prepare images for subsequent feature extraction tasks. The video preprocessing tasks involve frame-by-frame decomposition, followed by the application of image preprocessing techniques such as filtering, noise removal, and region segmentation to enable effective video analysis.

- **Feature extraction:** Feature extraction techniques transform preprocessed data into quantifiable data, which is called a feature of the data. This feature enables the identification of emotional states such as determining whether an emotion is positive, negative, or neutral, classifying specific emotional categories like joy or anger, or estimating emotional intensity by analyzing digital signals such as word patterns in text, voice signals in speech, and micro-expressions in image/video [8]. These representations are obtained using methodologies tailored to each data modality. For instance, text feature extraction utilizes techniques such as Bag-of-Words, Term Frequency Inverse Document Frequency, and Context-Aware Embeddings, e.g., Word2Vec [9] and BERT [10]. Audio features are extracted using methods like Statistical Features, Mel-Frequency Cepstral Coefficients (MFCCs) [11], and Spectral Features [12]. Image feature extraction includes handcrafted descriptors such as Histogram of Oriented Gradients (HOG) [13], Local Binary Patterns (LBP), Scale-Invariant Feature Transform (SIFT) [14], Bag-of-Words, and Sparse Representations, as well as deep learning models like VGG16 [15], ResNet50 [16], Inception-v3 [17], and MobileNet [18]. Video analysis employs the above image feature extraction methods on a frame-by-frame basis and further leverages spatiotemporal models, such as 3D CNNs [19] and deep convolutional architectures, for large-scale video classification [20].

- **Classification:** Once features are extracted from the preprocessed data, they are further utilized for classification tasks. The classification task is performed using a range of machine learning and deep learning methods. Traditional machine learning algorithms, especially those based on supervised learning, use labeled datasets to train models that automatically recognize sentiment or emotion patterns [21]. Common classifiers include Random Forest [22], Support Vector Machine (SVM) [23], Logistic Regression [24], Decision Tree [25], k-Nearest Neighbor [26] and Classification and Regression Tree [27]. In recent years, deep learning has significantly advanced the field, with architectures such as Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNNs) [28, 29, 30], Convolutional Neural Networks (CNNs), Bidirectional LSTM (BiLSTM), Gated Recurrent Units (GRUs), Deep Neural Networks (DNNs), and Fully Connected Networks (FCNs). Furthermore, the emergence of transformer-based architectures has revolutionized sentiment and emotion classification tasks. Models such as Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformers (GPT) provide a deeper and more nuanced understanding of language and context, enabling improved performance in a variety of natural language processing tasks [10]. Despite these advancements, several persistent challenges remain in building robust sentiment analysis systems [1]. One major limitation is the difficulty in interpreting figurative language, including sarcasm, irony, and humor, which often requires an understanding of subtle contextual cues and world knowledge that current models struggle to emulate [31]. These challenges underscore the need for more context-aware, multimodal, and semantically enriched models in the ongoing evolution of affective computing.

1.1 Characteristics of Sentiment Analysis

Sentiment analysis is the computational process of extracting, identifying, and classifying opinions, emotions, and attitudes from text, speech, image, and video data. Its distinguishing characteristics are the polarity classification (positive, negative, neutral), subjectivity detection (opinion or fact), emotion granularity (joy, anger, pain), intensity measurement (strong or weak sentiment), and context awareness

(handling sarcasm, slang, or domain-specific language). Sentiment Analysis System (SAS) is used in customer feedback analysis, social media tracking, and market research to gain actionable insights from unstructured information. Sentiment analysis (SA) focuses on understanding opinions expressed by individuals about various topics such as events, issues, aggression, anger, and general attitudes. However, recent advances have taken this one step further. Today, researchers are identifying the polarity of sentiment and quantifying the degree of positivity or negativity using sentiment scores. With the explosive growth of user activity on online platforms, interactions among individuals have increased dramatically. Unfortunately, this increase in digital engagement has also led to a surge in aggression-driven behaviors such as flaming, trolling, roasting, and cyberbullying on a global scale [32]. Sentiment analysis extends to several crucial areas, including spam detection, identifying bullying or suicidal tendencies, tracking fraudulent messages, recognizing aggression, and monitoring other harmful behaviors. Although SA is based on a single data type, it may be useful in many scenarios; however, there are situations where relying on just one modality may not provide a significant level of understanding. These limitations are discussed in more detail in the following subsections. The characteristics and components of sentiment analysis are illustrated in Fig. 1.3.

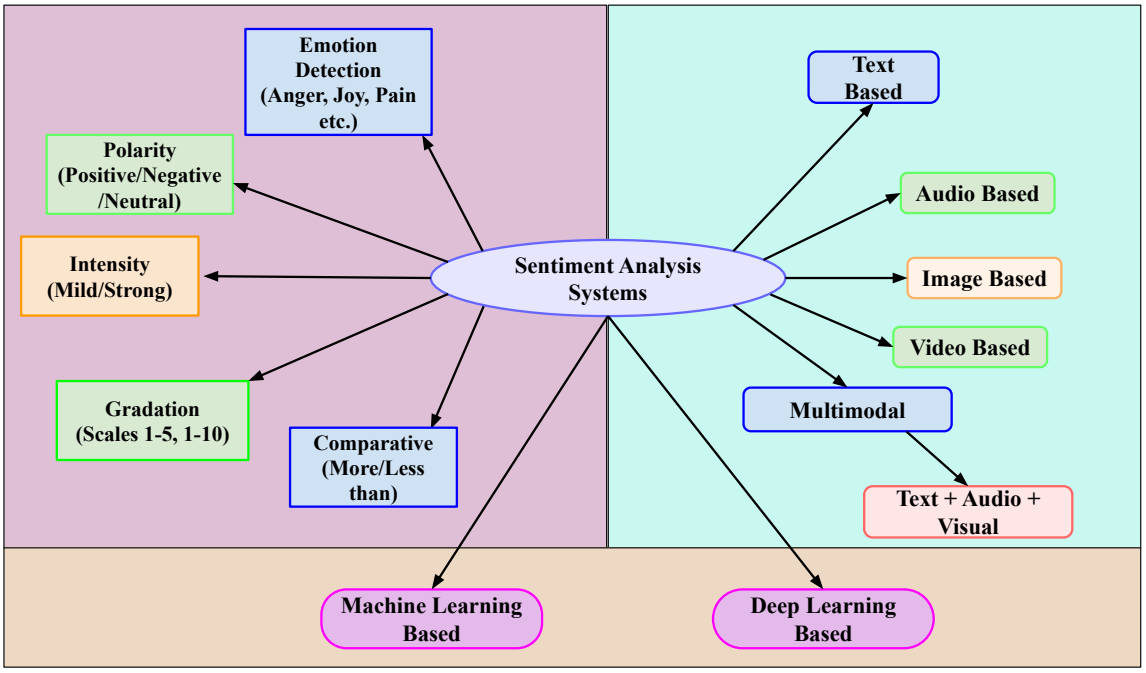


FIGURE 1.3: Characteristics and components of SAS.

1.2 Types of Sentiment Analysis

Sentiment analysis systems are classically divided into various types depending on their point of analytical focus and methodology [3]. Digital sentiments come from various sources, depending on the purpose of the data. Primary sources are social media sites (Twitter tweets, Facebook posts), product reviews (Amazon, Yelp), customer feedback forums, news stories, and forum posts [2]. These signals appear as structured data (star ratings, likes), semi-structured data (hashtags, emoticons), and unstructured free text. New sources include voice assistant interactions, chatbot dialogues, and video/audio transcripts. Digital sentiment signals can be classified into various categories. Lexical signals consist of explicit sentiment terms (e.g., 'excellent', 'terrible'), intensifiers ('very', 'extremely'), and negations [33]. Para-linguistic signals include emoticons, emojis, and punctuation usage patterns (e.g., a string of exclamation marks). Structural signals include review ratings, upvote/downvote ratios, and social media engagement metrics. Contextual signals include temporal patterns (sentiment evolution over time), author features, and domain-specific terms. Every type of signal needs specific processing methods to accurately determine its contribution to sentiment [34].

A sentiment analysis system can be attributed to sentiment categories such as polarity, intensity, etc. Different attributes of a sentiment analysis are shown in the upper left part of Fig. 1.3. A sentiment analysis system can also be classified according to the type of data used to recognize sentiment. The upper right part of Fig. 1.3 shows different types of systems. Further, sentiment analysis systems are categorized as machine learning-based methods or deep learning based methods on the basis of the tools used.

Attributes of Sentiment Analysis: Emotion detection is about recognizing and labeling feelings such as happiness, anger, or sadness from things people say, write, or show through expressions and voice. A visual representation of the SA attributes is illustrated in Fig. 1.3.

1. **Emotion Detection:** This is about recognizing human feelings, like joy, anger, or surprise, from things we say, write, or show in photos and videos. It uses smart machine learning tools to spot emotional patterns. Unlike simple

sentiment analysis, this digs deeper and can tell the difference between more subtle feelings.

2. **Polarity:** Polarity tells us whether something feels positive, negative, or neutral. It is the most basic way to understand the mood or opinion of someone. Most systems start with this step before doing anything more complex.

3. **Intensity:** This looks at how strong the feeling is, whether it is just a bit of annoyance or full-blown rage. It helps to distinguish emotions that may sound similar but have different impacts. Understanding intensity helps us know which emotions need more attention.

4. **Gradation:** Gradation follows how feelings slowly change over time, like going from being happy to feeling thrilled or from slight concern to real worry. It helps track emotional changes in a conversation or events.

5. **Comparative:** This compares emotions between two things, such as saying that one product is better than another. It is useful when people are making choices or expressing preferences.

Types of data in Sentiment Analysis: Here, we discuss the sentiment analysis system based on the data types used for analysis. SA with different types of data is presented in Fig. 1.3. The methodologies are broadly categorized as follows.

1. **Text-Based Sentiment Analysis:** Text-based sentiment analysis is designed to uncover emotional insights from written content, which is based on Lexicon [35]. It relies on dictionaries that assign semantic orientation and emotional intensity to words, often augmented by linguistic rules to handle negations, modifiers, and context. This type of analysis is beneficial for identifying harmful behaviors like bullying, aggression, trolling, and suicidal ideation, as well as for analyzing customer feedback to improve service quality and monitor employee performance. However, despite its utility, text-based sentiment analysis often struggles to accurately capture the full depth of human emotion, mainly due to the lack of contextual richness and emotional nuance in text alone. The available text datasets are designed for specific problems, and processing these datasets in a particular domain is a very challenging task; that

is, to make these datasets in usable format, some text preprocessing techniques are required. Additional technical challenges include the scarcity of annotated data, the complexity of natural language, and the difficulty of understanding the conversational context. The subcategories under this Text-based SA are:

(a) *Document level* sentiment analysis, which categorizes the total sentiment of an entire document or a unit of text. [21].

(b) *Sentence level* analysis analyzes separate sentences for subjective content and polarity, sometimes needing to make a differentiation between objective and subjective sentences [36].

(c) *Aspect based* sentiment analysis that recognizes sentiments for specific entities or attributes of entities presented in the text [37].

(d) *Fine grained* analysis based on multi-point scales (e.g., 1-5 star ratings) as opposed to mere polarity [38]. New forms of analysis include

(e) *Multilingual* sentiment analysis for cross-lingual use cases [39].

2. **Audio-Based Sentiment Analysis:** Audio-based sentiment analysis is growing rapidly as researchers work to understand emotions and feelings from the way people speak. This involves analyzing sound features like pitch, loudness, and unique voice patterns to detect emotional cues [40]. Tools such as Mel-frequency cepstral coefficients (MFCCs) and prosodic analysis help pull out these features effectively [12]. To make sense of the data, machine learning techniques, including SVMs and deep learning models like CNNs and LSTMs are widely used for classifying emotions [41]. Publicly available datasets like RAVDESS and IEMOCAP offer labeled voice recordings that support model training and testing [42]. These systems are finding real-world use in areas such as customer support, virtual assistants, and even mental health tracking [43]. However, challenges still exist, especially with noise in recordings or differences in how individuals speak. That is why strong preprocessing steps are vital [44], and combining audio with visual or text data, known as multimodal analysis, can lead to better results [45].

3. **Image-Based Sentiment Analysis:** Image-based sentiment analysis uses pictures to understand how people feel by looking at things like facial expressions, textures, and the overall scene. Smart computer models, especially ones

called CNN pick out important details from images to figure out emotions like happiness, sadness, or anger. These systems often work even better when they combine visual data with text or sound. But they also face some challenges, like different ways people show feelings in different cultures, blurry images, or parts of faces being covered. In real life, this technology is used in areas like marketing, checking customer feedback, supporting mental health, and improving how humans and computers interact. Before analyzing, the images usually go through steps like finding faces or mapping emotions to improve accuracy. Using models that are already trained, like ResNet or VGG, also helps tailor the system for specific tasks. Still, it is important to think about privacy and avoid unfair treatment when using these tools. Looking ahead, there is a push to make these systems lighter so they can run on smaller devices, and to make them easier to understand and trust. In the end, this technology helps computers better understand human emotions by learning from what they see. A systematic review of deep learning approaches in medical imaging, focusing on object detection tasks, was conducted by Zou et al. [46]. In the context of image and video-based sentiment analysis, Agrawal et al. [8] contributed valuable insights across multiple disciplines, advancing the field of affective computing for emotion recognition. Similarly, Meena et al. [47] discussed applying deep learning techniques to image sentiment analysis. Visual cues such as facial expressions and body language provide powerful indicators of emotional states [48].

4. **Video-Based Sentiment Analysis:** Today, with the widespread use of smartphones and other communication tools, a huge amount of video content is being shared online [49]. Social media platforms like YouTube and Facebook allow people to create and share videos on many topics, such as product reviews, movie opinions, politics, advice, and personal thoughts. These videos provide a powerful way for individuals to express their views and connect with others. As research in sentiment analysis continues to grow, relying only on text data is no longer enough, especially for complex tasks like detecting aggression. To get better and more complete results, it is also important to analyze images and videos [50]. In video-based sentiment analysis, among the many actions people perform, facial expressions play a very crucial role in conveying emotions and understanding sentiments accurately. Facial expression recognition

is a key component in video analysis for capturing human emotions [51]. There are two main approaches: spatial methods and spatiotemporal methods. Spatial methods analyze each video frame independently to identify facial features, while spatiotemporal methods examine how expressions change over time by looking at the relationship between consecutive frames [52, 53]. The foundational work by Ekman and Keltner [54] demonstrated that facial expressions are universal, providing a strong basis for emotion recognition. Early studies by Chen et al. [55] and De Silva et al. [56] combined visual and audio data to improve emotion detection. The videos serve as rich repositories of multimodal information. Beyond visual content, they include acoustic signals and textual elements, such as spoken language and subtitles, which together capture the full range of communication cues and nuances essential for comprehensive analysis.

5. **Multimodal-Based Sentiment Analysis:** To overcome these limitations, multimodal sentiment analysis has emerged as a more comprehensive alternative. By incorporating various data types such as text, images, and audio, multimodal systems [57, 58] sentiment analysis, aiming to harness the diverse and often distinct information from multiple sources to enhance system efficiency. Multimodal sentiment analysis involves the utilization of various combinations of three modalities, such as 'text-and-audio', 'text-and-image', and 'audio-and-image'. A system employing two modalities is called a bimodal sentiment analysis system, while a system incorporating all three modalities is termed a trimodal sentiment analysis system. However, the process of determining the most effective approach remains a challenging endeavour. Multimodal sentiment analysis delves into two distinct aspects: intramodal dynamics, referred to as View-specific dynamics, and intermodal dynamics, known as Cross-view dynamics [59]. Researchers have explored various methods to integrate information from different modalities in this pursuit. Intramodal dynamics, also known as view-specific dynamics [60], refer to interactions within a single modality, independent of others. For example, this includes analyzing how words relate to each other within a single sentence. In the context of language processing, especially when applied to multimodal sentiment analysis involving spoken language, capturing intramodal dynamics presents a unique challenge. Unlike written text, spoken opinions are often fragmented and less

structured. Consider a spoken phrase like, "I think it was alright... Hmmm... let me think... yeah... no... ok yeah." Such expressions, filled with pauses, hesitations, and shifts in thought, are rarely seen in written form, making their analysis significantly more complex. On the other hand, intermodal or cross-view dynamics focus on how different modalities, such as audio, text, and visual cues, interact. These interactions are typically classified into synchronous and asynchronous [61]. Synchronous dynamics occur when cues from different modalities align in time, for instance, when a person smiles while saying a positive word. In contrast, asynchronous dynamics involve cues that appear at other times, such as someone laughing a few moments after making a humorous remark. The multimodal systems that combine text with visual or auditory inputs have been discussed in [57]. The model in [62] detects the emotions instead of only polarity with multimodal data from different sources. The analysis type selection depends on the needs of the application, data available, and the preferred level of detail of insights to be performed in [34]. Additional hurdles include domain adaptation, where models trained in one context underperform in another; multilingual analysis, especially for under-resourced languages; and integrating multimodal data that combines text with images or audio [57].

The research challenges of these multimodal SA are (i) ethical considerations, such as privacy concerns, dataset bias, and the misuse of sentiment analysis technology, have also emerged as critical areas for research and regulatory focus [63]. (ii) Moreover, dealing with incomplete data where one or more modalities may be missing is a significant hurdle. Developing effective methods to process and analyze such incomplete multimodal input remains a considerable research challenge. (iii) In multimodal sentiment analysis, finding effective ways to model intramodal relationships within each modality while also determining the best strategy to integrate features across modalities. This delicate and complex balance requires thoughtful design and innovative techniques [64].

Tools applied in Sentiment Analysis: Here, classification and analysis of sentiment analysis are performed using tools such as machine learning and deep learning approaches (see lower part of Fig. 1.3)

1. **Machine Learning:** The machine learning based recognition system has two significant components apart from the preprocessing step, feature extraction, and the classification of the features. In the literature, there are different feature extraction (or presentation) methods and varieties of classification techniques. We may select one feature representation and one classifier as we wish; that is, there is some sort of independence in the selection of feature representation and classifier selection.

   Under the machine learning based approach, a developer has options, and we can select the feature extraction method or classification technique. It seems that manually we can select, so we refer to the methods that follow this pipeline as Hand Crafted System.

2. **Deep Learning:** In a deep learning based model, the model first computes the feature and then does the classification. These two steps will be executed under a single umbrella. We are not free to choose the feature representation and classification tools for the underlying architecture. If we want to do so, then we need to apply a new architecture. This type of model may be referred to as a Deep Learning System.

1.3 Pain Sentiment Analysis

The earlier sections discussed different types of sentiment. Among them, pain sentiment is especially important for improving modern smart healthcare systems. Pain sentiment is more important than other types of sentiment because it has a direct effect on diagnosing health problems, choosing the right treatment, and keeping patients safe, especially for those who cannot express their pain. In addition, measuring pain accurately with the help of technology allows doctors to decide treatment plans that are more suited to the needs of each person, which is a key goal of future healthcare systems. This section explains how a pain analysis system is designed and developed, which is the main focus of this thesis. Pain is a disagreeable sensation that can be physical or emotional [65]. It occurs when there is harm or a threat to the body and serves as an indication. Pain is related to feelings because the way we feel can influence how we perceive pain (for example, anxiety increases pain,

but relaxation decreases pain [66]). Furthermore, chronic pain has the potential to generate aversive emotions such as sadness or frustration. Methods of measuring feelings can identify emotions associated with pain through attention to words expressed (e.g., 'agonizing' or 'unbearable'), voice modulations (e.g., strained voice), or facial responses (e.g., grimacing [53]). This is useful for constructing models that relate individual pain accounts to objective emotion measures for better assessment and treatment. Sentiment analysis systems for pain are an advanced application of affective computing that automatically identifies, categorizes, and measures pain expressions from text, audio, and image data [3]. Such pain sentiment analysis (PSA) systems process patient-created content like clinical notes, pain diaries, and social media updates to determine subjective pain experience, usually applying natural language processing methods to discover pain descriptors (e.g., 'burning', 'throbbing'), intensity measures (e.g., numerical rating scales), and emotional valence (e.g., 'frustrating', 'debilitating') [67]. Fig. 1.4 shows an overall framework of a pain sentiment analysis system, and the individual modules are already discussed.

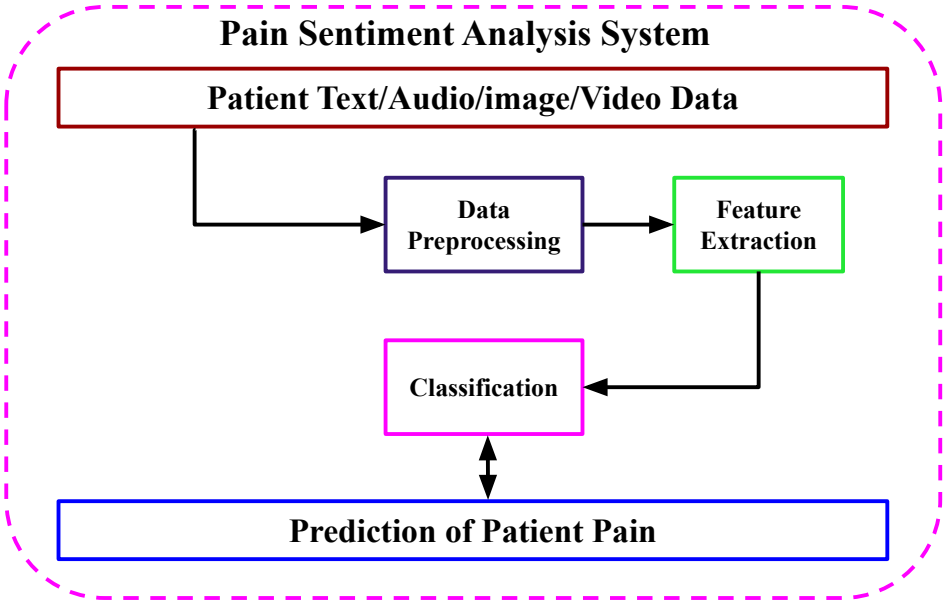


FIGURE 1.4: Block diagram of a Pain Sentiment Analysis System.

Pain sentiment includes: (i) text-based automatic recording of pain and its connection with mental health using machine learning tools that can understand both clear mentions of pain and hidden signs of stress in language; (ii) audio-based detection of pain in patients who cannot speak, by analyzing voice patterns that show signs of

stress in the nervous system; (iii) image-based tracking of facial expressions related to pain and checking for swelling using thermal images, which show heat changes linked to inflammation [60]; and (iv) video-based monitoring of small changes in facial movements to study face muscle problems with high accuracy [68]. When these different types of signals are combined to form (v) multimodal-based pain detection, they help turn the personal and often hard-to-measure experience of pain into clear, measurable information.

1.3.1 Text-Based Pain Analysis

Text-based pain signals [69] are important indicators of the pain experience of an individual. These include explicit pain-related terms (e.g., 'throbbing', 'stabbing'), frequent use of negative emotional language, and subtle features such as pronoun usage and speech disfluencies like hesitations or pauses. When analyzed, text-based sentiments are well characterized in terms of quantification and assessment of pain levels [70]. Variations in sentiment analysis are the differences in levels and approaches to measuring emotional tone and opinion, from general document-level tagging (estimating overall sentiment in a document) to granular sentence-level analysis (differentiating between subjective and objective content) and aspect-based sentiment analysis (focusing on sentiments about particular entities or characteristics). Apart from this, textual sentiment may not align with visual cues, so extracting meaningful features from each modality that contribute to overall emotional understanding is critical. In text-based pain detection, textual data plays a crucial role in capturing emotional cues. This process begins with preprocessing steps, including noise removal, tokenization, and normalization. Subsequently, feature extraction techniques such as Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and word embeddings are employed to identify key sentiment indicators. These features are then input into machine learning classifiers (e.g., Support Vector Machines, Random Forests) or deep learning architectures (e.g., Long Short-Term Memory Networks, Transformers) to perform pain classification.

In text-based pain recognition, there are several difficulties in processing and interpreting text-based pain, such as subjectivity and variability in individual descriptions, which makes standardization challenging; ambiguity and context dependence,

which can result in misinterpretations of the nature or intensity of pain; limited expressiveness and data scarcity because patients may use incomplete descriptions or limited vocabulary.

1.3.2 Audio-Based Pain Analysis

Pain can be measured through detectable changes in speech and breathing patterns, known as vocal and respiratory pain signals. Vocal pain signals include increased tension, pitch variability (jitter), and altered speech rhythm or articulation [71], while respiratory pain signals involve shallow breaths, irregular pauses, and strained inhalation or exhalation [72]. These biomarkers reflect physiological stress responses, such as voice instability (shimmer, harshness) and altered respiratory-phonatory coordination, indicating pain-induced autonomic and muscular effects. Audio-based analysis can complement traditional pain assessments, particularly for non-verbal patients. The audio-based PSA Systems processes audio data by first applying noise reduction using a Kalman Filter. Usually, three types of features are used in audio-based analysis: statistical features for temporal and spectral changes [73], Mel-Frequency Cepstral Coefficients (MFCCs) for spectral properties [11], and spectral features for frequency domain patterns [74]. These features are evaluated using two classification approaches: (1) conventional machine learning models applied to each feature set individually, and (2) a Fully Connected Network (FCN) trained on combined features to assess whether integration improves pain classification performance. Cutting-edge models such as Wav2Vec 2.0 are pushing the boundaries of what is possible in this space [75], although it is important to keep in mind ethical issues such as bias in training data [76]. Looking ahead, there is a strong focus on making these systems work well in different languages and cultures [77].

The challenges of audio-based pain signals include the following: the need for large, diverse datasets to train robust models, which are often scarce due to privacy concerns and the sensitive nature of pain-related audio recordings; the difficulty of accurately distinguishing pain-related sounds from other emotional or physical sounds, which could result in misclassification; and the variability in vocal expressions of pain across individuals and cultures, which makes standardization difficult. It can also be challenging to identify pain-specific audio patterns, as variations in voice

characteristics are influenced by factors such as age, gender, accent, speaking style, and emotional state. Few high-quality, labeled audio datasets are available for pain identification, and it is complex and subjective to accurately annotate pain levels from audio alone.

1.3.3 Image-Based Pain Analysis

The static visual signal, such as facial images, can reveal different expressions of pain, such as brow lowering, eye tightening, or nose wrinkling [78]. These subtle expressions are the result of specific movements of facial muscles and serve as reliable indicators of pain intensity. Unlike prolonged emotional expressions, pain-related facial cues typically appear involuntarily and last only 0.5 to 4 seconds after the onset of pain, making them particularly useful for rapid screening in clinical environments [79]. Automated vision systems can detect these signs by monitoring facial regions and identifying relevant Action Units (AUs), thus supporting traditional pain assessment techniques. An input image is preprocessed to extract and improve the facial region in image-based pain detection systems. To identify and measure the degree of pain, this preprocessed face image will be analyzed. Interpretable insights into muscle activity related to pain are provided by hand-crafted features, such as those based on Facial Action Units. They are more transparent and reliable because they align with accepted biomechanical and psychological concepts. In addition, deep learning techniques such as Vision Transformers and CNNs train advanced high-level features that capture complex visual patterns, thereby improving the precision and robustness of pain intensity detection.

In image-based pain detection, facial expressions face several challenges, including noise in the data, motion blur, and both macro- and micro-expressions of facial regions, as well as data imbalance across different pain classes. Additional difficulties arise from variability in facial expressions between individuals and cultures, which complicates the development of universally accurate models. Factors such as lighting, camera angle, and image quality can impact visibility and interpretation of pain indicators. Moreover, the subtle and transient nature of some expressions related to pain increases the risk of misclassification or missed cues. Ethical and privacy concerns also arise when capturing and analyzing facial images for pain assessment,

further complicating the deployment of such systems, which is a more challenging concern than text and audio-based pain signals.

1.3.4 Video-Based Pain Analysis

The dynamic visual pain signal, such as those captured in facial video sequences, reflects how facial and bodily movements change over time, revealing when pain begins, how long it lasts, and how its intensity fluctuates. Examples include furrowing the brows or pursing the lips in response to discomfort [80]. These time-based signals reveal specific pain-related patterns, including synchronized muscle movements and protective behaviors, such as guarding an injured area. Advanced video analysis techniques can track dynamic features, such as the rate of facial changes and the interaction of action units, to distinguish between acute and chronic pain conditions [81]. Compared to static images, this dynamic approach offers richer clinical insights by capturing real-time expressions and behavioral cues associated with pain. For each patient, video sequences of facial expressions are analyzed to assess the intensity of pain. Individual frames are first preprocessed after detecting and cropping the facial regions. The features extracted from these preprocessed frames are then concatenated to create a comprehensive feature representation. This final representation is passed through fully connected neural networks for classification, creating a unified pipeline that integrates feature extraction, fusion, and classification. The deep learning-derived features used for video-based pain detection often outperform traditional hand-crafted ones due to their ability to capture high-level representations automatically. With the rapid advancement of deep learning, particularly in solving complex visual problems, these techniques have become increasingly effective and reliable.

In terms of image-based pain signal challenges, video-based pain signals also suffer from similar types of challenges, including the variability of video sequencing length. The short-duration videos may not capture the high pain intensities, while the longer-duration videos may contain more or less similar patterns of various pain intensities that might be difficult to distinguish.

1.3.5 Multimodal-Based Pain Analysis

In multimodal pain analysis, digital sentiment serves as a computational indicator of emotional and physiological states of a person by integrating diverse data streams. These include text (e.g., lexical sentiment patterns and implicit linguistic markers such as word usage [82]), audio (e.g., vocal biomarkers like pitch variation and speech rhythm for detecting stress [83]), images (e.g., facial action units like brow furrowing that reflect pain intensity [84]), and video (for example, monitoring dynamic micro-expressions across time to capture pain responses [19]). These multimodal indicators translate subjective pain experiences into objective, quantifiable data, enabling machine learning systems to support pain assessment, mental health monitoring, and precision diagnostics while also addressing challenges such as privacy, clinical validation, and variability in populations. Advanced systems often rely on domain-specific lexicons and ontologies to interpret the nuanced vocabulary of pain as it varies across conditions and demographic groups [85]. In audio-based sentiment recognition, key features such as pitch fluctuations, voice instability, and vocal stress are particularly informative in identifying non-verbal expressions of pain. For image-based analysis, visible facial cues such as furrowed brows, wrinkled noses, and puffed lips act as crucial indicators of discomfort [66]. Moreover, examining how these expressions change over time is critical for video-based analysis, providing insight into the temporal dynamics of pain. Machine learning and deep neural networks, trained on labelled clinical datasets, have demonstrated strong performance in distinguishing between acute and chronic pain expressions while accounting for individual differences in how pain is reported [86]. In this thesis, multimodal pain detection refers to the process where each modality is employed independently to build pain prediction models, enabling an evaluation of the respective strengths and limitations of these modalities for the PSA Systems. Then, the outcomes of each modality are combined to obtain the result for the multimodal PSA Systems.

However, ongoing challenges include handling linguistic variability, accommodating cultural differences in expression, and ensuring computational models remain both clinically relevant and efficient [87]. Emerging applications range from real-time monitoring of postoperative patients to long-term tracking of chronic pain, with the potential to make pain assessments more objective and complementary to traditional clinical evaluations [88].

The simultaneous multimodal data may not always be available. They could be highly dependent on environmental conditions, such as lighting for video, background noise for audio, or network limitations for real-time streaming, which can affect the reliability and completeness of pain assessment. Ongoing research and system development actively address ethical concerns related to patient privacy, data security, and potential biases in pain assessment algorithms [89].

1.3.6 Hand-Crafted Features and Classifiers

We classify the sentiment analysis systems as Hand-Crafted systems and Deep Learning systems, considering the tools (machine learning / deep learning) they have used. We discuss this categorization in the paragraph 'Tools applied in Sentiment Analysis' of Section 1.2. First, we discuss some standard feature extraction methods and then some machine learning classifiers.

1.3.6.1 Hand-Crafted Feature Representations

Hand-crafted features are extracted from raw data to capture domain-specific patterns, and they have played a crucial role in machine learning across text, audio, and image analysis. In text processing, techniques such as Bag-of-Words [90] and TF-IDF [91] convert unstructured language into numerical vectors, enabling tasks like sentiment analysis and spam detection. For audio, features such as RMS energy [92], MFCCs [11], and spectral centroid [93] summarize acoustic characteristics, supporting applications like speech recognition and music classification. In image analysis, methods such as HOG [13] and LBP [94] encode textures, shapes, and edges essential for object detection and facial recognition. These features are highly interpretable, efficient, and perform well even with limited training data, making them ideal for real-time systems or cases where model transparency matters. Although deep learning has shifted its focus toward automatically learned features, hand-crafted features remain valuable due to their reliability in low-data environments, computational simplicity, and utility as benchmarking baselines. Ultimately, they bridge raw data and machine learning models, transforming unstructured inputs into structured, informative representations that boost model performance across various

domains. Some of the hand-crafted features are discussed below. We discussed only those feature extraction methods that are used in this thesis.

- **Bag of Words (BoW):** It is a simple yet powerful technique in natural language processing (NLP) that converts text into numerical vectors by counting the frequency of each word in a document [90]. It ignores grammar and word order, treating each document as a collection of individual words. BoW is widely used in tasks such as text classification, sentiment analysis, and spam detection due to its ease of use and efficiency. One of its main strengths is that it transforms messy, unstructured text into a format that machine learning models can process efficiently. It is also highly interpretable—word frequencies clearly show which terms dominate a document. However, BoW does not understand word meaning or context and treats all words equally, which can be a drawback. Without preprocessing, common words like 'the' or 'and' may drown out more informative ones. It can also produce large, sparse matrices when working with large vocabularies, which impacts performance. Despite its limitations, BoW remains a reliable baseline in NLP, particularly for small datasets or when quick results are required. With enhancements like n-grams, it can even capture some local word patterns, making it more flexible for basic language tasks.

- **Term Frequency - Inverse Document Frequency (TF-IDF):** This is a more sophisticated alternative to the basic Bag-of-Words (BoW) model for representing text. Unlike BoW, which only counts word occurrences, TF-IDF considers how important a word is in a document relative to how common it is across the entire corpus [91]. It achieves this by combining term frequency (TF) with inverse document frequency (IDF), thereby reducing the weight of common words like 'the' while boosting more unique and meaningful terms. This makes TF-IDF highly effective in tasks such as search engines, document classification, topic modelling, and clustering. Because it emphasizes keywords that matter, TF-IDF often produces better features for machine learning than Bag-of-Words (BoW). It also allows for customization through various scaling strategies, making it adaptable for diverse applications. However, like BoW, TF-IDF does not understand word order or meaning, which can limit its use in context-heavy tasks. Still, it is a strong go-to for many NLP pipelines,

particularly in the early stages of feature extraction. TF-IDF is often combined with other methods, such as word embeddings, to enhance performance further. Despite being slightly more complex and computationally demanding than BoW, its ability to strike a balance between simplicity and effectiveness has made it a staple in modern text analysis.

- **Statistical Feature:** It is a fundamental category in audio analysis, and this thesis focuses on Root-Mean-Square (RMS) energy [92] as a representative feature. RMS energy is widely used to measure the average power of an audio signal, providing a reliable sense of loudness over time. It is calculated by squaring each audio sample, averaging them, and then taking the square root, offering a smoothed measure that reflects sustained loudness rather than brief peaks. Unlike peak amplitude, which only captures sudden spikes, RMS energy aligns better with how humans perceive sound intensity. Its simplicity makes it computationally efficient, enabling real-time use in systems like hearing aids or voice-activated devices. RMS energy is commonly applied in speech activity detection, music volume balancing, and audio normalization. In speech processing, it helps differentiate between silence and active speaking, while in music, it supports consistent playback levels. However, it only captures amplitude information, not frequency details, so it is often paired with spectral features for richer analysis. Despite this limitation, RMS energy remains a go-to feature for audio segmentation tasks and dynamic audio adjustments. Its balance of simplicity, efficiency, and practical relevance makes it a reliable tool in both everyday and specialized audio applications.

- **Mel-Frequency Cepstral Coefficients (MFCCs):** These features are among the most widely used features in audio signal processing because they closely mimic how humans perceive sound [11]. MFCCs transform raw audio into a compact set of numbers that capture the power spectrum of a signal. This process involves applying a Fourier Transform, mapping frequencies onto the Mel scale (which reflects human pitch perception), and then using a Discrete Cosine Transform to produce a clean set of coefficients. These features are particularly valuable in speech and music tasks because they highlight the parts of the audio to which our ears are most sensitive. Their resistance to background noise makes them ideal for applications such as voice recognition, speaker

identification, and smart assistants. MFCCs are also efficient, providing rich information in a reduced form that simplifies machine learning models without sacrificing performance. Despite their strengths, they may struggle with rapidly changing sounds and require fine-tuning for optimal results. Enhancements like delta and delta-delta coefficients can capture changes over time, improving performance in dynamic audio environments. MFCCs continue to be widely used in various applications, ranging from music genre classification to forensic analysis, due to their balance of simplicity and accuracy. Even with the rise of deep learning, MFCCs remain a trusted, go-to tool in audio analysis due to their proven effectiveness and efficiency.

- **Spectral Feature:** This refers to frequency-domain characteristics of audio signals, and among the many types—such as spectral roll-off and spectral bandwidth—this thesis focuses on spectral centroid. The spectral centroid [93] gives us an idea of the 'brightness' of a sound by calculating the weighted average of frequencies, with the weights based on their magnitudes. Essentially, it indicates where most of the energy in the signal is located: higher centroids suggest a sharper or brighter tone, while lower ones point to a darker, more mellow sound. This feature is especially useful in music analysis, as it helps differentiate between instruments or genres based on tone. For example, a violin often has a higher spectral centroid than a bass guitar due to its brighter timbre. One of its biggest strengths is that it is easy to compute and interpret, which makes it practical for real-time systems. You'll often find it used in tasks like music information retrieval, sound classification, and audio segmentation. However, on its own, the spectral centroid may not be sufficient in complex scenarios, so it is commonly combined with features like MFCCs or spectral bandwidth. Despite this limitation, it remains valuable because it links raw signal data to how we naturally perceive sound. Its speed and simplicity make it a staple in modern audio analysis, particularly where tonal quality is a concern.

- **Local Boundary Pattern (LBP):** This feature is a very well-known and robust technique for texture identification from images. Previously, the LBP technique was used to fulfill the necessary measures of image contrast in the local scope [94]. In the current scenario, the modification of image texture and

the use of local primitives, such as the LBP technique, are widely employed. The working principle of the LBP operator is as follows: First, consider an input image I and label every pixel with the help of a threshold value. Now, select a particular pixel as the central pixel and consider a 3×3 neighbourhood around it. Perform a comparison between each of the 3×3 neighbourhood pixel values and central pixel values. Each cell value in the neighbourhood is labelled with 0 or 1, depending on whether the cell value is less than or greater than that of the central value. Following this approach, binary bits will be generated based on the consideration of a bit string. This binary string gives a local pattern of LBP codes computed for the entire image. Each LBP code for the whole of the input image I is considered as a micro-texton, and using these textons, a 256-bin histogram is prepared.

- **Histogram of Oriented Gradients (HOG):** This feature primarily retains shape and texture-oriented information [13]. Furthermore, it has been successfully used for human detection and is now efficiently applied for improved face recognition. The HOG [95] technique has similarities with the SIFT method. HOG feature techniques are determined using block-wise orientation, aided by Sobel edge detectors, to measure horizontal and vertical gradient details. These details play a crucial role in determining whether the orientation of feature representation and magnitude information correspond to an image. This feature extraction technique considers an image size 128×128, then computes an 81-dimensional feature vector.

These Hand-Crafted feature extraction techniques are applied to extract features from the data. The extracted features are then used to train various machine learning classifiers and evaluate their performance. Now, we are going to present some classifiers that are used in this thesis.

1.3.6.2 Machine Learning Classifiers

In this thesis, the following machine learning classifiers have been employed to analyze the pain level from the hand-crafted features.

- **k-Nearest Neighbors (kNN):** This is a simple yet effective method used for classification tasks. As a non-parametric, lazy learning algorithm, kNN does not build a model during training. Instead, it stores all training data and makes predictions by measuring distances, typically Euclidean, between the new input and all stored samples [26]. kNN decides the class of a new data point by looking at the 'k' closest neighbours and assigning the most frequent label among them. Its ease of implementation and intuitive logic make it a popular choice for prototype-based learning. However, its prediction speed slows down as the dataset grows because it must compute distances to every point in the training set. Choosing the right value for k is crucial: a small k might lead to overfitting, while a large k can introduce bias. Additionally, because the algorithm is distance-based, proper feature scaling is essential, and irrelevant features can degrade its performance. kNN handles multiclass classification naturally, requiring only minor modifications. It performs well in low-dimensional settings but struggles in high-dimensional ones due to the curse of dimensionality. Various distance measures (like Manhattan or Minkowski) can be used depending on the data. kNN is widely used in areas such as recommendation systems, pattern recognition, and anomaly detection. To make it efficient on large datasets, indexing structures such as KD trees are often used. Despite its simplicity, kNN remains a competitive choice for many real-world problems with complex or irregular decision boundaries.

- **Logistic Regression (LR):** This is a foundational algorithm used for predicting binary outcomes. It models the probability that a given input belongs to a particular class using the logistic (sigmoid) function to map a linear combination of features to a value between 0 and 1 [24]. The model is trained using maximum likelihood estimation, often optimized via gradient descent, to minimize the cross-entropy loss. One of the strengths of logistic regression is its interpretability: each coefficient directly influences the log odds of the outcome, making it particularly useful in fields such as healthcare and finance for risk estimation. Regularisation methods, such as L1 (Lasso) and L2 (Ridge), are frequently applied to prevent overfitting and perform feature selection, particularly in high-dimensional datasets. While it assumes a linear relationship between input features and the log odds of the target, this limitation can be addressed through feature engineering or kernel tricks. Although it

is designed for binary classification, logistic regression can handle multi-class tasks using strategies such as one-vs-rest or softmax regression. It scales well to large datasets and is computationally efficient. Even today, logistic regression remains a reliable and interpretable baseline for many classification problems.

- **Support Vector Machine (SVM):** This is a powerful supervised learning model that aims to find the best decision boundary—or hyperplane—that separates data into classes with the maximum possible margin [96]. What makes SVM distinctive is its reliance on a small subset of critical training examples, called support vectors, to define this boundary, which improves memory efficiency. For datasets that are not linearly separable, SVM uses kernel functions to map the input features into a higher-dimensional space where a separating hyperplane may exist. Standard kernels include polynomial, radial basis function (RBF), and sigmoid, each of which is suited to different data patterns. The regularisation parameter (C) balances the trade-off between maximising the margin and minimising classification errors. SVMs are particularly effective in high-dimensional spaces and scenarios where the number of features exceeds the number of samples, such as in text or genomic data. SVMs generalize well due to their maximum-margin principle, which helps avoid overfitting. They can also be adapted for multi-class classification using strategies like one-vs-one or one-vs-all. However, SVMs require careful selection of kernel and parameters, and training time can be substantial for very large datasets.

- **Decision Tree (DT):** This is an intuitive model that mimics human decision-making by using a flowchart-like structure of conditional statements. It works by recursively splitting the dataset into subsets based on feature values, aiming to reduce impurity at each step [25]. The algorithm selects features to split on by optimizing criteria such as information gain (based on entropy) or Gini impurity. One of the key strengths of decision trees is their ability to handle both numerical and categorical data without requiring extensive preprocessing or normalization. Because of their visual nature, decision trees are easy to interpret and explain. They can model non-linear relationships and capture interactions between variables through their branching structure. However, they are sensitive to small changes in the data, which can lead to significantly different trees—a phenomenon known as high variance. To combat overfitting,

pruning techniques (such as reduced error pruning) are used to simplify the tree after it has been built. Despite their limitations, decision trees are widely used and form the foundation for more advanced ensemble models, such as random forests and gradient boosting.

- **Random Forest (RF):** This is a robust and widely used ensemble learning method that improves upon decision trees by combining multiple decision trees into a single predictive model [97]. Each tree in the forest is trained on a different bootstrap sample of the data (i.e., sampling with replacement), and during training, only a random subset of features is considered at each split. This randomness helps decorate the trees and reduce overfitting. For classification, the final output is determined by majority voting, and for regression, it is the average of all tree outputs. Random forests are known for their stability, high accuracy, and ability to handle high-dimensional data and noisy features. The model provides built-in methods for estimating feature importance, which helps to understand the behavior of the model. It also scales well and supports parallel training, making it suitable for large datasets. While random forests are less interpretable than individual trees, they usually perform better and are more resilient to overfitting. Random forests are frequently applied in domains such as bioinformatics, finance, and environmental modelling. As 'off-the-shelf' models, they require minimal tuning and preprocessing, offering excellent performance with little hassle.

1.3.7 Deep Learning Features and Classifiers

Deep learning-based schemes are hybridized methods that encapsulate feature representation along with fully connected network-based classifiers with optimized activation functions. The following hybridized deep methods have been developed and employed in the proposed PSA Systems of this thesis.

- **Convolution Neural Networks (CNNs:)** Over time, it has been observed that these proposed feature extraction techniques mainly suffered from two types of problems: too over-engineered or too general as a result of that designing process of the filter becomes hard to generalize or too simple. To overcome

these situations, some feature learning techniques have been proposed. Using these feature learning techniques, all the relevant and available features from the images are extracted automatically, which replaces the efforts of manual feature engineering. Neural networks are there to help in feature learning with Multi-Layer Perceptron (MLP). There exist various drawbacks in MLP. For RGB images, the perceptrons are used by MLPs, and as a result, the weight amount within the network becomes unhealable within a short period. Another common problem of MLP is they are translation variants; thus, reactions of MLP to an input image and its shifted version is different. In the case of CNN mainly various kinds of filters are applied to capture relevant image features with the help of convolution operation. This convolution operation takes place throughout the image, and the filter is shifting pixel by pixel, starting from the top-left corner to the bottom-right corner [98]. With the extracted feature for any particular object within an image, the filters always guide how strongly a particular object can be found within the image. In that context, the location of the object and how many times the object appears within the image the performance does not get affected. In CNN, various layers are used, such as convolution, pooling, fully connected, and dense layers during the feature representation of an image [99]. Each CNN architecture consists of several layers, as outlined below:

(i) Convolutional Layer: In this layer, a convolution operation is performed on the input image (denoted as I) using a mask or kernel of size $t \times t$. The result is a set of feature maps, denoted as Z. The convolution process involves sliding the kernel over the input image I and applying a non-linear function at each position. For each position, element-wise matrix multiplication is carried out, and the results are compiled into a feature map, corresponding to the central portion of the mask pattern. The operation is defined as follows: $Z^{[l][M]} = ReLU(W^{[l][M]} * I + b^{[l][M]})$.

- M denotes the convolutional layers in the proposed CNN architectures.
- $W^{[l][M]}$ represents the weights of the M-th convolutional layer.
- $*$ signifies the convolution operation.
- $b^{[l][M]}$ represents the biases of the M-th convolutional layer.

(ii) Max Pooling Layer: This layer is designed to enhance the performance of the CNN architecture by applying max-pooling, a technique that calculates the maximum values within segments of a feature map, leading to down-sampled feature maps. Max-pooling offers benefits such as uniform translation and reduced computational overhead, while also improving the selection of distinctive features from the input image I. The max-pooling operation is defined as: $P^{[l][M]} = MaxPooling(Z^{[l][M]}, poolsize = (2, 2))$.

(iii) Batch Normalization Layer: Batch normalization helps stabilize gradients during training, making it easier to use higher learning rates and improving the overall performance of image classification tasks [100]. It reduces the impact of parameter scale or initialization problems. For a layer l in the CNN, the following batch normalization equations are applied:

- First, compute the mean and variance over the mini-batch: $\mu = \frac{1}{m}\sum_{i=1}^{m} P_i^{[l][M]}$; $\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(P_i^{[l][M]} - \mu)^2$, where m is the batch size, $P_i^{[l][M]}$ is the activation of the i-th sample in the mini-batch, μ is the mean, and σ^2 is the variance.
- Normalize the inputs: $(P_i^{[l][M]})\hat{} = \frac{P_i^{[l][M]} - \mu}{\sqrt{\sigma^2 + \varepsilon}}$, where ε is a small constant added to avoid division by zero and enhance stability.
- Scale and shift the normalized inputs: $Y_i^{[l][M]} = \gamma(P_i^{[l][M]}) + \beta$, where $Y_i^{[l][M]}$ is the normalized and scaled output, β is the shifting parameter, and γ is the scaling parameter learned during training. Both β and γ are updated via backpropagation along with other network parameters.

(iv) Dropout Layer: Dropout is a regularization technique used in CNNs to prevent overfitting. During training, a fraction of neurons are randomly deactivated, which forces the network to learn more robust and generalized features. By reducing the interdependence between neurons, the dropout layer enhances the generalization ability of the model, improving performance on unseen data.

(v) Flatten Layer: This layer connects the convolutional layers to the fully connected layers in the CNN architecture. It reshapes the multi-dimensional feature maps into a one-dimensional vector, making them compatible with the fully connected layers. This transformation enables the network to

capture higher-level patterns and correlations for tasks like classification and decision-making.

(vi) Fully Connected Layer: The fully connected layers play a critical role in transforming the feature maps into isolated vectors for classification. This step generates class scores and builds the final vector, whose size corresponds to the number of class labels.

(vii) Dense Layer: The dense layer establishes the connection between the input and output layers through the use of weights. The input to this layer is the output from the flattening layer, denoted as F, and is defined by the equation: $A^{[l][R]} = Activation(W^{[l][R]}F + b^{[l][R]})$, where $A^{[l][R]}$ is the output of the R-th fully connected layer.

(viii) Dropout Layer (within Fully Connected Layer): Dropout is also applied within the fully connected layer to further regularize the model and prevent overfitting.

(ix) Output Layer: The final classification task occurs in this layer. The Softmax activation function is used for classification, and the final output is calculated as: $Y = Softmax(W^{[l+1]}A^{[l]} + b^{[l+1]})$, where $A^{[l]}$ is the input from the fully connected layer, and Y is the final output of the classification.

- **Deep Attention Networks (DANs):** Attention mechanisms revolutionized how machines process sequential and structured data by enabling models to focus selectively on the most relevant parts of the input [101]. Unlike earlier models that compress input into a single, fixed-size vector, attention provides context-aware representations at each step of the output. The core idea is the query-key-value mechanism:

  (i) A query compares against a set of keys to compute attention scores.

  (ii) These scores determine how much focus should be given to each value.

  This approach allows the model to dynamically weigh different parts of the input, improving its ability to handle long sequences and complex dependencies. Attention became widely known through its use in transformer models like BERT and GPT, which use a specific type called self-attention. Unlike

RNNs or LSTMs, transformers do not process data sequentially, making attention highly parallelizable and efficient, even with long-range dependencies. In addition to NLP, attention has been successfully applied to Computer vision, Multimodal learning, and Reinforcement learning. An advanced form called multi-head attention allows the model to attend to information from multiple perspectives at once, further enriching its understanding of the data. One of the standout benefits of attention mechanisms is their interpretability. You can often visualize attention weights to see which parts of the input most influenced the prediction of a model. This not only helps in debugging but also builds trust in model decisions. While attention-based models usually require larger datasets and more compute power, they offer unmatched flexibility and performance across a wide range of AI tasks. They have become a cornerstone of modern deep learning, offering a unified framework for building models that are both powerful and adaptable.

- **Long Short Term Memory (LSTM):** LSTM is a type of recurrent neural network (RNN) specifically designed to overcome one of the major limitations of traditional RNNs—the vanishing gradient problem [102]. What sets LSTM apart is its memory cell and a set of gating mechanisms that intelligently control the flow of information over time.LSTM has three key gates, first one is the input gate decides how much of the new information should be written into the memory cell, the second one is the forget gate decides what part of the existing memory should be discarded, and the last one is the output gate controls how the memory content is passed to the next time step. This architecture enables LSTM to retain and update information adaptively, making it exceptionally good at learning long-term dependencies across sequences, something traditional RNNs struggle with. Thanks to these features, LSTMs are well-suited for tasks involving sequential data such as language modeling, speech recognition, time-series forecasting, and anomaly detection in sensor streams. They are particularly effective when the context or signal from distant past events needs to influence the present decision. While LSTMs are more computationally demanding than simple RNNs due to their gate operations, the performance payoff in modeling long sequences is often worth it. Enhancements like peephole connections further refine their ability to manage memory based on internal states. Even though transformer models have

become the new standard in many areas, LSTMs are still valuable, especially in resource-constrained environments or when interpretability and efficiency matter. Their modular nature also makes them easy to combine with other networks, such as convolutional LSTMs for tasks involving spatio-temporal data.

- **Bidirectional LSTM (BiLSTM):** BiLSTM enhances the basic LSTM structure by adding a second LSTM that processes the input sequence in the reverse direction [103]. This means BiLSTMs can learn from both past and future context, making them particularly powerful for understanding sequences where the meaning of one element depends on what comes both before and after. In practice, BiLSTMs consist of two separate LSTM layers—one reading the input forward and the other backward. Their outputs are then combined (typically via concatenation or averaging) to form a more comprehensive representation of the data. This dual-view approach has shown impressive results in natural language processing (NLP) tasks like named entity recognition, sentiment analysis, and part-of-speech tagging, where context from both directions is crucial for accurate prediction. However, BiLSTMs have a trade-off: they require access to the entire sequence upfront, which makes them unsuitable for real-time or streaming applications. Their larger size and computational cost are usually justified by their higher accuracy in offline processing tasks. In many cutting-edge systems, BiLSTMs are integrated with attention mechanisms or Conditional Random Fields (CRFs) to further boost performance. They've also found success beyond NLP in bioinformatics, for instance, by modeling dependencies in protein sequences. Although transformers have become more dominant recently, BiLSTMs remain a strong choice for sequence modeling when you need a good balance between performance, interpretability, and contextual understanding.

1.4 Challenges of the Thesis

The challenging issues discussed above have led to the consideration of the following problems in this thesis.

1. **Noisy Data:** In pain sentiment analysis (PSA) system, any input that interferes with the precise identification and interpretation of pain-related cues is referred to as noisy data. This includes low-quality, inconsistent, misleading, or irrelevant textual content in text-based PSA system. Background noise, recording errors, or emotional vocalizations unrelated to the pain may distort acoustic elements in audio-based PSA system. Visual distortions that obstruct important facial or body cues, such as blurriness, low resolution, facial shadows, and occlusions from hands, hair, glasses, or masks, can impact image-based PSA system. Aside from motion artifacts, camera instability, position and viewpoint fluctuations, and non-expressive or ambiguous behavior across frames, video-based PSA system have similar problems to image-based systems. Furthermore, the visibility and interpretation of pain indicators are significantly influenced by factors such as lighting, camera angle, and overall image quality.

2. **Cross Person Variability:** Individual and modal diversity in PSA system creates several issues. It is challenging to reliably interpret pain-related feelings in text-based PSA system due to variations in how people express pain, such as different levels of severity or aggressive language. The diversity in vocal expressions among people, emotional states, and cultural backgrounds makes it challenging to identify pain-related vocalizations in audio-based PSA system. Individual emotional responses, ethnic diversity, and inconsistent facial features, even within the same pain class, make it challenging for image-based PSA system to generalize pain accurately. Variations in feature representation techniques and model deployment methods further complicate the recognition of pain. These difficulties are exacerbated in video-based PSA system by frame-level irregularities, varying video acquisition conditions, and variations in posture, lighting, and temporal consistency, all of which impact the accuracy of pain detection systems.

3. **Imbalanced Data:** When data samples are not evenly distributed across different pain intensity levels or classes—such as low, semi-low, moderate, severe, and very high pain—biased models may be produced that perform poorly on underrepresented classes and tend to favor the majority classes. These class imbalances are frequently observed in datasets related to pain,

which makes it challenging to classify pain fairly and accurately across all severity levels.

4. **Subjectivity in Pain Perception:** Individual differences in how people express and experience pain make subjectivity in pain perception a major barrier for text-, audio-, image-, and video-based pain detection systems. People use a variety of terminology and emotional tones, influenced by their educational and cultural backgrounds, to communicate pain in text-based systems. Variations in vocal emotions, where pain levels are influenced by personality, gender, and cultural conventions, provide difficulties for audio-based systems. Different facial expressions, such as micro to macro expressions, where pain cues vary from person to person, are challenging for image-based algorithms to interpret. Temporal variations and body language ambiguity in video-based systems make it challenging to interpret pain across sequences accurately.

5. **Feature Distinctiveness:** In multimodal sentiment and pain analysis, one key challenge is feature non-distinctiveness, where features extracted from different modalities or contexts fail to distinguish between sentiment or pain levels clearly. This issue becomes more pronounced with the inclusion of fine-grained sentiment scales (e.g., 1–5 ratings), multilingual or cross-lingual analysis, and the integration of diverse modalities such as text, audio, and visual signals. Furthermore, the detection of discrete emotional states (e.g., joy, anger) adds complexity, as overlapping expressions across emotions and modalities may reduce feature discriminability. The extent and type of variation introduced in a system are typically guided by the goals of the application, the nature of the data, and the level of analytical precision required.

6. **Data Synchronization:** Ensuring precise alignment across multiple physiological signal modalities, including text, audio, pictures, and video, is challenging and prone to mistakes. In multimodal fusion techniques (such as early or late fusion), synchronization is crucial, as any feature misalignment can introduce noise and significantly impair model performance. The difficulty of synchronizing data from many sources is made worse in low-resource environments due to poor hardware or software support. Furthermore, the fact that datasets are gathered from a variety of contexts—with differing modalities,

environments, subject characteristics, sample quality, and sensor configurations—makes reliable data synchronization even more important for successful multimodal integration.

7. **Multimodal Models Integration:** Training classifiers on high-dimensional multimodal data poses a risk of overfitting, especially when the available dataset is small. Effective integration of features during the fusion stage requires advanced techniques to minimize redundancy while preserving the unique contributions of each modality. The limited availability of labeled multimodal pain datasets further restricts robust model training and evaluation. Additionally, deploying multimodal pain sentiment analysis in real-time environments demands high computational efficiency and rapid decision-making, which can be challenging given the complexity and resource intensity of such models.

1.5 Objectives of the Thesis

This thesis aims to design, develop, and study different pain sentiment analysis systems based on different types of data using different tools. Also, design and develop multimodal PSA systems by combining different types of data like 'text-and-audio', 'text-audio-and-video', etc. We should integrate text, audio, image, and video data to detect and classify pain intensity levels accurately. By addressing these challenges, the study acknowledges and mitigates several important issues, including noisy data, subjectivity in pain expression, data imbalance, non-distinctive features, and the difficulty of synchronising data across modalities. This thesis has the following objectives:

1. To apply reliable and effective data preprocessing methods to text, audio, image, and video data with the goals of standardizing inputs, removing noise, and lowering variability. This preprocessing will improve the overall quality of the data, making it possible to extract discriminative and significant features that enhance model generalization, lower computational overhead, and help multimodal pain sentiment categorization perform more accurately and consistently.

2. To address the inherent subjectivity in pain perception, given that pain detection is essentially an open-class problem, the thesis attempts to apply adaptive modeling techniques—such as transfer learning and domain adaptation—to optimize models based on subject-specific features. Furthermore, to create generalized models that can successfully account for individual and demographic diversity in pain expression, a variety of training data involving participants from different cultures, age groups, and genders will be incorporated.

3. To accept the challenges of feature non-distinctiveness, this thesis employs novel feature representation schemes tailored to individual modalities—text, audio, image, and video. Discriminative hand-crafted features, deep learning-based representations, and advanced deep learning techniques are utilized to extract meaningful and distinctive patterns from each modality. These methods are designed to enhance the separability of pain-related features while effectively handling the inherent challenges in feature computation, such as redundancy, noise, modality-specific variability, and overlapping representations.

4. To handle the challenges of data synchronization and multimodal integration by ensuring accurate temporal and contextual alignment across diverse physiological signal modalities—such as text, audio, image, and video—even under low-resource settings. This thesis acknowledges the complexities arising from heterogeneous data sources, sensor configurations, and acquisition environments and seeks to implement strategies that enable effective fusion. Again, the integration of high-dimensional multimodal data, employing advanced fusion techniques, will also facilitate real-time deployment and reliable performance in multimodal pain sentiment analysis systems.

1.6 Contributions of the Thesis

To address the aforementioned challenges and research objectives, this thesis proposes a series of pain sentiment analysis (PSA) models tailored to different data modalities. These models incorporate both unimodal and multimodal approaches,

designed to classify pain intensities accurately across various settings. The major contributions of this thesis are summarized as follows:

- A text-based PSA system is developed where various preprocessing techniques are applied to clean and standardize textual inputs. Both handcrafted and deep learning-based feature extraction techniques are explored to improve the distinctiveness of text-based cues. For both handcrafted and deep-learning based feature extraction some robust method of feature extarction techniques have been employed that addresses the challenges of noisy data, subjectivity in pain perception, and feature non-distinctiveness in text modality. The handcrafted features are classified using machine learning classifiers, whereas the deep learning techniques perform both feature extraction and classification of text-samples. (Publications C2, J1)

- An audio-based pain detection system is proposed by preprocessing speech data and extracting a combination of primary and secondary acoustic features—such as pitch, energy, MFCCs, and spectral descriptors. These features are modeled using machine learning classifiers, whereas the deep learning approaches integrates the handcrafted audio features to defined fully-connected network based classifiers to capture individual variations in vocal pain expression and improve classification performance across emotional and cultural contexts. (Publication: J4)

- The Image-based PSA models are introduced to detect pain from static facial visual cues. The models implement traditional handcrafted features, and some deep learning based convolutional neural network architectures including advanced techniques of feature learning and representations to identify visual expressions indicative of different pain levels. This contribution addresses challenges like facial variability, occlusion, and lighting inconsistencies in image-based PSA systems. (Publications: J1, J2, J3, C1)

- A video-based pain sentiment analysis system is designed to capture spatiotemporal variations in facial expressions and behavior using dynamic video sequences. Deep learning models are employed to extract temporal features from frame sequences, improving pain intensity classification while addressing frame-level inconsistencies, motion artifacts, and synchronization issues.

In this framework, novel *PainCapsule* models have been employed, which enhance the system's ability to model temporal dynamics and feature locality, making the PSA system more adaptive and accurate for pain detection in complex video sequences. (Publications: J5)

- A multimodal pain sentiment analysis framework is developed by integrating text, audio, and video data, where some fusion strategies are employed to synchronize and combine modality-specific features while minimizing redundancy. This approach enhances overall model performance and supports real-time deployment in resource-constrained environments. The proposed system demonstrates improved generalization and robustness in heterogeneous conditions by utilizing complementary strengths of multiple modalities. (Communicated)

- A comprehensive evaluation is conducted to analyze the usability, limitations, and discriminative capability of each individual modality. This analysis provides insights into the relative effectiveness of text, audio, image, and video cues in detecting pain intensities. The findings contribute to the design of adaptive models that can optimize modality selection based on available input conditions and application requirements.

This thesis as a whole adopts multimodal, multi-instance, and multi-algorithmic strategies to establish a robust framework for pain sentiment analysis. The proposed PSA systems offer meaningful contributions toward real-time, automated, and intelligent pain recognition, with potential applications in healthcare monitoring and affective computing.

1.7 Experimental Setup

This section details the experimental setup for implementing the proposed system, the databases used for testing, and the resulting performance analysis and discussion. The proposed Pain Sentiment Analysis System (PSAS) is implemented in Python and runs on a Windows 10 machine with 16GB RAM and an Intel Core i5 processor (3.20GHz). Various Python libraries are used, including NumPy, Keras [104], TensorFlow [105], and OpenCV. Specifically, TensorFlow and Keras implement the deep

learning based CNN architecture. The purpose of the research is to identify pain intensity levels by analyzing text-based sentiments, audio-based sentiments, image- or video-based sentiments (facial expressions) of a subject have been considered to differentiate pain intensity levels. The works in this thesis categorize sentiment into three/five distinct pain classes. Theoretically, pain is defined as: (i) hyperacusis, which is an increased sensitivity to common environmental noises that cause pain or discomfort even at low decibel levels; (ii) phonophobia, a severe fear of loud noises, often associated with physical discomfort and anxiety; and (iii) Tinnitus, a condition that causes the perception of phantom noises, such as ringing or buzzing, in the ears, though it is not always uncomfortable [106]. The acoustic elements of music, such as rhythm, speed, and harmony, have also been shown to play a role in pain relief and music therapy.

1.7.1 Performance Measure

Performance evaluation is one of the main important tasks that ensures the stability of any proposed system. The following are the important attributes based on which performance is measured.

1. Confusion Matrix: A confusion matrix [107] is used to summarize the performance of a machine learning model on test data. It displays the number of correct and incorrect predictions made by the model, and is particularly useful for evaluating classification models that predict categorical labels for input instances. The matrix consists of the following elements:

    - True Positive (TP): The model accurately predicted a positive outcome (actual outcome was positive).
    - True Negative (TN): The model accurately predicted a negative outcome (actual outcome was negative).
    - False Positive (FP): The model incorrectly predicted a positive outcome (actual outcome was negative), also known as a Type I error.
    - False Negative (FN): The model incorrectly predicted a negative outcome (actual outcome was positive), also known as a Type II error.

The confusion matrix plays a vital role in evaluating the performance of a classification model. It offers a comprehensive breakdown of the predictions of model specifically, true positives, true negatives, false positives, and false negatives. This detailed view helps in assessing key performance metrics such as recall, precision, accuracy, and the overall ability of the model to distinguish between classes. The confusion matrix is especially valuable when working with imbalanced datasets, as it provides deeper insights that go beyond standard accuracy measures.

2. Accuracy: This metric measures the performance of the model, defined as the ratio of total correct predictions to the total number of instances.

   $Accuracy = \frac{TP+FN}{TP+TN+FP+FN}$

3. Precision: Precision gauges the accuracy of the positive predictions of the model, calculated as the ratio of true positive predictions to the total number of positive predictions made.

   $Precision = \frac{TP}{TP+FP}$

4. Recall: Recall evaluates the ability of the model to identify all relevant instances, defined as the ratio of true positive (TP) instances to the sum of true positive and false negative (FN) instances.

   $Recall = \frac{TP}{TP+FN}$

5. F1-Score: The F1-score is a harmonic mean of precision and recall, used to assess the overall performance of a classification model.

   F1-Score=$2 \times \frac{Precision \times Recall}{Precision+Recall}$

6. Time Complexity: Time complexity [108] quantifies the computational time an algorithm takes to complete its task, based on the size of the input data. This is a critical metric in machine learning experiments to gauge the scalability and efficiency of models, especially with large datasets.

   - Training Time: This denotes the duration needed to train a machine learning model. Linear regression algorithms generally exhibit low training time complexity, whereas deep learning models, particularly those with extensive layers and neurons, have significantly higher complexity.

- Testing Time: This is the time required by a model to generate predictions on new data.

7. Space Complexity: Space complexity [109] refers to the memory requirements of an algorithm relative to the input data size. In machine learning experiments, this determines the memory needed to store data structures, model parameters, and intermediate computations.

   - Model Storage: This is the memory needed to store the trained model. Complex models, such as deep neural networks with millions of parameters, can demand substantial storage space [110].
   - Data Storage: This pertains to the memory necessary to store datasets used for training and testing. High-dimensional or large-scale data can present significant challenges in terms of memory usage [111].
   - Memory Usage During Training and Testing: Some algorithms, particularly in deep learning frameworks, may need additional memory for intermediate calculations.

   Though space complexity is an issue, at present scenario space becomes cheap so we ignore this feature from our analysis part.

1.8 Organization of the Thesis

The organizations of the thesis are as follows.

Chapter 2 proposes text-based pain sentiment analysis, the chapter starts with text preprocessing, then the preprocessed text data either handled by Hand-Crafted technique or Deep Learning to analyze the pain. (This work is published in C2, J1)

Chapter 3 presents the implementation of pain sentiment analysis using vocal signals in the form of audio data; the chapter begins with the audio preprocessing, then hand-crafted audio features are extracted. The hand-crafted features are then classified using traditional machine learning classifiers. Further, extracted audio features are processed and classified using fully connected networks. (This study appears in J4)

Chapter 4 introduces the analysis of pain using static visual behavioral pain signals in the form of image data. The chapter starts with the image preprocessing task and then features are extracted using hand-crafted and deep learning (CNN) techniques. The hand-crafted features are then classified using traditional machine learning classifiers, and the deep learning features are classified using fully connected layers. (Publication of this work occurred in J1, J2, J3, C1)

Chapter 5 demonstrates the pain sentiment analysis, which is carried out by examining dynamic behavioral visual pain signals, captured in the form of video data, the features are extracted from the sequence of video frames using various pre-trained model and for the classification purposes the fully connected layers and attention layers are implemented. (This article is in J5)

Chapter 6 discusses the concept of multimodal pain sentiment analysis, which is implemented by combining different types of data. (Communicated)

Chapter 7 provides the conclusions in which we have completed the results of the proposed works and highlights some directions for future research related to this thesis.

Chapter 2

Text-Based Pain Sentiment Analysis

In this chapter, we have developed two text-based pain sentiment analysis systems. This chapter talks about how healthcare communication is becoming more digital. In this digital age, text plays a vital role. It is now seen as a physiological signal. Texts can provide a lot of useful information about the health of a person. This includes his choice of words (lexical patterns) and the emotions he expresses. Compared to conventional biomedical signals that required specialized hardware for collection, text data can be easily obtained from individuals reporting their symptoms, and thus represents an easy and non-invasive source of initial healthcare evaluation [112]. Computerized sentiment analysis systems are usually multistage processing pipelines. The text goes through preprocessing with noise removal, tokenization, and normalization as the first stage. Feature extraction subsequently determines important sentiment indicators: BoW representations, TF-IDF representations, and word embeddings. Machine learning classifiers (such as LR, kNN, DT, RF, SVM) or deep neural networks (LSTMs, Transformers) do the feature extraction and classification under a single umbrella. Contemporary systems increasingly utilize transfer learning, where large corpus pre-trained models are adapted for particular sentiment analysis tasks and hence achieve higher performance on texts of a domain-specific nature [113].

This chapter comprehensively examines the approaches for text preprocessing, feature extraction, and classification of text-based pain-related sentiments, using both traditional machine learning methods and deep learning frameworks to obtain reliable and explainable results. The increased usage of digital media as a channel for health communication further highlighted the need for machine text analysis. Patients often communicate their discomfort, pain, and emotional states using textual language, in clinical surveys, online discussion groups, and even mobile apps for health [114]. These large quantities of unstructured textual data open avenues for training models with the potential to identify subtle feelings related to pain, which makes the possibility of immediate and personalized medical interventions highly promising [115]. Since pain is a personal experience that is usually difficult to measure using traditional diagnostic tools, natural language processing (NLP) tools developed through textual descriptions provide a promising means of achieving an objective evaluation. The text-based PSA system, PSA_{text} System, has been implemented in two ways, the first with hand-crafted features and machine learning classifiers, called HANDPSA_{text} System, and the second is deep learning techniques, which are called $\text{DLPSA}_{\text{text}}$ System for future reference.

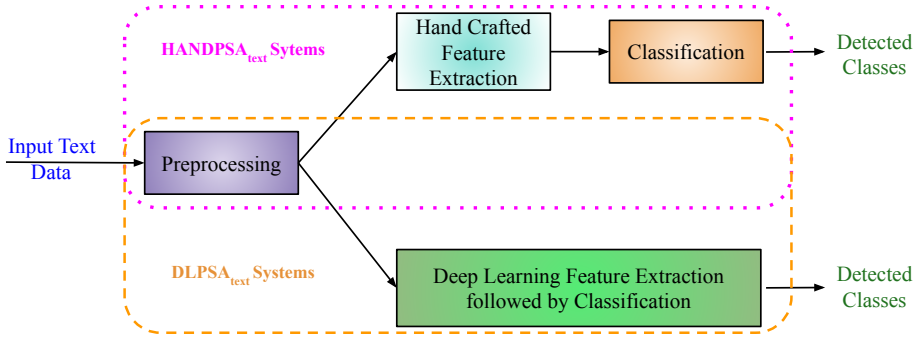


FIGURE 2.1: Workflow diagram of proposed PSA_{text} System.

Fig. 2.1 shows the flow diagram of the proposed PSA_{text} systems. A detailed examination of text preprocessing is conducted first, as it is necessary to transform raw, noisy data into a structured format for computational analysis. Next, we discuss feature extraction using hand-crafted techniques, where lexical characteristics are systematically encoded to identify pain-related problems [116]. Then, we explore traditional machine learning classifiers that employ the extracted features to classify pain sentiment in an interpretable and efficient manner. Later, in the second approach, we switch to deep learning-based methods. First, we do the text

preprocessing by the same method as in the first system, which eliminates feature extraction automation through hierarchical representations, thus minimizing manual feature extraction efforts. Neural networks such as LSTM [28] and BiLSTM [117] are tested for their ability to identify contextual pain signals from text. Lastly, we provide a comparative evaluation of these methods with state-of-the-art (SoA) methods, comparing their accuracy, generalizability, and computational cost. By synergizing traditional and novel approaches, this chapter attempts to create a robust model for pain sentiment classification, impacting the larger field of affective computing and healthcare analytics. Through this investigation, we highlight the importance of text-based pain analysis as a cost-efficient and scalable solution for contemporary healthcare, while also exploring challenges and potential avenues for improving these computational models in real-world clinical use.

The organization of this chapter is as follows. Section 2.1 discusses literature reviews related to PSA_{text} system. The proposed systems are presented in Section 2.2. Section 2.3 presents and analyzes the experimental results. Finally, Section 2.4 concludes the chapter.

2.1 Literature Review

Recent studies have increasingly focused on identifying and distinguishing various forms of harmful online behavior, such as hate speech, vulgarity, and aggression. For example, Kumar et al. [118] explored the distinction between hate speech and vulgarity, offering insights into how these can be further classified into covert and overt aggression. The work on cyberbullying detection began with the foundational research by Dinakar et al. [119]. Based on this, Dadvar et al. [120, 121] and Van Hee et al. [122] developed more refined systems with improved performance in detecting bullying behaviour online. Trolling, another toxic online behaviour, has also garnered research interest. Initial efforts by Cambria et al. [123] were followed by Kumar et al. [124], Mihaylov et al. [125], and Mojica [126], who worked on improving the accuracy of trolling detection systems. Racism, a critical issue in social discourse, has also been addressed using sentiment analysis techniques. Greevy and Smeaton [127] proposed early solutions for identifying racist language in textual

data. The problem of detecting hate speech has also been studied. Notable contributions include those of Burnap and Williams [128], Djuric et al. [129], Gitari et al. [130], and Badjatiya et al. [131]. Regarding abusive language detection, Chen et al. [132] proposed a system based on Lexical Syntactic Features, whereas Nobata et al. [133] adopted machine learning algorithms to identify abusive content across web platforms more effectively.

Deep learning algorithms can illuminate internal dimmed patterns in complex datasets, which play a key role in both feature extraction and classification [134]. Dashtipur et al. introduced a Multimodal Framework based on context awareness on Persian Sentiment Analysis [135]. Based on movie review comments on emotion analysis, Sagum et al. [136] have proposed a sentiment measurement technique. Rustam et al. [137] compared the performance of various supervised machine learning methodologies based on the Sentiment Analysis of COVID-19 tweets. The pain or anguish experienced when exposed to sound is called acoustic or auditory pain. Recent advances in text-based sentiment analysis have seen transformer models such as BERT (Devlin et al.) [7] and GPT-3 (Brown et al.) [138] achieve state-of-the-art performance by capturing deep contextual relationships. Researchers have developed specialized techniques for social media analysis (Zimbra et al.) [139], multilingual applications (Conneau et al.) [140], and handling informal language (Barbieri et al.) [141]. Aspect-based sentiment analysis has emerged as a crucial subfield (Pontiki et al.) [37], while new approaches address challenges such as sarcasm detection (Cai et al.) [142] and emotion recognition (Bostan and Klinger) [143]. Explainable AI methods (Ribeiro et al.) [144] have been applied to improve model interpretability, and domain adaptation techniques improve performance in specialized contexts. Recent work also focuses on bias mitigation (Blodgett et al.) [145] and low-resource language support, reflecting the expanding scope and ethical considerations of the field.

Recent progress in sentiment analysis has been made in specific areas, such as detecting pain-related emotions. In this case, text is used to understand if someone is in physical or emotional distress. Al-Hassan and Al-Dossari [146] created a hybrid deep learning model that could detect pain-related emotions in clinical texts. Their model worked well with data from medical forums. Ji et al. [147] used BERT and trained it to recognize the intensity of pain in the text written by the patients.

This showed that transformer models can be very effective in this field. Sarsam et al. [148] built a special pain-related word list and used it with SVM. This approach was added to earlier methods such as the one by Gitari et al. [130], which focused on hate speech. For analyzing pain using more than one type of data, Gratch et al. [149] combined text with sound features. This idea followed a similar multimodal approach by Dashtipour et al. [135]. In social networks, Sharma et al. [150] created a method to find pain-related tweets without labeled data, similar to how Zimbra et al. [139] analyzed the content of the social network. To handle multiple languages, López-Úbeda et al. [151] adapted XLM-R for Spanish texts about pain. Their work was based on the multilingual model of Conneau et al. [140]. Ethical issues in labeling pain data were discussed by Deerwester et al. [152], related to bias concerns raised by Blodgett et al. [145]. Lastly, Peng et al. [153] used explainable AI to make pain prediction results easier to understand, following methods such as LIME from Ribeiro et al. [144].

Cheng et al. [154] employed NLP-based sentiment analysis on Healthgrades reviews to quantitatively characterize patient perceptions and satisfaction in pain management services. Nunes et al. [155] provided a comprehensive systematic review of computational approaches, including sentiment analysis, used to model and interpret pain-related language in clinical and patient-generated texts. Aggarwal et al. [156] compared classical and deep learning sentiment models (VADER, BERT, Flair) on patient reviews to assess their effectiveness in capturing sentiment related to pain management care. Ghosh et al. [157] proposes a capsule-inspired deep model that jointly encodes textual facial patterns and semantic cues to classify fine-grained pain sentiment levels. Fang et al. [158] provides a comprehensive survey of machine-learning methods for pain detection across modalities, including text-based and sentiment-oriented approaches. Fernandez et al. [159] describes the fifth AI4Pain challenge, outlining tasks, datasets, and baselines for automated pain assessment across multiple data types.

2.2 Proposed PSA_{text} Systems

In this work, text-based two PSA systems have been proposed. The text documents contain several comments, each of which contains several sentences. Now each sentence contains several words. During sentiment analysis, text datasets may have noise due to the diverse domains [118]. The proposed systems have three components: (i) text preprocessing, (ii) feature extraction, and (iii) classification. As shown earlier (see Fig. 2.1), a common text preprocessing step is designed for both systems. In the next subsection, we describe the text preprocessing step. After text preprocessing, feature extraction and classification are two fundamental steps to analyze pain. The features are one of the most important ingredients of this classification; the performance of the classifier is highly dependent on the features used, as they represent abstract patterns. After text preprocessing, in $\text{HANDPSA}_{\text{text}}$ System, the feature extraction and classification are executed sequentially; on the other hand, in $\text{DLPSA}_{\text{text}}$ System, the feature extraction and classification are done under a single umbrella.

2.2.1 Text Preprocessing

Text preprocessing is the first phase of PSA_{text} System. The proposed text preprocessing has four steps, which are discussed below.

Text Cleaning: Text cleaning is the first and most important step in preparing raw text. Removes things that are not useful, such as HTML tags, website links, emojis, or any broken characters. These can clutter the data and confuse the model. All letters are changed to lowercase so that similar words like 'The' and 'the' are treated the same. Punctuations and special symbols are also removed to reduce noise. Regular expressions help in finding and removing patterns like email addresses or phone numbers. Common spelling mistakes and contractions have been fixed to make the text more consistent. Extra spaces and repeated letters are cleaned up.

Repeated letters refer to cases where a letter appears more times than necessary, usually because of informal typing, emphasis, or errors. For example, people may write:

- "soooo good" instead of "so good".
- "cooool" instead of "cool".
- "happppy" instead of "happy".

Cleaning repeated letters is necessary because NLP models treat each spelling variation as a different word, so leaving exaggerated forms like "soooo" or "cooool" increases noise, makes the vocabulary unnecessarily large, and prevents the model from recognizing that these words actually mean the same thing; by reducing such repetitions to their standard form, the text becomes more consistent, easier to process, and helps the model learn clearer and more meaningful patterns. After cleaning, the text becomes much easier to work with and provides a solid base for further processing in NLP tasks.

Tokenization: Tokenization is the process of breaking the cleaned text into smaller parts. These parts are called tokens and can be words, phrases, or even smaller units such as subwords. Most of the time, the text is split using spaces or punctuation marks. However, more advanced tools can handle tricky cases like hyphenated words (e.g., 'well-being') or words with apostrophes (e.g., 'can't'). Sometimes, the text is broken down into sentences using clues such as capital letters and periods. In special cases, like with rare or complex words, tokenization splits them into meaningful parts. Languages such as Chinese or German have unique rules, and specific tokenizers are designed for them. In general, tokenization turns unstructured text into parts that machines can easily understand and analyze.

It has been observed that text preprocessing is not fully robust even after text cleaning is complete, tokenization still uses elements like hyphens, apostrophes, and capital letters because cleaning only removes unnecessary noise; it does not remove meaningful structures inside valid words or sentences. Tokenizers rely on these features to correctly split the text. For example, a hyphen in "well-being" or an apostrophe in "can't" is preserved during cleaning because they change the meaning of the word, so the tokenizer uses them to decide

whether the word should stay whole or be split into smaller parts. Similarly, capital letters are kept in sentence boundaries, helping the tokenizer recognize where one sentence ends and another begins. In this way, cleaning prepares the text by removing irrelevant symbols, while tokenization uses the remaining meaningful punctuation and structure to break the text into accurate, analyzable units.

Stop Word Removal: Stop word removal helps simplify text by taking out very common words like 'the', 'is', or 'and'. These words usually do not carry much meaning, especially when the goal is to understand the main message. Removing them reduces the amount of data and speeds up processing. However, in some cases, certain common words may still be important. For example, in sentiment analysis, words like 'not' can change the meaning completely. That is why custom stop word lists are used depending on the domain. Instead of removing all stop words, some modern methods just reduce their importance. This step helps the system focus more on the meaningful parts of the text.

Stemming: Stemming cuts words down to their basic form. For example, 'running', 'runs', and 'runner' are all reduced to 'run'. This is done using rule-based methods that remove common word endings. Porter stemmer is a popular tool for this, which applies a series of steps to strip suffixes. Although this method is fast, it does not always give real words and can sometimes mix up unrelated words that look similar. A lighter version of stemming is used for certain languages like Arabic, where only common endings are removed. Even though stemming is not as precise as other methods like lemmatization, it is still useful when speed is more important than perfect accuracy.

The step-by-step result of text preprocessing, as an example, is shown in Fig. 2.2.

2.2.2 HANDPSA$_{\text{text}}$ System

In this system, two well-known techniques, Bag of Words (BoW) and Term Frequency Inverse Document Frequency (TF-IDF), have been used to compute the features from the preprocessed tests. The extracted features are then classified using some machine learning classifiers such as kNN, RF, SVM, LR, and DT.

FIGURE 2.2: Steps with result of the text preprocessing of PSA_{text} System.

2.2.2.1 Feature Extraction

Both BoW and TF-IDF are dictionary-based approaches to represent a document. So, a global dictionary is available and with respect to this dictionary, any given comment will be represented by some numeric values. First, we discuss the basic concept of the dictionary and how it is designed.

Let D be the corpus (collection of documents) and assume that there are N comments (in the form of small documents) in the corpus $D = \{d_1, d_2, \ldots, d_N\}$. d_i is then tokenized into several words. Tokenized comment d_i may contain several stop words [160] such as 'is', 'are', 'i am', 'would', 'will', 'what is', 'more', 'such', 'has', 'have', etc. During processing, these stop words are removed from d_i so that after preprocessing, d_i is transformed into d_i'.

The preprocessed corpus $D' = \{d_1', d_2', \ldots, d_N'\}$ contains words relevant to assessing the aggressiveness of an individual. Since a word may have multiple meanings, converting words into numbers is crucial while preserving distinguishing characteristics [161]. In addition, numerical transformation enhances classifier performance. The collection of all unique words from D' has been considered, and for the desired dictionary, $Dict = \{w_1, w_2, \ldots, w_M\}$ where M is the number of words in the dictionary $Dict$.

Bag of Words (BoW): Let a preprocessed comment c_i' have 't' tokens, i.e., $c_i' = w_1'||w_2'||\cdots||w_{t-1}'||w_t'$ and this comment is represented as a vector $\mathbf{Vc}_i'$ with

respect to the dictionary *Dict*. $\mathbf{Vc}'_i$ is defined as

$$\mathbf{Vc}'_i = (n_1, n_2, \cdots, n_M) \tag{2.1}$$

where n_i represents the number of occurrences of $w_i \in Dict$ in c'_i. The steps of the BoW method are shown in Fig. 2.3.

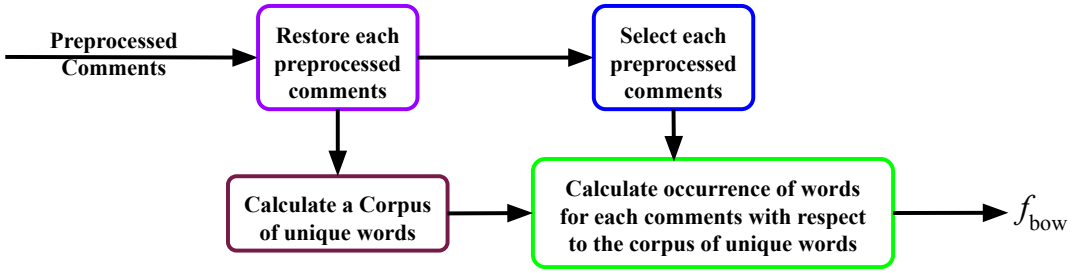


FIGURE 2.3: Block diagram of BoW based feature extraction method.

Example 1: Let us have a dictionary $Dict = \{\text{cat}, \text{dog}, \text{log}, \text{mat}, \text{on}, \text{sat}, \text{the}\}$, $M = 7$.

Input comments:

1. d_1: *"The cat sat on the mat."*
2. d_2: *"The dog sat on the log."*

Preprocessing:

- Lowercase conversion and punctuation removal
  1. $d'_1 \rightarrow \{\textit{the cat sat on the mat}\}$
  2. $d'_2 \rightarrow \{\textit{the dog sat on the log}\}$
- Tokenization
  1. $c'_1 \rightarrow \{\text{the, cat, sat, on, the, mat}\}$
  2. $c'_2 \rightarrow \{\text{the, dog, sat, on, the, log}\}$
- Stop word removal
  1. $c'_1 \rightarrow \{\text{cat, sat, mat}\}$
  2. $c'_2 \rightarrow \{\text{dog, sat, log}\}$

- Stemming
  1. $c'_1 \rightarrow$ {cat, sit, mat}
  2. $c'_2 \rightarrow$ {dog, sit, log}

Representation of the comments:

1. $Vc'_1 = (\underbrace{1}_{\text{freq of 'cat'}} , \underbrace{0}_{\text{freq of 'dog'}} , \underbrace{0}_{\text{freq of 'log'}} , \underbrace{1}_{\text{freq of 'mat'}} , \underbrace{1}_{\text{freq of 'sit'}})$
2. $Vc'_2 = (\underbrace{0}_{\text{freq of 'cat'}} , \underbrace{1}_{\text{freq of 'dog'}} , \underbrace{1}_{\text{freq of 'log'}} , \underbrace{0}_{\text{freq of 'mat'}} , \underbrace{1}_{\text{freq of 'sit'}})$

The final feature vector using the BoW method for the comments d_1 and d_2 is Vc'_1 and Vc'_2, respectively. For simplicity, the feature vector computed using the BoW method is denoted by f_{bow}.

Term Frequency Inverse Document Frequency (TF-IDF): Let c' be a pre-processed comment that has 't' tokens. The term frequency (TF) of a word $w \in c'$ is defined as

$$\text{TF}(w, c') = \frac{\text{frequency of } w \text{ in } c'}{\text{number of tokens in } c'} \tag{2.2}$$

The inverse document frequency (IDF) of a token 'w' can be defined as

$$\text{IDF}(w, D) = log(\frac{\text{Number of documents in } Dict}{\text{number of documents containing } w}) \tag{2.3}$$

Finally, TF-IDF of a token of a comment is computed as

$$\text{TF-IDF}(w, c', D) = \text{TF}(w, c') \times \text{IDF}(w, D) \tag{2.4}$$

The steps of the TF-IDF method are shown in Fig. 2.4.

Example 2: Compute the TF, IDF, and TF-IDF values of the tokens from comments as given in **Example 1**.

Term frequency (TF) value of the tokens.

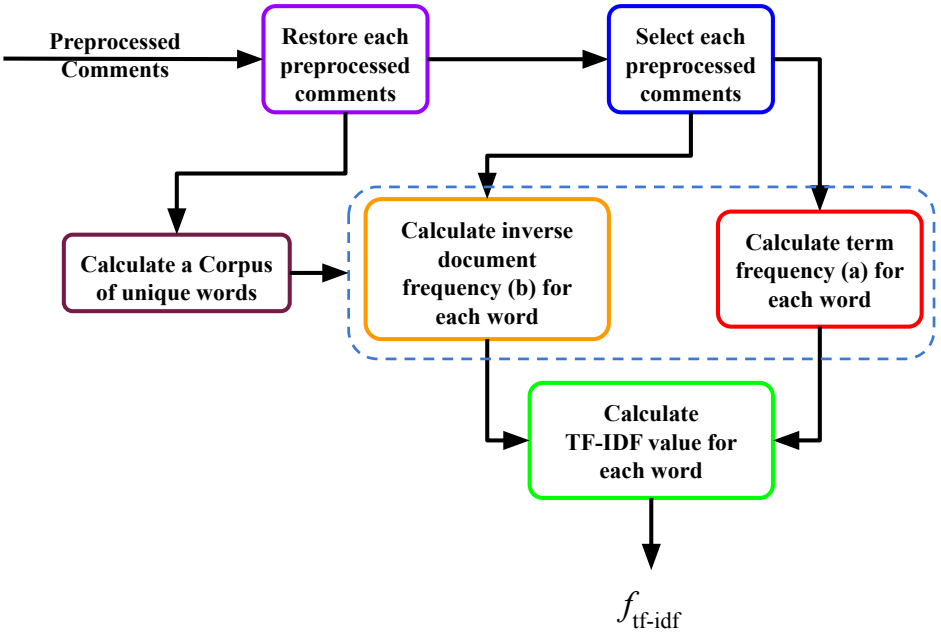


FIGURE 2.4: Block diagram to compute TF-IDF features.

	cat	dog	log	mat	sit
c_1'	$\frac{1}{3}$	0	0	$\frac{1}{3}$	$\frac{1}{3}$
c_2'	0	$\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{1}{3}$

IDF value of the tokens are as follows.

$$\begin{aligned}
\text{IDF}(\text{cat}) &= \log(2/1) = \log(2) \approx 0.693 \\
\text{IDF}(\text{dog}) &= \log(2/1) = \log(2) \approx 0.693 \\
\text{IDF}(\text{log}) &= \log(2/1) = \log(2) \approx 0.693 \\
\text{IDF}(\text{mat}) &= \log(2/1) = \log(2) \approx 0.693 \\
\text{IDF}(\text{sit}) &= \log(2/2) = \log(1) = 0
\end{aligned}$$

TF-IDF score of the tokens

	cat	dog	log	mat	sit
c'_1	$\frac{1}{3} \times 0.693$	0	0	$\frac{1}{3} \times 0.693$	0
c'_2	0	$\frac{1}{3} \times 0.693$	$\frac{1}{3} \times 0.693$	0	0

2.2.2.2 Classification

In the classification stage, the feature vectors f_{bow} and $f_{\text{tf-idf}}$ are used separately to train the classifiers, kNN, RF, SVM, LR, and DT (described in Chapter 1). After the training process is completed, the performance of the models is evaluated using the test vector. Here, we handle 3-class and 5-class pain problems.

2.2.3 DLPSA$_{\text{text}}$ Systems

Traditional hand-crafted features like BoW and TF-IDF have fundamental limitations: Both BoW and TF-IDF give importance to the frequency of individual tokens. Both methods produce fixed-size, sparse vectors that lack word order and fail to capture semantic or contextual relationships between words in sentences. These semantic relationships are crucial for achieving better classification performance, as they preserve meaningful linguistic patterns. To address this, deep learning approaches employ embedding layers that transform words into dense vector representations and encode the semantic meaning. These semantic features are combined with the LSTM [102] and BiLSTM [117] layers separately as two different architectures, both of which process text sequentially rather than as unordered data representations such as BoW and TF-IDF, handling variable-length inputs through adaptive hidden states. LSTMs capture long-term dependencies in a forward direction, and BiLSTMs extend this by analyzing relationships bidirectionally, enabling comprehensive context understanding, a capability entirely absent in hand-crafted features. Such long-term word dependencies are essential for accurately interpreting context, meaning, and sentiment in textual data.

LSTM [102] and BiLSTM are a specific type of Recurrent Neural Network (RNN), which is applied for sequence labeling and sequence prediction tasks. They basically overcome the architectural weakness of RNN [162]. The feature extraction using

this scheme is as follows: each preprocessed comment d_i' undergoes space-separated sequences of words, which are further split into a list of tokens, and then these tokens are vectorized. This list of tokens is finally input to the LSTM and BiLSTM that performs feature learning of tokens from the respective comment and classifies the comment into either three or five different pain classes, depending on the datasets. The simple LSTM is capable of processing the sequences in the forward direction only; on the other hand, the BiLSTM processes the sequences in both forward and backward directions.

The LSTM model starts with an input layer of size 200. Next, it passes through an Embedding layer that turns the input into a 200×100 matrix, using around 1 million trainable parameters. After that, an LSTM layer with 100 units processes this data, adding about 80,400 more parameters. The output then goes to a Dense layer with 128 units and ReLU activation, which has around 12,928 parameters. Finally, there is another Dense layer with 3 units, using sigmoid activation, that produces the final 3-value output. Altogether, the model has approximately 1,093,715 trainable parameters. The LSTM-based architecture is described in Fig. 2.5, while the parameters are shown in Table 2.1.

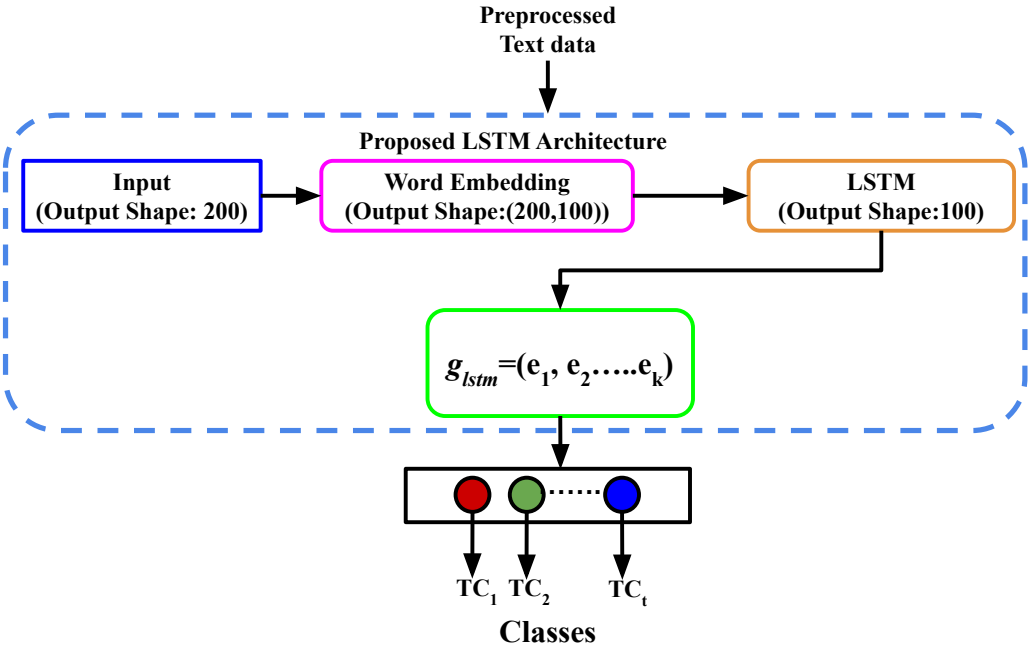


FIGURE 2.5: Block diagram of LSTM based $DLPSA_{text}$ System ($DLPSA_{textlstm}$ System).

TABLE 2.1: Parameter details of the used LSTM architecture.

Layer	**Output Shape**	**Parameter**
Input	200	0
Embedding	(200,100)	1000000
LSTM	100	80400
Dense	128	12928
Activation (ReLu)	3	0
Dense	3	387
Activation (sigmoid)	3	0
Total	**1093715**	

The BiLSTM model begins with an input layer of size 50. This is followed by an Embedding layer that converts the input into a 50×128 matrix, using about 1,280,000 parameters. Then, a Bidirectional layer processes the sequence and gives a 1×128 output, adding around 98,816 parameters. After that, Batch Normalization is applied, using 512 parameters. A Dense layer then reduces the output to 1×32, with 4,128 parameters. Following a dropout layer (referred to as DOUT), there is a final Dense layer that produces a 1×5 output using 165 parameters. In total, the model has 1,383,621 parameters, of which 1,383,365 are trainable, and 256 are non-trainable, likely coming from the Batch Normalization layer. The BiLSTM-based architecture is presented in Fig. 2.6, and the parameters are listed in Table 2.2.

TABLE 2.2: Parameter details of the used BiLSTM architecture.

Layer	Output Shape	Parameter
Input	50	0
Embedding	(50,128)	1280000
Bidirectional	(1,128)	$(1 + 771) \times 128$=98816
BatchNormalization	(50,128)	4×128=512
Dense	(1,32)	$(1 + 128) \times 32$=4128
DOUT	(1,64)	0
Dense	(1,5)	$(32+1) \times 5$=165
Total Parameters		1383621
Total Trainable Parameters:		1383365
Non-trainable Parameters:		256

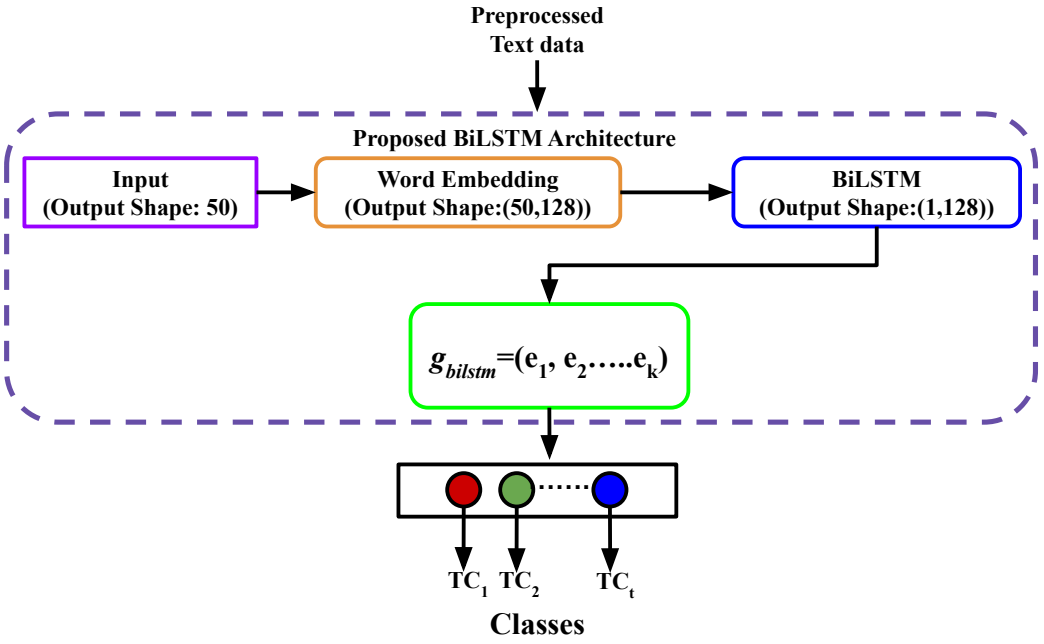


FIGURE 2.6: Block diagram of BiLSTM based $DLPSA_{text}$ System ($DLPSA_{textbilstm}$ System).

Comments	Class
Most of Private Banks ATM's Like HDFC, ICICI etc are out of cash. Only Public sector bank's ATM working	TC_1
Wondering why Educated Ambassador is struggling to pay through Credit/Debit at a Decent Restaurant! Cant imagine that diplomat of a Developed nation is not having a Card and he needs Cash only for Dinner.	TC_2
again the same thing in eyes of judiciary he is not terrorist he is accused in our eyes is terrorist and he deserved to killed who ever keep a bad eye on our nation he are she should be killed without no mercy	TC_3

FIGURE 2.7: Some samples of TD_{aggr} (3-Class Problem).

Hence, by employing both LSTM and BiLSTM the obtained feature vector size is $N \times L$, where N is the total number of samples for a particular dataset and L is the dimension of the embedding operation here for both LSTM and BiLSTM the value of L is 100 and 128 respectively.

2.3 Experiments and Results

In this section, we present a detailed discussion of the experiments and analysis of the results for both systems. For evaluation purposes, we used three datasets.

The first text dataset is an Aggression Dataset (TD_{aggr}) [118] which has three categories, such as Covertly Aggressive (CAG), Overtly Aggressive (OAG), and Non-Aggressive (NAG) classes. It has been mentioned that the OAG class of comments basically represents those comments where the aggression of the user is expressed with a great amount of intensity against any particular topic. For these types of comments, both the external and internal statements are highly aggressive in nature. For the CAG class of comments, the intensity of the overall aggressiveness is quite low with respect to OAG comments. Moreover, if we notice the external statement of the comments, it may not look aggressive at all, but if we notice the internal statement of the comments, the clear aggressiveness will be identified distinctly. In the case of the NAG class of comments, no aggressiveness can be identified from either the external or internal statements of the comments. We consider OAG as TC_3, CAG as TC_2 and NAG as TC_1 classes. Some sample comments and the corresponding class label are depicted in Fig. 2.7. Practically, this dataset is not a pain dataset; due to the scarcity, in our work, we consider this dataset and assume that NAG indicates 'No Pain' (PI_0), CAG represents 'Low Pain' (PI_1), and OAG denotes 'High Pain' (PI_2). The mapping between the actual class label of the original dataset and the assumed plain label is shown in Table 2.3.

TABLE 2.3: Description of TD_{aggr} Samples.

Pain	**Class**	**Samples**
PI_0	TC_1	4240
PI_1	TC_2	2708
PI_2	TC_3	5052

The second dataset used is Amazon Fine Food Reviews (TD_{amazon}). In this dataset, users have provided feedback and rated food quality on a five-point scale, indicating that the dataset includes reviews with text classes ranging from 1 to 5. Like in the previous, this dataset has no relation to pain. We consider this in our work where we consider the rank of the original data as the pain level, i.e., the higher the rank of the data, the higher the pain level. Some sample comments and the corresponding

class label are depicted in Fig. 2.8. The level correspondence between the actual rank and the pain level is given in Table 2.4.

Comments	Ratings
Product arrived labeled as Jumbo Salted Peanuts...the peanuts were actually small sized unsalted. Not sure if this was an error or if the vendor intended to represent the product as "Jumbo".	TC_1
If you are looking for the secret ingredient in Robitussin I believe I have found it. I got this in addition to the Root Beer Extract I ordered (which was good) and made some cherry soda. The flavor is very medicinal.	TC_2
The flavors are good. However, I do not see any differce between this and Oaker Oats brand - they are both mushy.	TC_3
good flavor! these came securely packed... they were fresh and delicious! i love these Twizzlers!	TC_4
This offer is a great price and a great taste, thanks Amazon for selling this product. Staral	TC_5

FIGURE 2.8: Some samples of TD_{amazon} (5-Class Problem).

TABLE 2.4: Detailed Description of TD_{amazon}.

Pain	Class	Samples
PI_0	TC_1	52268
PI_1	TC_2	29769
PI_2	TC_3	42640
PI_3	TC_4	80655
PI_4	TC_5	52268

The third dataset is our own curated dataset (TD_{llm}), which mainly includes comments reflecting the intensity of pain of individuals. These pain intensities that have a scale from 0 to 4 and are mentioned as Pain Intensity-0 (PI_0) implies No Pain, Pain Intensity-1 (PI_1) indicates very low pain, Pain Intensity-2 (PI_2) for low pain, Pain Intensity-3 (PI_3) represents moderate pain, Pain Intensity-4 (PI_4) implies high pain. The comments are collected from people from various age groups through personal interviews. Additional details about this dataset are provided in Table 2.5, and some sample comments and their label are shown in Fig. 2.9.

During the implementation of PSA_{text} System, each comment from all the above datasets is classified into three or five categories, depending on the dataset. Initially,

TABLE 2.5: Detailed Description of TD_{llm}.

Pain	Class	Samples
PI_0	TC_1	200
PI_1	TC_2	200
PI_2	TC_3	200
PI_3	TC_4	200
PI_4	TC_5	200

Comments	Class
I feel completely fine today. There is no pain or discomfort in my body. I can move around freely without any issues.	TC_1
I feel a slight discomfort in my lower back. It's not too bad, but it's noticeable. I can still go about my day without much trouble.	TC_2
I have a moderate pain in my lower back. It's more noticeable than before, but I can still manage. I might need to take a break and rest for a bit.	TC_3
I have a severe pain in my lower back. It's making it difficult to sit or stand for long periods. I need to lie down and rest frequently.	TC_4
I have an excruciating pain in my lower back. It's unbearable and makes it impossible to move. I need immediate medical attention.	TC_5

FIGURE 2.9: Some samples of TD_{llm} (5-Class Problem).

TABLE 2.6: Feature size of the datasets TD_{aggr}, TD_{amazon}, and TD_{llm}.

Dataset	No. of comments	Feature Size
TD_{aggr}	12000	$R^{12000\times1000}$
TD_{amazon}	568454	$R^{568454\times1000}$
TD_{llm}	1000	$R^{1000\times1000}$

every comment d_i is preprocessed in d_i' using the technique described in Section 2.2.2. The proposed feature extraction methods of $HANDPSA_{text}$ System (i.e., BoW and TF-IDF) return a feature vector of size 1×1000. The size of the different datasets is different; therefore, the size of the feature matrix (taking into account all comments) is different. The summary of the feature space for the different datasets is given in Table 2.6.

These feature matrices are then randomly divided, with 50% assigned for training and the other 50% for testing purposes to evaluate the model performance. For

the experiments using HANDPSA$_{\text{text}}$ System, two types of feature vectors are considered: BoW model (f_{bow}) and the TF-IDF ($f_{\text{tf-idf}}$) model. The training dataset was used to build classification models using various machine learning algorithms: kNN, RF, SVM, LR, and DT. These classifiers were chosen for their complementary strengths in text classification tasks. RF is a fast, probabilistic generative model that performs well on sparse and high-dimensional data, such as TF-IDF, because of its simplifying independence assumptions. LR, a discriminative model, provides interpretable coefficients and calibrated probability estimates, making it suitable for both the baseline evaluation and feature importance analysis. SVM focuses on maximizing the geometric margin between classes and offers flexibility through kernel functions, making it robust for high-dimensional and nonlinear problems. DT delivers clear, rule-based decisions, improving interpretability for domain experts. Lastly, kNN is a non-parametric, instance-based algorithm that relies on local patterns and does not assume linear separability. Together, these models span diverse paradigms, such as generative, discriminative, geometric, rule-based, and non-parametric, allowing for a comprehensive evaluation of algorithmic behavior without adding the complexity of advanced techniques like ensembles or deep learning. After training, each classifier generated a model, and these models were evaluated using the test dataset to assess the performance of the proposed methodology. The results are presented in Table 2.7, which shows the classification performance for both the 3-class and 5-class problems.

TABLE 2.7: Performance of the proposed HANDPSA$_{\text{text}}$ System with f_{bow} and $f_{\text{tf-idf}}$ features.

Bag of Words Feature							**TF-IDF Feature**					
3 Class Problem (TD$_{\text{aggr}}$)							**3 Class Problem (TD$_{\text{aggr}}$)**					
Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Training Time**	**Testing Time**	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Training Time**	**Testing Time**
LR	30.51	33.29	29.72	33.72	5.13	0.03	32.78	35.35	32.14	36.24	5.14	0.03
kNN	53.38	52.84	52.61	53.16	5.14	0.02	52.95	57.17	52.26	57.73	5.12	0.04
DT	52.74	53.26	52.09	54.08	5.08	0.02	52.71	56.72	51.83	57.21	5.14	0.03
RF	54.22	54.19	53.36	54.83	5.09	0.02	54.33	57.48	53.74	58.33	5.13	0.02
SVM	56.82	56.14	55.83	57.16	5.11	0.04	58.56	59.67	87.83	60.52	5.13	0.03
5 Class Problem (TD$_{\text{amazon}}$)							**5 Class Problem (TD$_{\text{amazon}}$)**					
Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Training Time**	**Testing Time**	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Training Time**	**Testing Time**
LR	63.13	62.42	62.49	63.29	6.13	0.06	66.27	65.72	65.31	66.46	6.22	0.06
kNN	63.19	61.72	62.56	62.37	6.11	0.06	64.57	64.23	63.72	65.37	6.07	0.05
DT	60.41	59.36	59.75	61.06	6.17	0.05	67.18	67.54	66.84	68.26	6.21	0.07
RF	62.28	60.71	61.38	61.43	6.14	0.05	67.63	65.29	66.68	66.14	6.18	0.06
SVM	64.87	64.29	63.96	65.51	6.18	0.06	68.21	68.57	67.59	69.37	6.17	0.07
5 Class Problem (TD$_{\text{llm}}$)							**5 Class Problem (TD$_{\text{llm}}$)**					
Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Training Time**	**Testing Time**	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Training Time**	**Testing Time**
LR	74.25	73.47	73.53	74.18	4.26	0.03	77.26	77.64	76.39	78.46	4.31	0.04
kNN	73.84	74.58	72.87	75.31	4.18	0.04	76.35	77.17	75.53	77.82	4.28	0.04
DT	72.69	72.12	72.03	73.64	4.21	0.03	76.81	75.24	75.16	76.31	4.33	0.03
RF	74.77	73.34	73.78	74.27	4.27	0.04	77.17	76.52	76.28	77.27	4.31	0.02
SVM	77.21	76.32	76.62	77.36	4.24	0.02	78.62	78.29	77.34	79.25	4.29	0.03

From Table 2.7, it has been observed that the accuracy score obtained with the TF-IDF features is better than that achieved using the BoW features. Both feature types have been used separately on various machine learning classifiers. The experiments have demonstrated that for the 3-class classification problem, SVM has produced the best performance, achieving an accuracy of 59.67% in the Aggression dataset. The same classifier has outperformed other traditional models in the 5-class classification task, producing an accuracy of 66.57% for the Amazon Fine Food Reviews dataset and 78.29% for the Own Pain dataset. So in this chapter, on the basis of experimental results as a traditional feature extraction technique, the TF-IDF methodology is considered along with the SVM classifier as the HANDPSA$_{\mathbf{text}}$ System.

In DLPSA$_{\text{text}}$ System, the same 50% training-testing split has been maintained. The performance of DLPSA$_{\text{text}}$ System across datasets has been reported in Table 2.8.

TABLE 2.8: Performance of the proposed system using DLPSA$_{\text{text}}$ systems along with training and testing times in *Sec.*

LSTM						
Dataset	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Training Time**	**Testing Time**
TD$_{\text{aggr}}$	57.91	58.03	57.15	58.26	6.21	0.03
TD$_{\text{amazon}}$	67.84	67.26	66.92	68.07	7.17	0.11
TD$_{\text{llm}}$	76.37	75.62	75.63	76.34	4.22	0.02
BiLSTM						
	Accuracy	**F1-Score**	**Precision**	**Recall**	**Training Time**	**Testing Time**
TD$_{\text{aggr}}$	68.59	68.34	67.71	69.52	7.09	0.04
TD$_{\text{amazon}}$	73.22	74.77	72.58	75.31	8.12	0.13
TD$_{\text{llm}}$	87.39	86.32	86.24	87.22	5.08	0.02

The experimental results presented in Table 2.8 have clearly demonstrated that the BiLSTM architecture has been able to capture better semantic relationships compared to LSTM. Consequently, better accuracy scores have been obtained in all datasets using the BiLSTM technique relative to the LSTM approach. For this reason, the BiLSTM technique has been selected as the feature extractor and classifier for this chapter. Henceforth, DLPSA$_{\text{text}}$ System refers to a BiLSTM-based deep model.

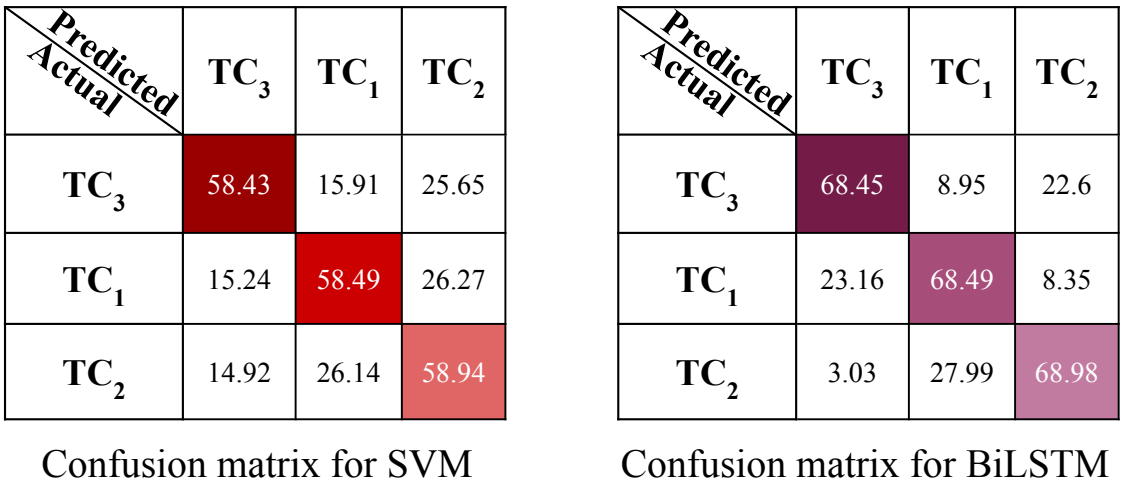


FIGURE 2.10: The confusion matrix in percentage using $\text{HANDPSA}_{\text{text}}$ System and $\text{DLPSA}_{\text{text}}$ System with TD_{aggr} dataset for 3-class classification.

From Table 2.7 and Table 2.8, it is noticeable that the proposed system achieves a better result using SVM with respect to other machine learning models and with LSTM models for both *3-class* and *5-class* classification problems. LSTM model basically able to stress upon some important factors such as sequential text processing, represents words contextually, thus the contextual sense is mapped as feature, it also takes care of words having semantic similarities, but with statistical approaches such as BoW and TF-IDF the sentences are considered as unordered set of words these methods are unable to preserve neither contextual sense nor semantic relationship among the words within their feature maps and as a result of that better performance is obtained with BiLSTM. The conventional LSTM learns input sequences in one direction only (past $\rightarrow$ future), i.e., its features are learned only from past contexts. Bidirectional LSTM (BiLSTM), on the other hand, learns sequences in both directions (past $\rightleftarrows$ future). This two-way learning ability is the reason that $\text{DLPSA}_{\text{text}}$ System exhibits better feature extraction.

The side-by-side confusion matrices shown in Fig. 2.10 compare the performance of the SVM and BiLSTM models in the 3-class classification task (TC_1, TC_2, TC_3). The SVM model demonstrates moderate classification ability, with notable misclassifications between categories, particularly for TC_3 (1476 correct predictions against 402 misclassified as TC_1 and 648 as TC_2). In contrast, the BiLSTM model shows superior performance across all classes, significantly improving correct predictions

Predicted / Actual	TC_1	TC_2	TC_3	TC_4	TC_5
TC_1	59.68	12.38	13.37	8.67	5.89
TC_2	15.01	65.72	10.05	5.11	4.11
TC_3	11.62	6.51	61.44	14.04	6.38
TC_4	9.40	4.32	7.34	69.76	9.18
TC_5	6.07	2.72	4.14	5.38	81.68

Confusion matrix for SVM

Predicted / Actual	TC_1	TC_2	TC_3	TC_4	TC_5
TC_1	70.33	8.07	8.66	8.59	4.35
TC_2	10.91	74.21	5.87	5.18	3.84
TC_3	5.00	8.83	73.19	5.87	7.10
TC_4	4.70	5.33	9.26	71.13	9.59
TC_5	4.50	4.81	5.66	4.13	80.90

Confusion matrix for BiLSTM

FIGURE 2.11: The confusion matrix in percentage using $\text{HANDPSA}_{\text{text}}$ System and $\text{DLPSA}_{\text{text}}$ System with $\text{TD}_{\text{amazon}}$ for 5-class classification.

for TC_3 (1729 vs. 1476 of SVM), TC_1 (1452 vs. 1240), and TC_2 (934 vs. 798), while substantially reducing cross-category confusion, particularly between TC_1 and TC_2 (177 misclassifications vs. 557 of SVM) and between TC_3 and TC_1 (226 vs. 402). This comparative analysis highlights a stronger contextual understanding in distinguishing the classes for the BiLSTM model.

Fig. 2.11 shows a comparison between two confusion matrices generated for a five-class classification task (classes TC_1 to TC_5) in $\text{TD}_{\text{amazon}}$. Each matrix shows how well the predictions of the model match the actual classes. The numbers along the diagonal (such as 9782 for TC_2 in SVM and 11044 for TC_2 in BiLSTM) represent correct predictions. The SVM model makes more errors, often confusing TC_3 with TC_1 and TC_4. In contrast, the BiLSTM model performs better, with stronger diagonal values (for example, 15604 for TC_3), which means it gets more predictions right. Mistakes are still present, especially for TC_4 and TC_5, which both models found to be more difficult to classify. Overall, the comparison shows that BiLSTM is more accurate for this task.

Fig. 2.12 shows how two models perform a classification task with five categories (TC_1 to TC_5) in TD_{llm}. Both models do well, as most predictions fall along the diagonal, which means that they correctly match the actual classes. However, BiLSTM does a little better overall. It has more correct predictions (like 87 for TC_1

Predicted / Actual	TC_1	TC_2	TC_3	TC_4	TC_5
TC_1	81.00	5.00	4.00	4.00	6.00
TC_2	5.00	79.00	5.00	4.00	7.00
TC_3	6.00	5.00	77.00	5.00	7.00
TC_4	5.00	5.00	7.00	77.00	6.00
TC_5	5.00	5.00	5.00	5.00	80.00

Confusion matrix for SVM

Predicted / Actual	TC_1	TC_2	TC_3	TC_4	TC_5
TC_1	87.00	5.00	3.00	2.00	3.00
TC_2	3.00	87.00	3.00	4.00	3.00
TC_3	2.00	5.00	88.00	2.00	3.00
TC_4	2.00	4.00	2.00	89.00	3.00
TC_5	2.00	4.00	5.00	3.00	86.00

Confusion matrix for BiLSTM

FIGURE 2.12: The confusion matrix in percentage using $\text{HANDPSA}_{\text{text}}$ System and $\text{DLPSA}_{\text{text}}$ System with TD_{llm} for 5-class classification.

compared to 81 in SVM) and makes fewer mistakes (its off-diagonal numbers are lower). The SVM tends to confuse the nearby classes TC_3 and TC_4, while the errors of BiLSTM are spread more evenly, showing that it handles the task more reliably. In short, BiLSTM gives better results than SVM in this case.

The results in Fig. 2.13 show that the performance of the model is better with more epochs. The figure suggests that 200 can be considered a good choice for the epochs. For the configurations tested, a batch size of 50 produces the best results. We chose a batch size of 50 with 200 training epochs for parameter tuning in our suggested BiLSTM architecture based on these observations.

The performance of the proposed systems has been compared with some existing SoA methods. The comparative study is reported in Table 2.9. The analysis reveals that the proposed systems for sentiment analysis outperform the existing methods. Among these, the BiLSTM technique demonstrates the most superior performance. Consequently, for the purposes of this chapter, the proposed BiLSTM approach has been selected as the primary methodology.

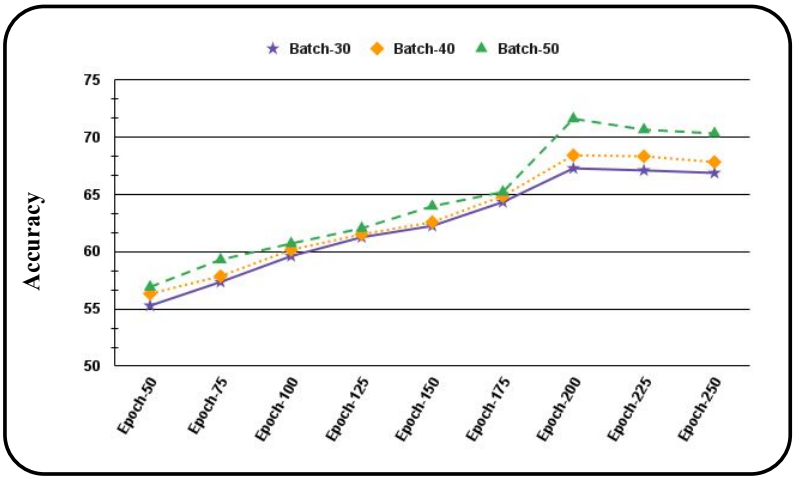


FIGURE 2.13: Effect of batch vs. epoch for the proposed BiLSTM Sentiment Analysis system for 3-class (with TD_{aggr}) classification.

TABLE 2.9: Comparison of Performance of the proposed system with the other competing methods for the 3-class (with TD_{aggr}) and 5-class (with TD_{amazon}) classification problems.

Method	Accuracy (%)	F1-Score
3-Class Problem		
Samphabadi et al. [163]	56.17	0.5875
Kumar et al. [164]	33.72	0.3572
Modha et al. [165]	53.76	0.5583
Constantin Orasan [166]	55.79	0.5832
$HANDPSA_{text}$ System	58.56	0.5967
$DLPSA_{text}$ System	68.59	0.6834
5-Class Problem		
Pang & Lee [2]	58.22	0.5524
Wang et al. [167]	62.17	0.6036
Kim et al. [168]	59.56	0.5743
$HANDPSA_{text}$ System	68.31	0.6692
$DLPSA_{text}$ System	73.59	0.7267

2.4 Conclusions

This chapter presents the text-based pain sentiment analysis system, denoted as PSA_{text}. The system is implemented through three primary steps: (i) text preprocessing, (ii) feature extraction, and (iii) classification. Two distinct implementation strategies are explored. The first, $HANDPSA_{text}$ System, follows a traditional pipeline in which feature extraction and classification are performed sequentially.

The second, $\text{DLPSA}_{\text{text}}$ System, adopts a deep learning-based approach, integrating feature extraction and classification within a unified framework. The performance of both approaches is evaluated using three benchmark datasets: the Aggression dataset (TD_{aggr}), Amazon Fine Food Reviews ($\text{TD}_{\text{amazon}}$), and a custom-curated dataset (TD_{llm}). Extensive experiments and comparative analysis against SoA methods reveal that $\text{DLPSA}_{\text{text}}$ System consistently outperforms its counterparts, including $\text{HANDPSA}_{\text{text}}$ System. Among all the techniques designed in this chapter, the BiLSTM-based $\text{DLPSA}_{\text{text}}$ System demonstrates superior performance. This variant is selected as the representative model for PSA_{text}.

Text-based pain sentiment provides a special view by capturing the personal pain description of individuals, allowing a subjective and in-depth record of their pain experience. Patients can report their pain in their own words through this method, perhaps eliciting some aspects missed by structured questioning. However, text-based analysis of pain descriptions poses challenges because of the extreme range of how people describe their pain. Some might describe it with rich metaphors, while others will downplay their symptoms, with this rendering generalization of findings or the creation of standardized interpretations problematic. Other challenges include cultural and linguistic variations in the way pain is expressed, and some words or phrases may have particular meanings within particular societies, restricting the applicability of text-based models. So, there may be one drawback of the PSA_{text} system that patients who cannot write cannot express their pain through text. Due to this, it becomes difficult to accurately assess the pain level in these patients using only the PSA_{text} system. Another notable limitation of the PSA_{text} system may be the inability to pick up nonverbal cues of pain, which tend to be important sources of information. For example, a patient may report that their pain is 'manageable' in writing, but the pitch volume is high. Without these points, the evaluation may not be as comprehensive and could result in an underestimation or misunderstanding of the severity of the pain. Considering that audio signal data may be able to alleviate these constraints while preserving the advantages of textual data, in the next chapter, audio-based pain sentiment analysis has been proposed to resolve the issues of textual-based pain sentiment analysis.

Chapter 3

Audio-Based Pain Sentiment Analysis

In this chapter, audio signals have been used as vocal physiological signals in pain sentiment analysis to explore pain-related emotions, overcoming the major limitations of the use of PSA_{text} system. Text-based data is a plain signal, which is emotionless. Vocal signals are rich sources of emotion, picking up subtle changes in pitch, speech rhythm, and instability in the voice that are omitted or hard to detect when written. In contrast to text, which can be confining for illiteracy, speech disabilities, or physical paralysis, audio data represents a more spontaneous and accessible way of expressing the intensity of pain and affective states. This makes it especially useful in healthcare and assistive technology, where the conventional PSA_{text} system will not capture all the nuances of human expression. The approach followed in this chapter is one of a controlled pipeline, from preprocessing to implementation of the PSA_{audio} system. The same working flow (as shown in Fig. 1.2) is also followed for PSA_{audio} System. The implications of the finding go beyond pain assessment and have potential for applications in monitoring mental health, human-computer interaction, and assistive technology for patients with communication impairments. Through this research, we underscore the importance of audio information as a strong medium to interpret human emotions beyond what textual information can accomplish.

This chapter introduces a systematic method to overcome certain disadvantages in pain sentiment analysis by moving the focus from text data to audio signals for better evaluation of human responses. The study starts with the collection of audio data from legitimate channels and then preprocessing the signals to remove the noise, and for this purpose a Kalman filter is used. Then, three sets of hand-crafted features are derived: i) statistical features [73], which measure temporal and spectral changes; ii) Mel-Frequency Cepstral Coefficients (MFCCs) [11], capturing the spectral properties of speech; and iii) spectral features [74], examining frequency domain patterns. Subsequently, these features are tested using two different classification methods. First, conventional machine learning models are applied separately to every set of features to assess their discriminative power. We call this class of system as HANDPSA$_{\text{audio}}$ System. It is to be noted that these types of features are the fundamental features for most of the audio processing, and for this reason, in our second approach, we combine all the hand-crafted features and process through a Fully Connected Network (FCN) to investigate whether the combination of multiple acoustic descriptors enhances classification performance. In this work, we refer to our second approach as DLPSA$_{\text{audio}}$ System for further reference.

This chapter is structured to guide the reader through the development and evaluation of the proposed approach. Section 3.1 discusses existing research works available in the literature. Section 3.2 introduces the methodology for audio-based pain sentiment analysis along with the various classification techniques used in the system. Section 3.3 presents the experimental setup and provides a detailed analysis of the results. Finally, Section 3.4 wraps up the chapter with key conclusions and insights.

3.1 Literature Review

A number of pioneering studies have explored the use of audio signals for pain recognition. Giordano et al. [72] introduced an early and influential approach focused on identifying acoustic pain indicators in neonatal populations by analyzing audio features. Building on this foundation, Oshrat et al. [169] proposed an innovative method to investigate the relationship between self-reported pain intensity and measurable bioacoustic markers in human vocalizations, with a particular focus on

prosody. In a related effort, Ren et al. [170] developed an automated system to assess pain using paralinguistic features in speech, with the aim of enhancing both the objectivity and the reliability of pain diagnosis. Expanding the scope further, Zeng et al. [171] conducted a psychological survey on how human emotions are perceived. Most recently, Hong et al. [172] proposed an improved method for assessing the pain severity, including pain localization as a secondary task to refine overall diagnostic accuracy.

Audio-based pain sentiment analysis is becoming a powerful, non-invasive way to understand and measure how much pain someone is in by listening to their voice. Researchers found early on that pain changes how people sound, like altering pitch, jitter, and shimmer in their speech [173]. Schuller et al. [174] used features like MFCCs and speech rhythm to build machine learning models that detect pain with higher accuracy than earlier methods. The INTERSPEECH ComParE challenge [175] helped standardize how pain detection from audio is tested. Deep learning moved the field forward, with Han et al. [176] creating a model that combined CNNs and RNNs to outperform traditional approaches. Lefter et al. [177] showed that adding spoken words to the audio analysis improved results, while Gratch et al. [178] discovered that certain vocal pain cues work across different cultures. Privacy-focused techniques like federated learning [179] made it possible to train models without risking patient data. Other studies looked into how chaotic patterns in voice [180], real-time monitoring systems [181], and the difference in vocal patterns between chronic and acute pain [182] can be used to improve analysis. Data limitations were addressed with synthetic audio [183], and multilingual systems [184] made it possible to detect pain in many languages. Hammal et al. [185] found that things like speech rate and pauses are strong indicators of pain. Personalized models [186] helped tailor predictions to individual voice traits. The ComParE 2020 Sub-challenge [187] introduced deep spectrum features to raise the performance bar. Researchers also linked pain-related voice changes to body functions like the nervous system [188], while semi-supervised models [189] reduced the need for lots of labeled data. Lopez-Otero et al. [190] found voice changes after medication, and neural architecture search [191] helped design better models automatically. INTERSPEECH 2021 included pain as a topic [192], which pushed researchers to build even better features. Yao et al. [193] used contrastive learning to make models stronger against poor recording conditions. Vocal formant frequencies [194] also showed promise for detecting pain, and Xu et

al. [195] improved federated learning for broader collaboration. Competitions like the IEEE Signal Processing Cup 2022 [196] focused on pain recognition, pushing for real-time solutions. Self-supervised pretraining [197] made use of unlabeled audio data, and microphone differences were tackled with compensation algorithms [198]. Multiscale analysis [199] helped models understand both quick and slow changes in voice, while the INTERSPEECH 2023 challenge [200] introduced transformer-based models that further raised the performance standard. Transformer-based models have taken audio pain analysis to the next level. For example, Liang et al. [201] used self-attention to understand long-range patterns in pain-related speech, achieving top results. Chen et al. [202] worked on making these models easier to understand by visualizing which parts of the audio matter most. Park et al. [203] found that things like age and gender can impact how accurately pain is detected from voice. Wang et al. [204] introduced new audio features specifically designed to pick up pain distortions. The INTERSPEECH 2022 challenge [205] pushed things forward by including more detailed pain intensity ratings. Zhang et al. [206] explored few-shot learning to make progress even when data is limited. Johnson et al. [207] looked at how pain sounds differ across medical conditions, and Smith et al. [208] built lightweight models that could run on devices in hospitals. Anderson et al. [209] confirmed that using voice to monitor pain after surgery works well in real-world trials. Lee et al. [210] suggested combining audio with medical text records to get better pain predictions. A special issue in IEEE Transactions on Affective Computing [211] summarized many of these breakthroughs. Brown et al. [212] tackled noisy hospital audio and made it cleaner for analysis. Taylor et al. [213] studied how psychology influences pain sounds. Wilson et al. [214] helped hospitals collaborate on model training without sharing sensitive data. Garcia et al. [215] showed that culture affects how pain is expressed vocally. Roberts et al. [216] used dynamic time warping to match pain sounds across different voices. A Journal of Voice special issue [217] focused on how this research is used in clinics. Harris et al. [218] created models that can tell both how intense the pain is and what kind of pain it is. Martin et al. [219] tested voice assistants for letting patients report their pain and found they are well accepted. Thompson et al. [220] used graph neural networks to model complex relationships in pain speech, and the INTERSPEECH 2023 challenge [221] centered entirely on recognizing medical pain. Adams et al. [222] used active learning to save time when labeling big pain datasets. Green et al. [223] found that voice pain

cues often match up with physical stress signals, and Clark et al. [224] developed methods to analyze pain while keeping the identity of general people confidential. Finally, a special section in the IEEE Journal of Biomedical and Health Informatics [225] collected a number of studies on using voice as a health signal, including pain detection.

Borna et al. updated of the voice-based pain detection review summarizes [226] newer machine-learning approaches that derive vocal biomarkers from speech to improve automated identification of pain in adults, building directly on the original survey work. Early work within the TAME Pain project [227] uses advanced audio analysis pipelines and rigorous validation protocols to develop trustworthy, reproducible models for estimating pain from speech as part of a broader multimodal assessment framework. PainNOVA Poster [228] introduces a privacy-aware framework for assessing pain levels from voice signals while minimizing exposure of lexical content. Ghosh et al. [229] integrates an audio-based subnetwork into a multimodal IoT framework to improve pain sentiment recognition performance.

3.2 Proposed $\text{PSA}_{\text{audio}}$ Systems

A $\text{PSA}_{\text{audio}}$ system works through a systematic and multistep process to achieve optimal classification accuracy. Raw audio signals are first processed to improve signal quality by noise reduction, normalization, and filtering to remove distortions and inconsistencies. The system then extracts three different acoustic features that are essential for pain sentiment analysis: Statistical features [73] to extract quantitative signal properties, Mel-Frequency Cepstral Coefficients (MFCCs) [11] to represent vocal properties and the spectral envelope, and spectral features [74] to describe frequency domain behavior. In the first approach, each of these sets of features is subsequently classified separately employing conventional machine learning classifiers like SVM, RF, and LR to make initial pain sentiment predictions based on individual feature performance. As mentioned earlier, this system is referred to as $\text{HANDPSA}_{\text{audio}}$ System. In the second approach, we fuse the extracted feature sets into a single high-dimensional representation that captures their complementary strengths. This fused feature vector is then input into an FCN, which uses

deep learning to identify complex, nonlinear relationships within the data, significantly improving classification accuracy over single feature-based models. Through systematic preprocessing, extraction of varied acoustic features, performing preliminary classifications, feature fusion, and ANN-based refinement, the system provides strong and accurate pain sentiment analysis, rendering it very effective for use in healthcare, patient monitoring, and affective computing applications. This system is called DLPSA$_{\text{audio}}$ System. The general model for pain recognition using audio data is shown in Fig. 3.1.

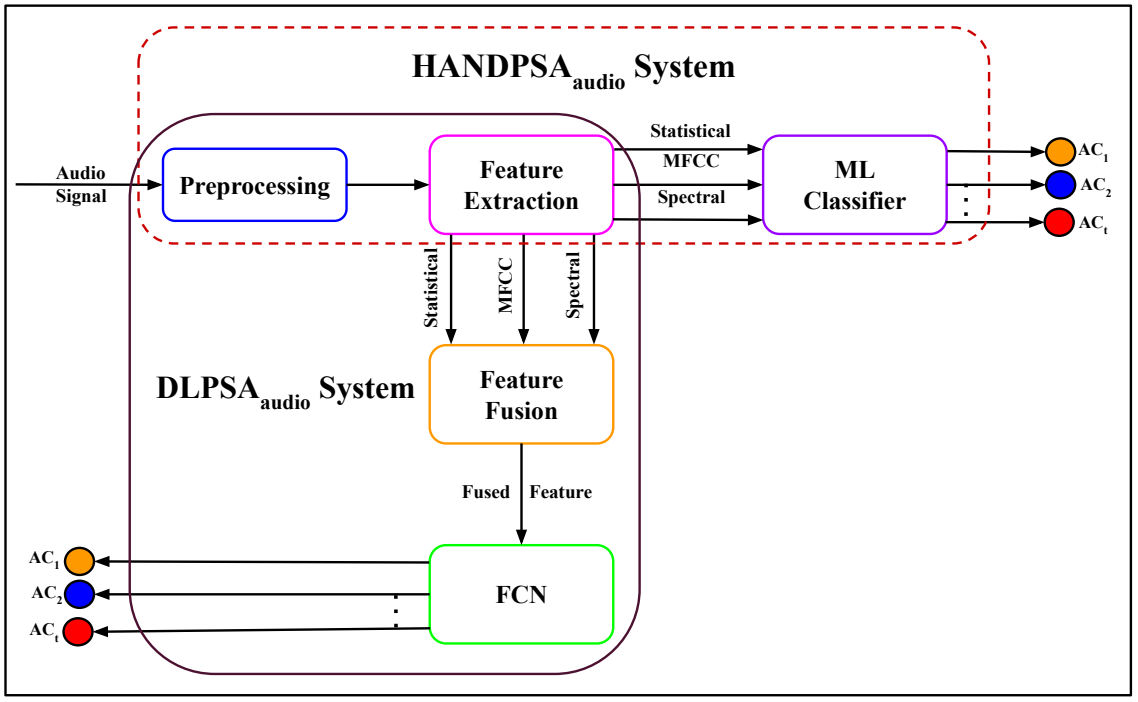


FIGURE 3.1: Demonstration of proposed PSA$_{\text{audio}}$ Systems.

3.2.1 Audio Preprocessing

The proposed methodology uses vocal audio signals, which often contain noise and artifacts; therefore, preprocessing of audio data is highly essential. In the preprocessing phase, to ensure the same sampling rate for all the audio samples, the resampling method is used. For the improvement of vocalization, it is very important to apply different noise reduction techniques so that noises such as speech and artifacts attached to the audio signal can be accurately removed. In preprocessing, the first

task is to apply sampling, where the amplitude of the audio signal is measured at regular intervals. The sampling process is described by Eq. (3.1). Following sampling, the next preprocessing step is quantization. In this step, each sampled value is assigned to the nearest digital level, as defined by Eq. (3.2). The audio signal is represented by Eq. (3.3), which summarizes the overall digitization process of the audio signal. As the audio signal may contain noise, the next preprocessing step is to apply the Kalman filter shown in Eq. (3.4). The Kalman filter works recursively to reduce errors as much as possible. It helps to clearly pick out the main audio signal by removing the background noise. A comprehensive overview of the entire audio preprocessing pipeline is described in the following and is illustrated in Fig. 3.2.

1. Sampling (Discretization in Time) The signal $x(t)$ is sampled at intervals $T = \frac{1}{f_s}$, where f_s is the sampling frequency. The **sampled signal** $x[n]$ is:

$$x[n] = x(nT) \quad \text{where} \quad n \in \mathbb{Z} \text{ (integer indices)}, \quad T = \frac{1}{f_s} \tag{3.1}$$

n is an integer that counts the number of samples in the discrete-time sequence.

2. Quantization (Discretization in Amplitude) The sampled signal $x[n]$ has continuous amplitude values. **Quantization** maps these to a finite set of levels (e.g., 16-bit audio has $2^{16} = 65{,}536$ levels). Let:

- $Q(\cdot)$ = quantization function
- $\Delta = \frac{\text{Dynamic Range}}{2^B}$ = step size, where B = bits per sample

$$x_q[n] = Q\left(x[n]\right) = \Delta \cdot \text{round}\left(\frac{x[n]}{\Delta}\right) \tag{3.2}$$

Quantization Error: $e[n] = x_q[n] - x[n]$ (treated as noise).

3. Final Digital Signal Representation The **digital signal** is fully described by:

$$x_d[n] = x_q[n] = Q\left(x(nT)\right) \tag{3.3}$$

4. Applying Kalman filter to the digitized audio signal

$$y_d[n] = K(x_d[n]) \tag{3.4}$$

where:

- $x_d[n]$ = digitized audio signal
- $K(\cdot)$ = Kalman filter operation
- $y_d[n]$ = filtered (noise-removed) digitized audio signal

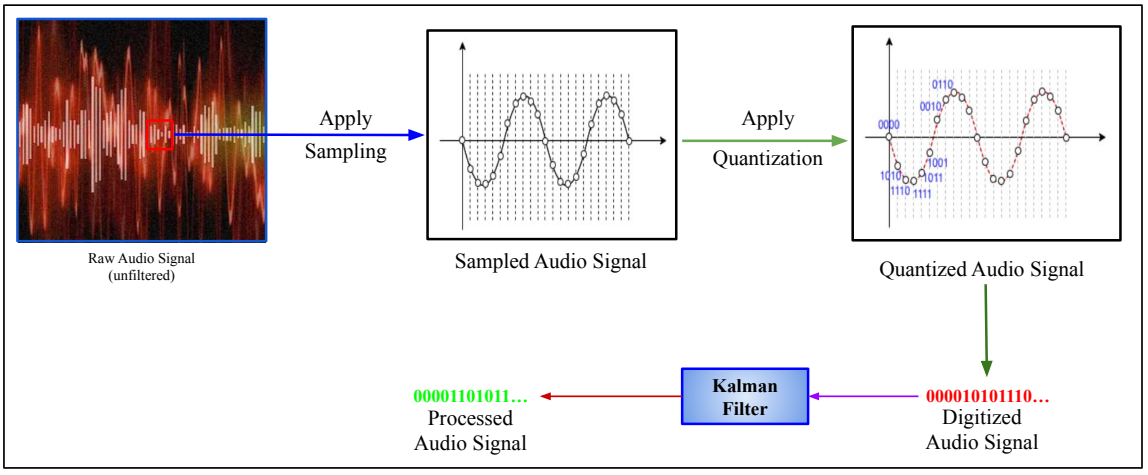


FIGURE 3.2: Steps of audio preprocessing.

3.2.2 HANDPSA$_{\text{audio}}$ System

This work uses speaker audio recordings for audio-based pain sentiment evaluation. The audio data is provided in waveform format, consisting of a sequence of bytes representing audio signals over time. Hence, analyzing these sequences of bytes, feature extraction techniques are employed for feature computation from the audio files. These feature extraction techniques are described below.

- **Statistical audio features:** Here, the audio signal initially undergoes a frequency domain signal analysis using fast Fourier transformation (FFT) [73].

$$D[p] = \sum_{n=0}^{N-1} y_d[n].e^{-j\frac{2\pi}{N}pn} \tag{3.5}$$

The computed frequencies are subsequently employed to calculate descriptive statistics, including mean, median, standard deviation, quartiles, and kurtosis. The magnitude of frequency components is used to calculate Energy and the Root Mean Square (RMS) value. For the calculation of the energy of a frame the Eq. (3.6) is used, and for the RMS value, Eq. (3.7) is used. The energy of a frame of R samples:

$$E = \sum_{p=0}^{N-1} |D[p]|^2 \tag{3.6}$$

where $D[p]$ frequency domain transformed audio signal

The Root Mean Square (RMS) value:

$$\text{RMS} = \sqrt{\frac{1}{N}\sum_{p=0}^{N-1} |D[p]|^2} \tag{3.7}$$

- **MFCC:** This technique, considered a widely used method for audio-based feature computation [11], begins by applying a log-amplitude transformation to the frequency components derived from the FFT. The Mel scale is then applied to the logarithmic amplitude spectrum. Afterward, a Discrete Cosine Transform (DCT) is performed on the Mel-scaled spectrum, and typically only the 2nd to 13th DCT coefficients are retained for feature computation, and the remaining coefficients are discarded. The process begins with applying the Discrete Fourier Transform (DFT) to a frame of the filtered and digitized audio signal $y_d[n]$, as shown in Eq. (3.5). Next, Mel filterbanks and logarithm are then applied to the power spectrum, as described in Eq. (3.8).

$$S[m] = \log\left(\sum_{p} |D[p]|^2.H_m[p]\right) \tag{3.8}$$

Here, $S[m]$ denotes the Log-Mel Spectrum and $H_m[p]$ represents the m^{th} Mel filter. Finally, MFCCs are calculated by applying the DCT to the Log-Mel Spectrum, as given in Eq. (3.9).

$$\text{MFCC}[l] = \sum_{m=0}^{M-1} S[m] \cos\left(\frac{\pi l}{M}(m+0.5)\right) \tag{3.9}$$

In this equation, $l = 0, 1, \ldots, L-1$ denotes the index of the MFCC coefficients, M is the total number of Mel filterbanks used, and m runs from 0 to $(M-1)$.

- **Spectral Features:** There are many types of spectral features available, but the spectral centroid was chosen because it gave better results compared to the others. The spectral centroid is derived by Eq. (3.11). These spectral features [74] are related to the spectrogram of those audio files. The spectrogram represents the frequency intensities over time. It is measured from the squared magnitude of the STFT [230], which is obtained by computing the FFT over successive signal frames. The frequency domain representation $D[p]$ is obtained by applying DFT to the filtered and digitized audio signal $\mathbf{y_d}[\mathbf{n}]$, $f[p]$ is the frequency (in Hz) of the p^{th} bin in the DFT and $f[p]$ is obtained using Eq. (3.10).

$$f[p] = \frac{p \cdot f_s}{R} \tag{3.10}$$

where f_s is the sampling rate (in Hz) and the spectral centroid (Centroid) is computed as in Eq. (3.11).

$$\text{Centroid} = \frac{\sum_{p=0}^{R-1} f[p]\,|D[p]|}{\sum_{p=0}^{R-1} |D[p]|} \tag{3.11}$$

These feature extraction techniques derive d_1-dimensional statistical features (f_{stat}), d_2-dimensional MFCC features (f_{mfcc}), and d_3-dimensional f_{spec} features from each audio file. Then, these extracted feature vectors f_{stat}, f_{mfcc}, and f_{spec} are considered here as the input feature vector for HANDPSA$_{\text{audio}}$ System. Now, the individual feature vector undergoes the classifiers employed (discussed in Chapter 1) to build the pain detection model for the proposed HANDPSA$_{\text{audio}}$ System.

3.2.3 Classification

During analysis, the above extracted features and the combinations of these feature sets are classified using machine learning classifiers such as DT, LR, kNN, and SVM, with 50:50 and 75:25 training-testing splits to classify the features. These feature sets, both individually and in combination, are also employed for $\text{DLPSA}_{\text{audio}}$ System.

3.2.4 $\text{DLPSA}_{\text{audio}}$ System

In this model, the above three extracted features, statistical, MFCC, and spectral, are combined, as mentioned above, and then passed through three fully connected (FC) layers before reaching the final dense output layer. The three FC layers progressively refine the combined features by learning hierarchical and discriminative representations. The first FC layer plays a key role in processing the raw features. It is enhanced with batch normalization, ReLU activation, and dropout. Batch normalization helps stabilize and speed up training by normalizing the inputs. ReLU adds nonlinearity, allowing the model to learn more complex patterns. Dropout helps prevent overfitting by randomly disabling some neurons during training. This enhancement allows the FC layer to capture important initial interactions in the data while maintaining generalization. The second FC layer further abstracts these features, emphasizing meaningful patterns and suppressing noise through another nonlinear transformation. The third FC layer compresses the refined features into a compact, class-separable representation, optimizing them for the final SoftMax classifier. These three layers improve feature discriminability, stabilize training, and improve generalization, enabling more accurate audio classification. Fig. 3.3 demonstrated the FCN architecture in terms of deep feature representation for audio features.

These extracted feature vectors are combined so that $f_{audio} = \langle f_{\text{stat}}, f_{\text{mfcc}}, f_{\text{spec}} \rangle$. Now, $f_{audio} \in \mathbf{R}^{1\times d}$, $d = d_1 + d_2 + d_3$, undergoes the proposed deep learning-based feature representation followed by classification. During feature representation using the dense layers of a fully connected network (which contains three fully connected layers), a d-dimensional input feature vector is first transformed into 512 dimensions,

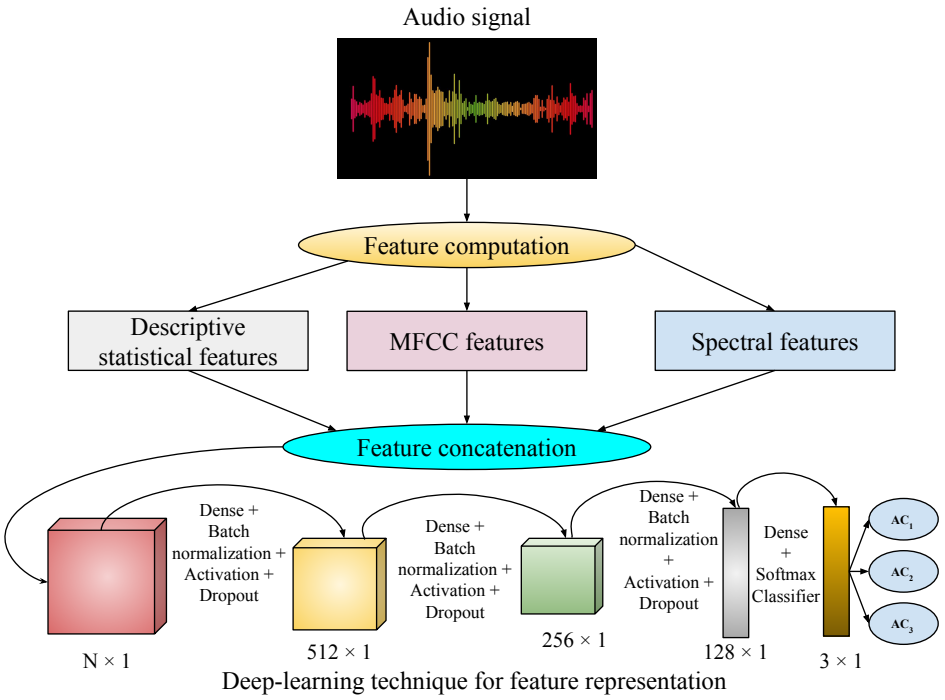


FIGURE 3.3: Proposed $\text{DLPSA}_{\text{audio}}$ System.

then reduced to 256, and finally to 128 dimensions. This specific arrangement is designed to take advantage of the progressive reduction of dimensionality, allowing the network to extract and refine high-level features in a hierarchical manner. The initial layer with 512 units captures a wide range of complex patterns from the input. As features pass through the 256 and 128 unit layers, they are compressed into more compact and discriminative representations. This gradual reduction not only helps prevent overfitting by limiting the number of parameters but also preserves essential information. Striking a balance between model capacity and generalization, enabling the network to learn effectively without becoming overly complex. Moreover, this structured compression supports stable gradient flow during backpropagation, which improves training efficiency and leads to better overall performance on the task. Finally, during classifications, the feature is mapped into 3 or 5-class audio-pain problems. The list of parameters required for audio-based pain detection using the deep learning technique is demonstrated in Table 3.1.

TABLE 3.1: List of parameters for $DLPSA_{audio}$ System.

Layer	**Output-shape**	**Feature-size**	**Parameter**
Flatten	(1, d)	(1,d)	0
$Block_1$			
Dense	(1, 512)	(1, 512)	(1 + d)×512
BatNorm	(1, 512)	(1, 512)	2048
ActRelu	(1, 512)	(1, 512)	0
Dropout	(1, 512)	(1, 512)	0
$Block_2$			
Dense	(1, 256)	(1, 256)	(1 + 512)×256 = 131328
BatNorm	(1, 256)	(1, 256)	1024
ActRelu	(1, 256)	(1, 256)	0
Dropout	(1, 256)	(1, 256)	0
$Block_3$			
Dense	(1, 128)	(1,128)	(1 + 256)×128 = 32896
BatNorm	(1, 128)	(1,128)	512
ActRelu	(1, 128)	(1,128)	0
Dropout	(1, 128)	(1,128)	0
Dense	(1, 3)	(1, 3)	(128 + 1)×3 = 387
Total No. of Parameters			168195 + ((1 + d)×512)

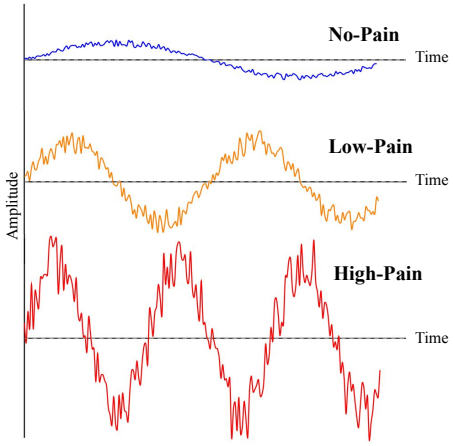


FIGURE 3.4: Some sample audio signals of AD_{VIVAE} (3-Class).

3.3 Experiments and Results

The PSA_{audio} System has two versions: i) hand-crafted and ii) deep learning, as described in Section 3.2.2 and 3.2.4 (see Fig. 3.3), respectively. In this experiment,

two datasets have been used to evaluate the performance of the proposed systems. First, we discuss the datasets and then report the performance.

(i) Variably Intense Vocalizations of Affect and Emotion Corpus: This is a speech audio database [231]. We rephrase this database as AD_{VIVAE}. Some sample signals from the dataset are shown in Fig. 3.4. The database comprises non-speech emotion vocalizations recorded from human participants. It contains 1085 audio recordings collected from 11 speakers, each expressing various emotions, including fear, anger, surprise, pain, achievements, and sexual pleasure. For each emotion, recordings were captured at four intensity levels: 'low', 'moderate', 'peak', and 'strong'. This study focuses on detecting pain intensities, categorizing the recordings into three main classes: Pain Intensity-0 to Pain Intensity-2. For simplicity, recordings labeled with 'low' and 'moderate' intensities are grouped under the Pain Intensity-0 class, 'peak' is assigned to the Pain Intensity-1 class, and 'strong' is classified as Pain Intensity-2. The emotions in the dataset are therefore distributed across these three pain intensity classes based on their intensity levels. The summary of this dataset is reported in Table 3.2. Experiments with the proposed system evaluate its performance using different training and testing sets derived from these recordings.

TABLE 3.2: Number of sample in each class of AD_{VIVAE} (3-Class).

Pain	**Class**	**Sample**
PI_0	AC_1	530
PI_1	AC_2	272
PI_2	AC_3	282

(ii) Ryerson Audio-Visual Database of Emotional Speech and Song: This is an emotional Speech and Song database [42]. The dataset is referred as $\text{AD}_{\text{RAVDESS}}$. Some samples of the dataset are given in Fig. 3.5. It encompasses 7356 files, totaling 24.8 GB. The dataset features 24 professional actors (12 female, 12 male), who vocalize two lexically matched statements in a neutral North American accent. The speech component captures expressions of calm, happiness, sadness, anger, fear, surprise, and disgust, while the song component includes emotions of calm, happiness, sadness, anger, and fear. Among various facial expressions, emotions such as sadness, anger, disgust, and fear have been found to share a meaningful correlation with the expression of pain. These emotions, when contrasted with a neutral or

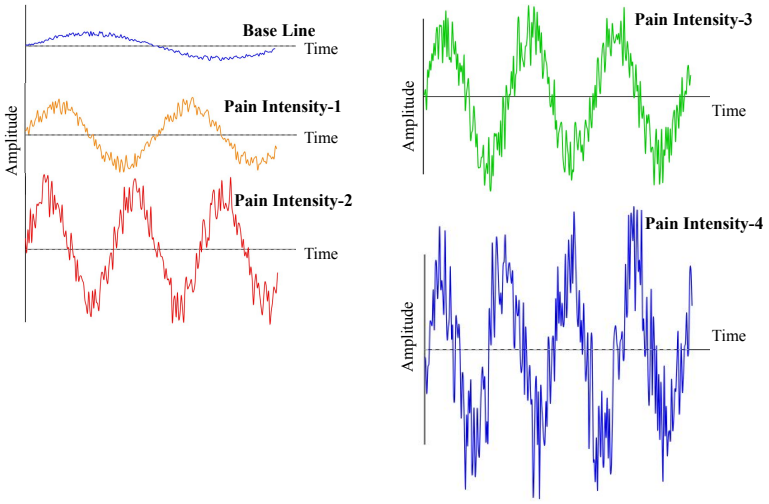


FIGURE 3.5: Some samples of $AD_{RAVDESS}$ (5-Class).

calm demeanor, are indicative of varying degrees of discomfort. In this study, calm expressions are considered under the no-pain category, which is expressed as Pain Intensity-0, while the other above-mentioned expressions are classified into four distinct pain level ranges from Pain Intensity-1 to Pain Intensity-4. The summary of the dataset is presented in Table 3.3.

TABLE 3.3: Number of sample in each class of $AD_{RAVDESS}$ (5-Class).

Pain	**Class**	**Sample**
PI_0	AC_1	96
PI_1	AC_2	384
PI_2	AC_3	384
PI_3	AC_4	384
PI_4	AC_5	192

$HANDPSA_{audio}$ System used various audio features, including statistical features, MFCCs, and spectral features of each audio signal. This set of features includes 11 statistical features, 128 MFCC features, and 224 spectral features. Table 3.4 summarizes the performance of the proposed $HANDPSA_{audio}$ System when evaluated using these machine learning classifiers. All machine learning classifiers classify features into 3 or 5 pain classes corresponding to AD_{VIVAE} and $AD_{RAVDESS}$, respectively. The results of the proposed $HANDPSA_{audio}$ System on the dataset AD_{VIVAE} and $AD_{RAVDESS}$ are reported in Table 3.4 and 3.5, respectively.

TABLE 3.4: The performance the proposed $\text{HANDPSA}_{\text{audio}}$ System on AD_{VIVAE} using different classifiers.

	50-50% training-testing set				**75-25% training-testing set**			
				$f_{\textbf{stat}}$				
Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Accuracy**	**F1-Score**	**Precision**	**Recall**
LR	49.56	33.52	24.76	49.47	50.21	33.39	25.12	50.21
kNN	75.02	49.22	58.17	75.09	52.41	49.22	60.53	49.05
DT	**75.11**	**47.47**	**58.17**	**75.13**	**50.31**	**33.43**	**25.12**	**50.41**
RF	72.86	68.56	69.37	70.49	48.74	46.26	39.28	43.57
SVM	49.63	33.52	24.76	49.32	50.24	33.36	25.14	50.19
				$f_{\textbf{mfcc}}$				
LR	52.23	40.52	35.63	52.14	75.34	64.74	58.18	75.07
kNN	52.23	40.53	35.63	52.23	75.66	67.39	63.29	75.22
DT	**95.18**	**92.95**	**93.52**	**92.38**	**97.57**	**95.08**	**96.45**	**95.88**
RF	91.74	89.43	90.56	92.41	94.33	91.42	92.63	994.04
SVM	42.31	35.64	31.24	41.39	67.43	94.69	96.21	95.72
				$f_{\textbf{spec}}$				
LR	49.71	33.12	49.76	40.44	75.37	64.68	58.37	75.21
kNN	75.13	58.17	75.21	65.77	75.18	67.77	58.43	75.26
DT	**96.45**	**95.24**	**94.03**	**94.63**	**98.19**	**97.15**	**97.45**	**97.13**
RF	92.67	92.14	92.81	92.18	96.29	92.38	96.57	95.53
SVM	96.31	95.23	94.24	94.73	97.56	94.52	96.57	97.18

TABLE 3.5: The performance of $\text{HANDPSA}_{\text{audio}}$ System on $\text{AD}_{\text{RAVDESS}}$ using different classifiers.

	50-50% training-testing set				**75-25% training-testing set**			
				f_{stat}				
Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Accuracy**	**F1-Score**	**Precision**	**Recall**
LR	48.42	47.73	46.72	57.44	51.81	50.16	49.99	60.55
kNN	66.81	60.75	58.67	66.81	73.54	65.61	61.63	73.24
DT	**91.23**	**91.1**	**90.15**	**92.45**	**92.88**	**92.07**	**91.32**	**91.99**
RF	88.24	85.34	86.18	87.09	89.62	86.47	87.55	88.23
SVM	54.67	42.13	36.47	54.67	51.99	51.99	51.99	51.99
				f_{mfcc}				
LR	53.21	52.45	51.34	63.12	56.93	56.12	54.93	67.54
kNN	73.12	66.75	64.53	73.12	78.56	71.42	69.01	78.56
DT	**93.62**	**93.26**	**91.73**	**94.16**	**94.72**	**95.16**	**93.29**	**93.89**
RF	91.24	88.59	88.94	91.43	92.84	91.41	91.68	91.72
SVM	60.08	46.3	40.07	60.08	64.29	49.54	42.87	64.29
				f_{spec}				
LR	46.15	40.58	47.76	56.67	48.46	42.61	50.15	59.5
kNN	66.72	60.04	67.82	66.72	70.06	63.05	71.21	70.06
DT	**96.83**	**95.16**	**95.57**	**95.32**	**97.62**	**96.51**	**96.19**	**96.43**
RF	94.73	92.58	92.39	92.47	93.21	95.35	92.62	92.17
SVM	53.36	53.12	52.27	52.69	55.42	55.78	55.09	55.09

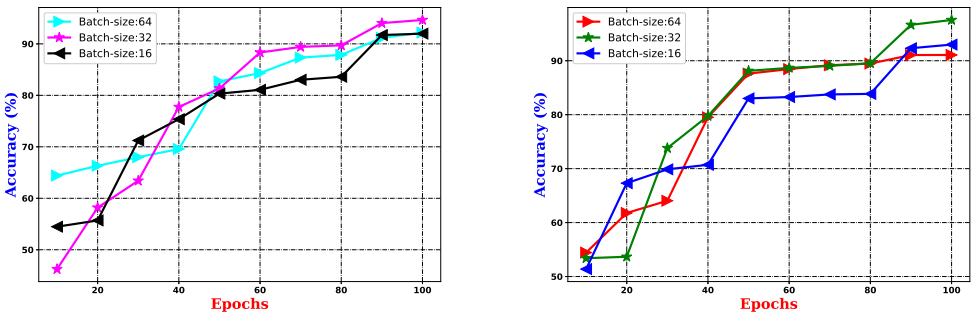
The performance displayed in Tables 3.4 and 3.5 indicates that, in most cases, the Decision Tree classifier achieved the highest accuracy in various feature types for both the 50:50 and 75:25 training-testing splits and for AD_{VIVAE} and $\text{AD}_{\text{RAVDESS}}$. Moreover, MFCC and spectral features were found to have a greater impact on

TABLE 3.6: Performance of the proposed $\text{DLPSA}_{\text{audio}}$ System on AD_{VIVAE} and $\text{AD}_{\text{RAVDESS}}$.

Dataset	50-50% training-testing set				75-25% training-testing set			
	Accuracy	**F1-Score**	**Precision**	**Recall**	**Accuracy**	**F1-Score**	**Precision**	**Recall**
AD_{VIVAE}	98.05	97.24	96.72	98.38	98.32	97.81	96.92	98.53
$\text{AD}_{\text{RAVDESS}}$	97.84	96.39	96.18	98.13	98.58	98.11	97.29	98.46

performance than statistical audio frequency features. Hence, in this chapter, as $\text{HANDPSA}_{\text{audio}}$ System DT has been considered along with f_{mfcc}.

In $\text{DLPSA}_{\text{audio}}$ system, 11 statistical features, 128 MFCC features, and 224 spectral features are combined to form a 363-dimensional feature vector for each speech audio sample. This feature vector is then used as input to the proposed deep learning architecture in the sentiment analysis system based on audio data. However, in the $\text{DLPAS}_{\text{audio}}$ system, the performance depends on the batch size and epochs, and this is justified in Fig. 3.6. This figure helps to fix batch size = 32 and epoch = 100, and these settings for batch size and number of epochs are selected for further experimentation in the $\text{DLPSA}_{\text{audio}}$ system. Table 3.6 displays the performance of $\text{DLPSA}_{\text{audio}}$ on AD_{VIVAE} and $\text{AD}_{\text{RAVDESS}}$.



(a) (b)

FIGURE 3.6: The performance of the proposed $\text{DLPSA}_{\text{audio}}$ System with (a) 50-50% training-testing, and (b) 75-25% training-testing sets for 3-class classification problem.

The results of these two experimental setups are further compared in Fig. 3.7 and 3.8, where the performance of the $\text{HANDPSA}_{\text{audio}}$ System and $\text{DLPSA}_{\text{audio}}$ System is presented under different training-testing protocols with AD_{VIVAE} and $\text{AD}_{\text{RAVDESS}}$, respectively. Fig. 3.7 and Fig. 3.8 help in visualizing the performance for the $\text{PSA}_{\text{audio}}$ systems.

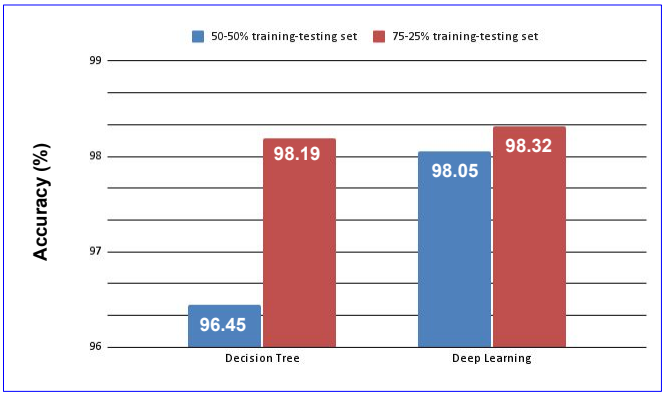


FIGURE 3.7: Comparison of accuracy of the proposed $\text{HANDPSA}_{\text{audio}}$ System and $\text{DLPSA}_{\text{audio}}$ System on AD_{VIVAE} dataset.

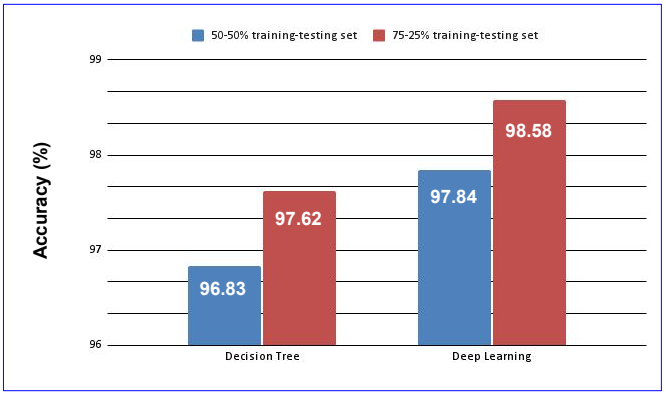


FIGURE 3.8: Comparison of accuracy of the proposed $\text{HANDPSA}_{\text{audio}}$ System and $\text{DLPSA}_{\text{audio}}$ System on $\text{AD}_{\text{RAVDESS}}$ dataset.

Fig. 3.9 represents how well two models, DT and FCN, performed in classifying three categories: AC_1, AC_2, and AC_3 on AD_{VIVAE}. The DT model has done a good job in correctly identifying 259 AC_1, 133 AC_2, and 138 AC_3 cases. It made only a few errors, such as misclassifying 4 AC_1 cases as AC_2. The FCN model performed even better. It correctly predicted 263 AC_1 cases and made just one AC_1-to-AC_2 mistake. Its accuracy for AC_2 and AC_3 was nearly the same as DT, with 132 and 138 correct predictions, respectively. Overall, both models worked well, but the FCN was slightly better. When it comes to the AC_3 class, both models perform well, with FCN achieving perfect recall (138 of 141), and DT matching closely. These results suggest that FCN is a bit stronger, particularly in reducing confusion between similar groups.

Fig. 3.10 presents the way the DT and FCN models performed across five classes AC_1

Predicted / Actual	AC_1	AC_2	AC_3
AC_1	97.74	1.51	0.75
AC_2	0.74	97.79	1.47
AC_3	0	2.13	97.87

Confusion matrix for DT

Predicted / Actual	AC_1	AC_2	AC_3
AC_1	99.25	0.38	0.38
AC_2	1.47	96.32	2.21
AC_3	1.42	0	98.58

Confusion matrix for FCN

FIGURE 3.9: Confusion matrices in percentage for $\text{HANDPSA}_{\text{audio}}$ System and $\text{DLPSA}_{\text{audio}}$ System on 3-class classification problem for AD_{VIVAE} dataset.

Predicted / Actual	AC_1	AC_2	AC_3	AC_4	AC_5
AC_1	95.83	2.08	2.08	0	0
AC_2	0.52	97.92	1.56	0	0
AC_3	0	1.04	97.40	0.52	1.04
AC_4	0	0.52	0.52	98.44	0.52
AC_5	0	0	2.08	1.04	96.88

Confusion matrix for DT

Predicted / Actual	AC_1	AC_2	AC_3	AC_4	AC_5
AC_1	97.92	2.08	0	0	0
AC_2	0	99.48	0	0.52	0
AC_3	0.52	1.04	96.35	1.04	1.04
AC_4	0	1.04	1.04	97.92	0
AC_5	0	0	0	2.08	97.92

Confusion matrix for FCN

FIGURE 3.10: Confusion matrices in percentage for $\text{HANDPSA}_{\text{audio}}$ System and $\text{DLPSA}_{\text{audio}}$ System on 5-class classification problem for $\text{AD}_{\text{RAVDESS}}$ dataset.

to AC_5 on $\text{AD}_{\text{RAVDESS}}$. The DT model does a solid job, especially with 46 correct predictions for the AC_1 class and very few mistakes, like just one case misclassified as AC_2. It also handles the AC_2 to AC_4 classes well, getting around 187 to 189 right, though there is some mix-up between AC_3 and the nearby classes AC_4 and AC_5. The FCN model, on the other hand, takes it a step further by making even more accurate predictions, like 191 correct for AC_2 and 94 for AC_5, while also making fewer mistakes across the board. For instance, it avoids misclassifying any AC_5 cases as AC_3, something the DT model didn't quite get right. Overall, both models do a great job, but the FCN stands out by drawing clearer lines between classes, especially for the less common ones like AC_1 and AC_5. This shows that the FCN is better at dealing with uneven class sizes and subtle similarities between classes, thanks to its stronger focus and fewer errors.

The audio-based features are extracted from the basic theory of signal processing systems from the continuous domain to the discrete domain. Hence, the employed

features are very well known in terms of audio-based feature extraction. Table 3.7 represents the comparison with some SoA methods. From this table, it is observed that both proposed systems ($\text{HANDPSA}_{\text{audio}}$ and $\text{DLPSA}_{\text{audio}}$) perform better than SoA methods. Further, it is to be noted that $\text{DLPSA}_{\text{audio}}$ System has better performance than $\text{HANDPSA}_{\text{audio}}$ System. Hence, $\text{DLPSA}_{\text{audio}}$ System is superior to all other competing methods. Therefore, in the future, when we say $\text{PSA}_{\text{audio}}$ System, it says about $\text{DLPSA}_{\text{audio}}$ System.

TABLE 3.7: Comparison of the performance of the proposed systems with the SoA methods.

Method	**Accuracy (%)**	**F1-Score**
3-Class Problem (AD_{VIVAE})		
Martinez et al. [232]	72.31	0.7131
Valstar et al. [233]	68.54	0.6744
Pantic & Patras [234]	65.16	0.6338
$\text{HANDPSA}_{\text{audio}}$ System	96.45	0.9541
$\text{DLPSA}_{\text{audio}}$ System	98.32	0.9629
5-Class Problem ($\text{AD}_{\text{RAVDESS}}$)		
Livingstone & Russo [235]	74.27	0.7346
Schuller et al. [236]	69.83	0.6828
Trigeorgis et al. [237]	81.49	0.8437
$\text{HANDPSA}_{\text{audio}}$ System	96.83	0.9516
$\text{DLPSA}_{\text{audio}}$ System	97.84	0.9569

3.4 Conclusions

This chapter presents the audio-based pain sentiment analysis system, denoted as $\text{PSA}_{\text{audio}}$. The system is developed in three main stages. First, audio signals undergo preprocessing, where noise is reduced using a Kalman filter to enhance signal quality. Subsequently, three hand-crafted feature sets are extracted: (i) statistical features, (ii) Mel-Frequency Cepstral Coefficients (MFCCs), and (iii) spectral features, each capturing different characteristics of the frequency domain. These features are evaluated through two distinct experimental approaches. In the first, known as $\text{HANDPSA}_{\text{audio}}$, conventional machine learning classifiers are applied independently to each feature set to assess their individual discriminative potential.

In the second approach, termed $\text{DLPSA}_{\text{audio}}$, the extracted features are concatenated and fed into a Fully Connected Network (FCN) to investigate whether the integration of the diverse acoustic descriptors improves the classification performance. Both systems are evaluated using two benchmark datasets: AD_{VIVAE} and $\text{AD}_{\text{RAVDESS}}$. The results of extensive experiments demonstrate that the $\text{DLPSA}_{\text{audio}}$ system consistently outperforms not only $\text{HANDPSA}_{\text{audio}}$ but also several state-of-the-art methods. Consequently, the $\text{DLPSA}_{\text{audio}}$ framework is adopted as the final prediction model for the $\text{PSA}_{\text{audio}}$ system in this chapter. The incorporation of audio data into pain classification represents a significant advancement in telemedicine by enabling real-time, non-invasive assessment of the conditions of patients. Variations in vocal parameters such as pitch modulation, speech rhythm, and voice stability serve as reliable acoustic biomarkers to estimate pain intensity. By utilizing these auditory cues, the $\text{PSA}_{\text{audio}}$ system facilitates automatic classification of pain level and eliminates the limitations of subjective self-reporting methods such as pain rating scales. This approach not only improves diagnostic accuracy but also supports timely medical intervention, thereby enhancing the effectiveness of remote healthcare delivery.

Despite the notable advantages offered by the $\text{PSA}_{\text{audio}}$ System, it also faces important challenges, especially related to data ambiguity caused by differences in tone, language, cultural background, and the authenticity of spoken expressions. These issues can affect the reliability of audio-based pain recognition and highlight the need for careful and reliable verification processes to ensure trustworthy input signals. To overcome these limitations, the next work of this thesis includes the use of computer vision methods by incorporating image-based data as visual signals. Facial expressions displayed during painful vocalizations serve as additional supportive visual evidence. This helps to improve both the accuracy and reliability of automated pain detection systems. The details of this approach are presented in the next chapter.

Chapter 4

Image-Based Pain Sentiment Analysis

In this chapter, we have developed an image-based pain sentiment analysis (PSA_{image}) system to address several challenges identified in the case of an audio-based pain analysis system. One of the main challenges is to ensure that the tone of the speaker accurately reflects the true level of pain. This fundamental reliability issue prompted us to incorporate facial expression analysis as a complementary modality because, for the human being, facial movements are governed by consistent neuromuscular patterns and are difficult to consciously suppress, offering a more stable reflection of true pain labels. The PSA_{image} system is a static visual signal analysis to develop a robust pain assessment framework that will overcome the limitation of the PSA_{audio} system.

Pain assessment through facial expression analysis has emerged as a critical tool in healthcare, particularly for patients who cannot write or speak the personnel discomfort, such as neonates, the elderly, and individuals with cognitive impairments. PSA_{image} system utilizes several methods from image processing, pattern recognition, computer vision, machine learning, and deep learning areas to detect and quantify pain intensity from facial images. For the detection of pain intensity, both (i) hand-crafted features and classification, and (ii) deep learning techniques have been designed. The hand-crafted features, such as facial Action Units (AUs) like brow lowering or lip tightening, are clinically validated and provide interpretable

insights into muscle movements associated with pain. On the other hand, deep learning methods like CNNs and Vision Transformers excel at identifying subtle, complex patterns such as micro-expressions and temporal changes that might be missed by manual techniques. These methods adapt well to individual differences in pain expression, including cultural variations or personal tendencies. By combining both approaches, the system becomes more robust, such as hand-crafted features reduce overfitting by incorporating domain knowledge [238], while deep learning enhances generalization across different populations, lighting conditions, and facial poses [16]. This hybrid strategy effectively balances interpretability with predictive power, an essential combination for reliable and accurate pain assessment, especially in real-world or clinical environments.

Recent approaches employ deep learning models, mainly CNNs, to analyze subtle facial cues such as brow lowering, lip tightening, and eye closure, which are strongly correlated with pain. Benchmark datasets like UNBC Shoulder Pain Archive [239] and the BioVid Heat Pain [240] databases have enabled significant progress in this field by providing standardized, annotated facial expressions for model training and validation. Despite advancements, challenges remain, including variability in pain expression across cultures, occlusions due to medical equipment, and the need for real-time, deployable solutions in clinical environments.

Two types of $\text{PSA}_{\text{image}}$ systems have been proposed in this chapter. One system is a hand-crafted feature-based system ($\text{HANDPSA}_{\text{image}}$ System) and another one is a deep learning based system ($\text{DLPSA}_{\text{image}}$ System). For both systems, a common image preprocessing step is applied. A schematic diagram for the proposed systems is given in Fig. 4.1.

This chapter reviews the literature in Section 4.1. In Section 4.2, the implementation of the proposed $\text{PSA}_{\text{image}}$ systems has been elaborated. The experiments and results are discussed in Section 4.3. This chapter is concluded in Section 4.4.

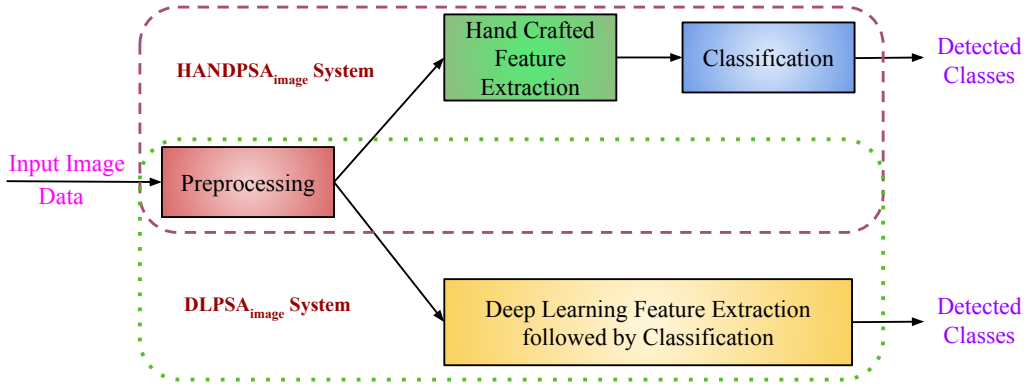


FIGURE 4.1: Block diagram of the proposed PSA$_{\text{image}}$ Systems.

4.1 Literature Review

Many existing automated techniques for pain detection struggle to accurately recognize pain based solely on facial expressions. Within the field of facial expression-based pain recognition, several well-established feature extraction methods are widely used. These include Active Appearance Models (AAM) and Active Shape Models (ASM) [241], which capture both shape and appearance features. Other commonly employed techniques include Local Binary Patterns (LBP) [242], known for their effectiveness in texture analysis, and Gabor wavelets [243], which are useful for capturing spatial frequency information in facial images. It has been noted that the approaches now in use are related to facial representation-based pain detection. Specific techniques rely on deep learning, whereas others are focused on non-deep learning [244]. Nowadays, substantial progress has been achieved in this particular field of study. Regarding categorization, it has been noted that the retrieved face features [245] influence classification. A few popular feature extraction approaches perform this feature extraction task. The classification tasks are undertaken using well-established supervised machine learning methodologies like kNN [246], SVM [247], and Logistic Regression [248].

For efficient identification of face and other objects from images as some deep learning-based CNN architectures have also been used in [249] and [250]. Rodriguez et al. [81] proposed a hybrid CNN-Transformer architecture for enhanced pain recognition, while Gupta et al. [251] introduced a lightweight model optimized for edge devices. In recent years, it has been observed that the performance of CNN models

is very high using GoogleNet [252] and AlexNet [134] CNN architectures. A new method of emotional analysis based on a CNN-BiLSTM hybrid neural network has been proposed by Liu et al. [253]. A comparative study on bio-inspired algorithms for sentiment analysis had been done by Yadav and Vishwakarma [6]. In computer vision, extracted features by pre-trained CNNs are used for various objectives such as object identification, emotion recognition, etc. It is also observed that the performance of CNN-pulled features is much better than hand-crafted features. In recent years, the growth in the research field associated with deep learning has provided solutions to several cutting-edge problems. Deep learning-based algorithms are capable of revealing inherently obscured patterns in complicated datasets, thus for feature extraction and classification purposes [134].

Recent developments in image-based pain sentiment analysis are making systems smarter and more reliable by combining facial expressions with signals like EEG and EMG to understand pain better [254], while self-supervised learning helps to reduce the need for large labeled datasets by using unlabeled videos [255]. Popular datasets like the UNBC-McMaster Shoulder Pain Archive [256] continue to lead the field, though newer ones such as BioVid [257] and X-ITE [258] include advanced features like thermal imaging. Cutting-edge architectures like transformers [259] and graph neural networks [260] are better at spotting pain over time and modeling how facial features interact. Techniques like contrastive learning [261] and vision-language models like CLIP [262] help improve accuracy and even allow systems to classify pain without prior examples. Weak supervision [263] and active learning [264] are used to focus on important frames and reduce labeling effort, while the EmoPain Challenge 2023 brought focus to real-world continuous pain prediction [265]. Meta-learning [266] allows systems to adapt to new people with little data, and explainable AI tools [267] build clinical trust by showing which parts of the face signal pain. Privacy-friendly training methods like federated learning [268], synthetic pain generation using diffusion models [269], and automatic model design [270] are making systems more practical. Researchers are also addressing biases across datasets [271] with new benchmarks like Aff-Wild2 [272], and multitask learning [273] is helping models do several related jobs at once. Attention mechanisms [274], few-shot learning [275], and model compression [276] further boost performance and make deployment easier. Other efforts include using datasets like DISFA+ with pain-specific

labels [277], semi-supervised training [278], hybrid models [279], adversarial learning [280], mobile optimization [281], and causal discovery to filter out misleading features [282]. Projects like the OMG-Pain Challenge [283] encourage tracking pain over time, and multimodal transformers [284] now merge data from facial, voice, and movement inputs. Self-attention in vision transformers [285], unsupervised methods like contrastive predictive coding [286], and dynamic graph models [287] are also enhancing how pain is recognized. Real-time adaptation [288], better generalization across people [289], modeling pain changes with neural ODEs [290], and few-sample recognition with prototypical networks [291] are pushing boundaries. Synthetic augmentation that keeps identities intact [292], hybrid transformer-CNN models [293], and causal interventions [294] are helping reduce bias, while memory-augmented networks [295] bring in external knowledge for smarter pain detection.

4.2 Proposed $\text{PSA}_{\text{image}}$ Systems

The Fig. 4.1 illustrates the flow diagram of the proposed $\text{PSA}_{\text{image}}$ systems. The figure shows that a common preprocessing technique is applied to both systems to extract the facial region from the corresponding input image. Then, in the $\text{HANDPSA}_{\text{image}}$ system, the preprocessed facial image undergoes feature extraction followed by classification. On the other hand, the preprocessed image in the $\text{DLPSA}_{\text{image}}$ System is processed by a deep model to identify the pain level. In the following, first we describe the preprocessing step, since it is common to both the systems, and subsequently we present $\text{HANDPSA}_{\text{image}}$ System and $\text{DLPSA}_{\text{image}}$ System.

4.2.1 Image Preprocessing

In an unconstrained imaging environment, noise, illuminations, variations in poses, and background are main challenges in any image-based application [244]. So, preprocessing of the input image is very important to reduce the effect of these challenging issues. In the proposed $\text{PSA}_{\text{image}}$ system, the first step of the preprocessing is to detect the facial region from an input image. The extracted face region is then

normalized such that the fixed-dimensional feature vector can be extracted from each preprocessed image. In this work for face detection, a tree-structured part model has been employed, which works for all variants of facial poses. The Tree Structured Part Model (TSPM) [296] is a hierarchical framework for facial feature localization, which combines a global deformable model with local part templates to robustly detect faces and their keypoints under varying poses and expressions. The model represents a face as a mixture of trees, where each tree corresponds to a different viewpoint (e.g., frontal, profile), and each node in the tree represents a facial part (e.g., eyes, nose, mouth) associated with a spatial constraint. The working flow diagram of the TSPM method has been shown in Fig. 4.2.

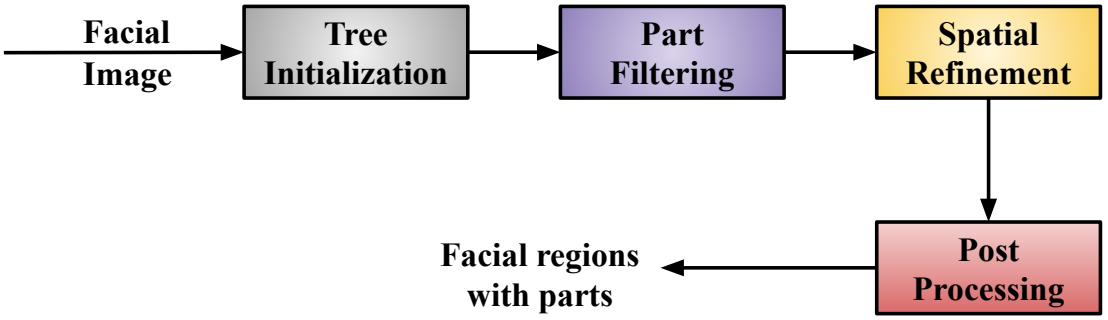


FIGURE 4.2: Block diagram of TSPM.

From Fig. 4.2, it has been observed that the input image contains potential faces, and from this image we have to find an approximate (initial) facial region, and then this facial region is refined by the TSPM method.

To find the initial facial region, we do the following steps:

(i) First, we convert the input image into a grayscale image.

(ii) Next, apply the histogram equalization for lighting normalization, followed by the Gaussian filtering to remove noise.

(iii) Then, use a method for coarse to fine details with finding the scores for landmark detection using $\text{Score}(x, y) = \sum_i w_i \cdot \phi_i(x, y)$, where the ϕ_i is HOG features computed over the position of the landmark. The w_i is the weight adjusted by applying the SVM classifier for landmark and non-landmark positions. This results initial facial bounding box.

The approximate face region is passed to TSPM for refinement of the initial detection through hierarchical part localization. The steps of TSMP are as follows:

1. **Tree-Structured Model Initialization**
   - Define facial parts in a hierarchical tree structure:
     - Root node: Face bounding box
     - Level 1: Eyes, nose, mouth regions
     - Level 2: Eye corners, nose tip, mouth corners
   - Establish geometric constraints between parts
2. **Part Detection**
   - Apply deformable part models at multiple scales:

$$\text{PartScore}(p_i) = \text{Appearance}(p_i) + \text{DeformationCost}(p_i, p_{parent}) \tag{4.1}$$

   - Use sliding window approach with learned part filters
3. **Spatial Refinement**
   - Optimize part locations using spring-like constraints:

$$E(\mathbf{p}) = \sum_i \text{Appearance}(p_i) + \sum_{(i,j)\in E} \text{Dist}(p_i, p_j) \tag{4.2}$$

   - Solve using dynamic programming for efficient inference
4. **Post-Processing**
   - **Non-Maximum Suppression (NMS)**:
     - Remove duplicate detections using IoU thresholding
     - Keep the highest-scoring detection in each neighborhood
   - **Geometric Verification**:
     - Enforce facial symmetry constraints
     - Validate anthropometric ratios between parts

5. **Output**

   - Final detected facial regions with:
     - Precise bounding boxes
     - Labeled facial parts (eyes, nose, mouth)
     - Confidence scores for each component

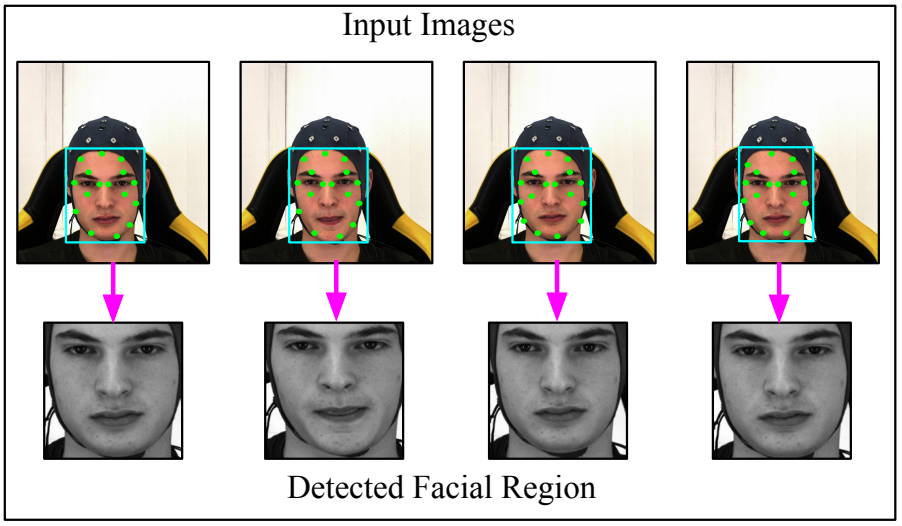
For the frontal face, this method computes sixty-eight landmark points, while for the profile face, thirty-nine landmark points have been extracted. For normalization purposes, the bilinear image interpolation method has been employed on each extracted face region. The output of the image preprocessing is shown in Fig. 4.3.

So, from the above image preprocessing, from each input image, a facial region of size $N \times N \times 3$ dimension image F is obtained, which goes to feature computation.

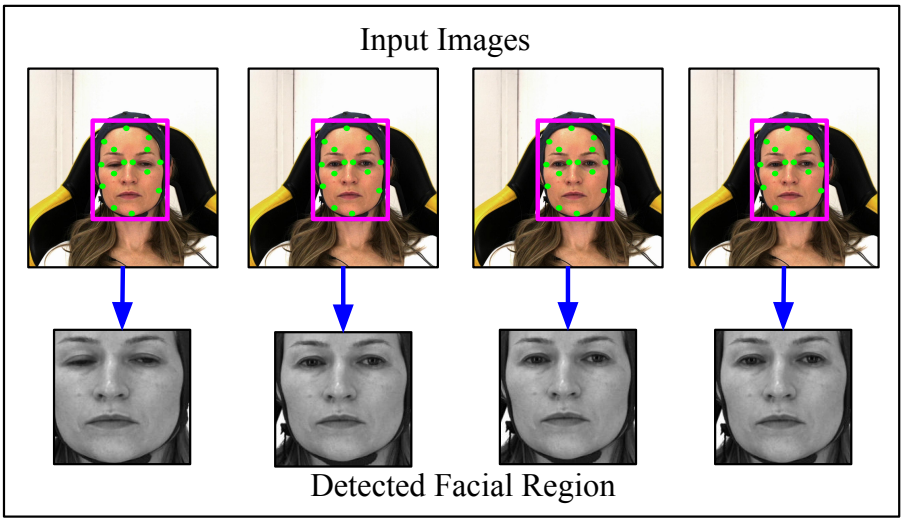
4.2.2 HANDPSA$_{\text{image}}$ System

The above face extracted region (see Fig. 4.3) contains tones in regular or irregular patterns. These textural characteristics allow us to extract features that are more distinct and meaningful. This helps in better separating different classes and reduces confusion between similar ones, both within the same category (intra-class) and across different categories (inter-class). It also helps to handle the issues caused by poor image quality, especially when images are affected by noise and other artifacts. The feature extraction module works as an interpreter to extract admissible non-rigid information, which is very important for manifesting Action Units (AUs) from the perceptions of computer vision [297]. There can be two categories of features, such as appearance-based and geometric-based [298]. Various geometric measurements related to coordinates and fiducial points are extracted in the geometric-based feature extraction technique. Similarly, the appearance-based approach extracts various features from pixel intensity values.

In feature representation, the textures in the facial region are mainly analyzed using three approaches : (i) structural, (ii) statistical, and (iii) transform-based approaches [299]. Non-rigid facial muscular action exposes emotions. The statistical



(a)



(b)

FIGURE 4.3: Results of image preprocessing of $\text{PSA}_{\text{image}}$ System.

analysis of the texture pattern is more convenient and practical than the structural and transform-based approaches. In the $\text{HANDPSA}_{\text{image}}$ System, we have used two feature extraction techniques, Local Binary Pattern (LBP) and histogram of oriented gradients (HOG). These feature extraction methods have been discussed as follows:

Local Binary Pattern (LBP): It is a powerful texture descriptor that encodes local image patterns by comparing each pixel with its neighbours. The working principle of LBP is that for any colour image, at first converted into a gray-scale image F. Then, consider a mask $w_{3\times3}$ over F, which slides over image. Then, for

each central position of w on F, 8-neighbouring pixels are considered. Next, with respect to the central pixel, each neighbouring pixel is assigned a value of 1 if its intensity is greater than or equal to the centre pixel and 0 otherwise. These binary values are then concatenated to form the LBP code for the central position of w. Now perform this for all possible positions of w over F such that a LBP-coded image F_{lbp} is obtained. The generated F_{lbp} has pixel intensity from 0 to 255. So, the histogram of F_{lbp} is 256-dimensional f_{lbp} feature vector. An example of LBP-based feature extraction has been shown in Fig. 4.4.

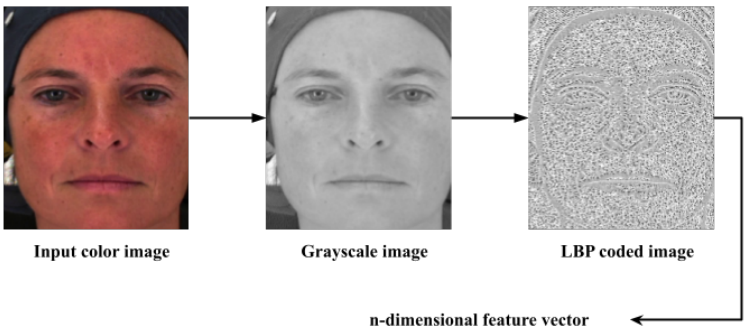


FIGURE 4.4: An example of feature extraction using the LBP technique.

Histogram of Oriented Gradient (HOG): This technique is used to extract edge and shape information from images by computing the distribution of gradient orientations within localized cells. Here, a colour image is first converted to grayscale, denoted as F. Then, considering the two Sobel operators with 3×3 filter masks, gradient images g_x and g_y are obtained by applying these filters over F. Then based on the computation of g_x, and g_y, the magnitude image $\sqrt{g_x^2 + g_y^2}$, and phase image $\tan^{-1}\left(\frac{g_y}{g_x}\right)$, are obtained. Then, based on orientation binning, an orientation histogram of its local cell (e.g., 8×8 pixels) is created, and the effects of illumination and contrast are reduced through the normalization of groups of neighbouring cells. Then, this creates the HOG image (F_{HOG}) of F. In this, from each F_{HOG}, 81-dimensional f_{hog} feature vector is generated. An example of HOG-based feature extraction has been shown in Fig. 4.5.

This statistical feature extraction approach evaluates more information from the pixel intensity values of the image. It helps to compile and present the appearance-based features from an image. Since the preprocessed face image F may have both

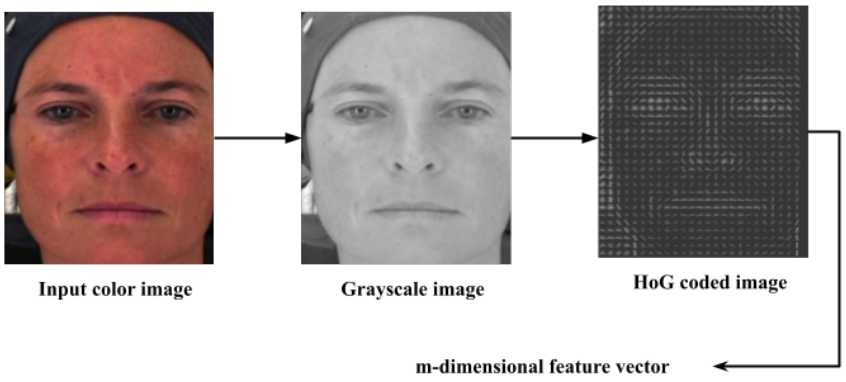


FIGURE 4.5: An example of feature extraction using the HOG technique.

regular and non-regular patterns [244]. So, the statistical-based approaches are more suitable to analyze regular as well as non-regular patterns.

Here, during feature computation, both global and local feature representation schemes have been considered, and for that, different image partitioning setups have been applied.

i) P_1: In this setup, the entire image is considered as a whole, and a global feature is extracted.

ii) P_2: The image is partitioned horizontally into two sub-images, which helps to extract features from the sub-images.

iii) P_3: In this case, the image is partitioned vertically into two sub-images, and features are computed from each sub-image.

iv) P_4: The image is halved both horizontally and vertically, and features are extracted from each of the four sub-images.

The first scheme (P_1) gives a global feature of the facial image; whereas, schemes P_2 - P_4 give local features. These schemes are shown in Fig. 4.6. During statistical-based feature computation, the facial image F of $N \times N \times 3$ is converted to a gray-scaled image $N \times N$. Then, Local Binary Pattern (LBP), Histogram of Oriented Gradients (HOG) features are computed accordingly from the whole F to obtain its global feature using P_1 representation as f_G. To extract more local features, each F is divided into either two-equal halves (horizontally or vertically) (f_{HL}/f_{VL}) using P_2

and P_3 respectively, or four-equal quarters (f_{LG}) with the help of P_4. Here, f_G indicates feature computed globally, f_{HL} indicates feature computed horizontally local, f_{VL} indicates feature computed vertically local, and f_{LG} indicates feature computed locally to globally [300]. The f_{HL} is a feature vector obtained by concatenation of features extracted from two horizontal halves of F. Similarly, f_{VL} is a feature vector obtained by concatenation of features extracted from two vertical halves of F. The f_{LG} is a feature vector obtained by concatenation of features extracted from four matrices (two horizontally and then two vertically) of F. Therefore, $\text{HANDPSA}_{\text{image}}$ system may use four types of features such as f_G, f_{HL}, f_{VL}, and f_{LG}.

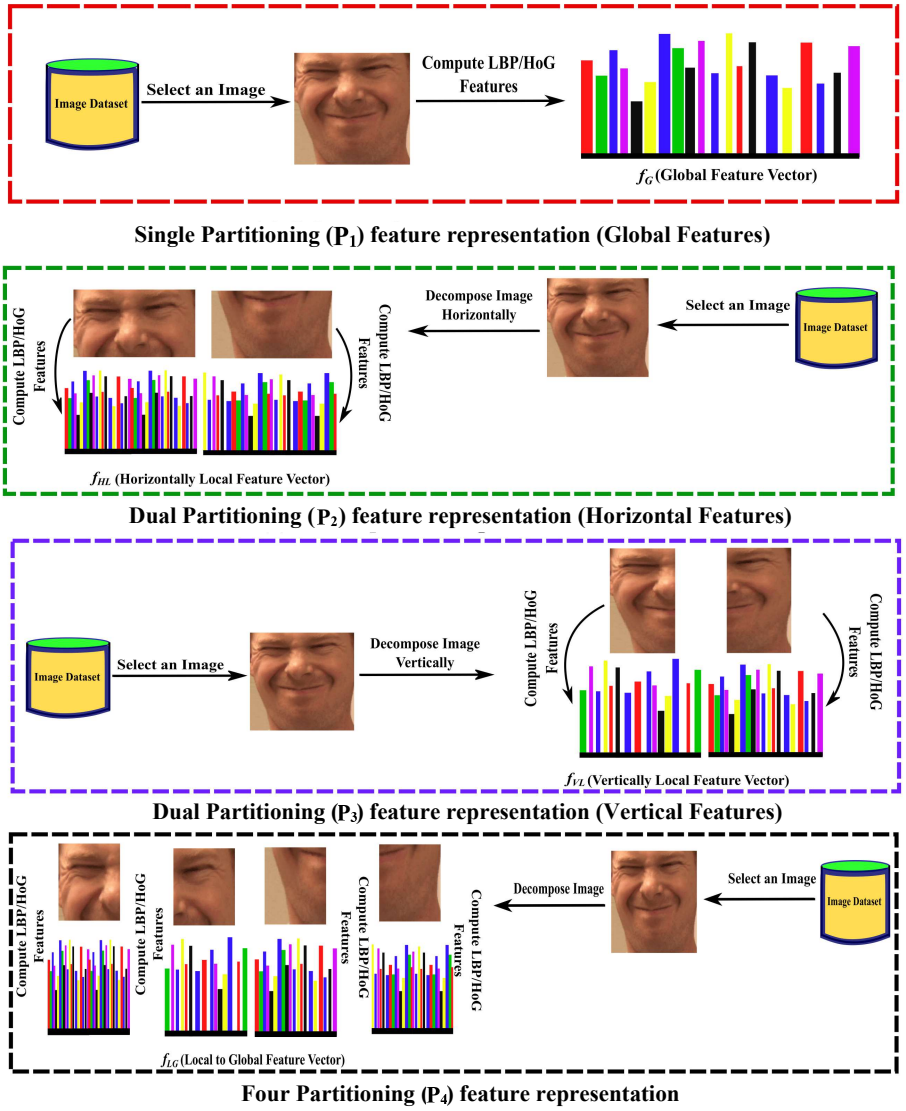


FIGURE 4.6: Statistical-based approach for feature representation from the facial region F.

4.2.3 Classification

Now the features extracted from the above schemes (shown in Fig. 4.6) undergo for classification task. For classification purposes, the DT, SVM, kNN, DT and LR classifiers have been employed to detect pain for a 3-class or 5-class problem.

4.2.4 $\mathbf{DLPSA_{image}}$ System

Although $\text{HANDPSA}_{\text{image}}$ system features might be helpful for interpretability or low-resource settings, deep learning always achieves better performance for most contemporary image classification tasks by learning data-driven optimal representations instead of depending on pre-specified feature extraction rules. Deep learning-based feature extractors, especially CNNs [252], have several key benefits compared to conventional $\text{HANDPSA}_{\text{image}}$ system approaches such as LBP and HOG for image classification. In contrast to the $\text{HANDPSA}_{\text{image}}$ system feature representation scheme is manually chosen, and domain-specific expertise is required to construct fixed feature descriptors, whereas deep learning models learn automatically hierarchical, discriminative features from raw data within their multi-layer architectures. This allows them to pick up intricate patterns, spatial hierarchies, and abstract representations usually lost in $\text{HANDPSA}_{\text{image}}$ system methods. Deep learning features work better even when the lighting changes, different poses, etc. Also, $\text{DLPSA}_{\text{image}}$ System combines feature extraction and classification into a single end-to-end framework that can be optimized together for the task at hand, as opposed to the $\text{HANDPSA}_{\text{image}}$ System features for which different classifiers need to be tuned. In the $\text{DLPSA}_{\text{image}}$ system, the transfer learning and fine-tuning techniques have also been applied such as to increase the performance over several challenging issues, which may be difficult in the case of the $\text{HANDPSA}_{\text{image}}$ system. In this work, the $\text{DLPSA}_{\text{image}}$ system using CNN [252] is employed for feature learning and the classification of images into predefined categories. These CNNs are widely recognized for their efficiency and effectiveness in tackling complex image analysis problems. The design and implementation of these CNNs have been demonstrated in the subsections below.

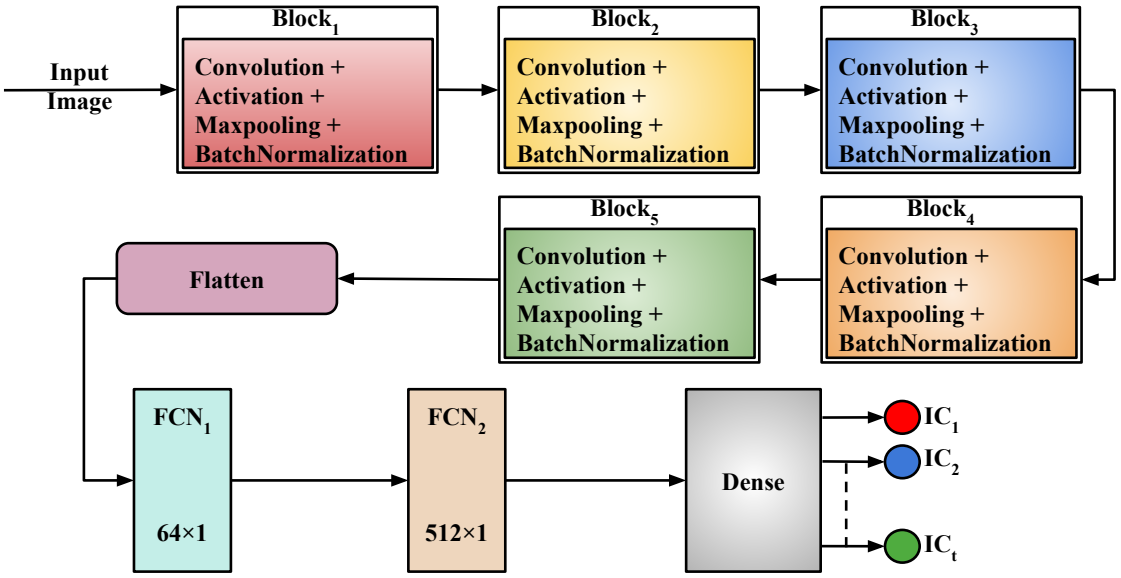


FIGURE 4.7: Proposed CNN_1 architecture.

4.2.5 Proposed CNN_1 System

The CNN_1 architecture has been structured into five blocks and three fully connected layers. It starts with $Block_1$, where a 3×3 convolutional layer with 64 filters is used on a facial input image of size $48 \times 48 \times 3$, followed by ReLU activation, 2×2 max-pooling that reduces the spatial dimensions by half, and batch normalization. $Block_2$ follows the same sequence with another 3×3 convolution (again using 64 filters) with pooling and normalization. In $Block_3$, the number of filters increases to 96, but the same convolution-pooling-normalization architecture is repeated, which is repeated in $Block_4$ with the same configurations. $Block_5$ scales back filter count to 64 but continues to employ the same layer operations. Every block, therefore, essentially halves the size of the image and extracts more and more sophisticated features. Following the last convolutional block, the output is flattened into a 64-dimensional vector and fed through three fully connected layers of 512 neurons each, with dropout added for regularization. Finally, a dense with softmax classification layer of three-class pain sentiments is used. The model consists of a total of 475875 parameters, optimized for balanced complexity and performance in a classification problem with 3 categories. The architectural design of the proposed CNN architecture has been

shown in Fig. 4.7. The detailed description of the proposed CNN_1 architecture, along with adopted layers, output shape of feature maps at each layer, and the parameters involved at each layer, has been demonstrated in Table 4.1.

TABLE 4.1: Description of the CNN_1 architecture.

Layer	Output Shape	Image Size	Parameters
$Block_1$			
Convolution2D (3×3@64) (Activation: Relu)	$(n, n, 64)$	(48,48,64)	((3×3×3)+1)×64=1792
Maxpooling2D (2×2)	$(n_1, n_1, 64)$	(24,24,64)	0
Batch Normalization	$(n_1, n_1, 32)$	(24,24,64)	4×64=256
$Block_2$			
Convolution2D (3×3@64) (Activation: Relu)	$(n_1, n_1, 64)$	(24,24,64)	((3×3×64)+1)×64= 36928
Maxpooling2D (2×2)	$(n_2, n_2, 64)$	(12,12,64)	0
Batch Normalization	$(n_2, n_2, 64)$	(12,12,64)	4×64=256
$Block_3$			
Convolution2D (3×3@96) (Activation: Relu)	$(n_2, n_2, 96)$	(12,12,96)	((3×3×64)+1)×96=55392
Maxpooling2D (2×2)	$(n_3, n_3, 96)$	(6,6,96)	0
Batch Normalization	$(n_3, n_3, 96)$	(6,6,96)	4×96=384
$Block_4$			
Convolution2D (3×3@96) (Activation: Relu)	$(n_3, n_3, 96)$	(6,6,96)	((3×3×96)+1)×96=83040
Maxpooling2D (2×2)	$(n_4, n_4, 96)$	(3,3,96)	0
Batch Normalization	$(n_4, n_4, 96)$	(6,6,96)	4×96=384
$Block_5$			
Convolution2D (3×3@64) (Activation: Relu)	$(n_4, n_4, 64)$	(3,3,64)	((3×3×96)+1)×64=55360
Maxpooling2D (2×2)	$(n_5, n_5, 64)$	(1,1,64)	0
Batch Normalization	$(n_5, n_5, 64)$	(1,1,64)	4×64=256
Fully Connected			
Flatten	$1 \times 1 \times 64$=64		0
Dense+Dropout	512		$(64+1) \times 512$= 33280
Dense+Dropout	512		$(512+1) \times 512$=262656
Dense	3		$(512+1) \times 3 = 1539$
Total Parameters			**475875**

In the CNN_1 architecture, the fully connected layer and output layer are arranged in such a manner to progressively extract, refine, and interpret features for effective classification. This hierarchical structure enables the network to learn increasingly complex features as the data flows deeper. After feature extraction, the fully connected layers integrate these features across the entire image, with dropout applied to prevent overfitting. Finally, the output layer with softmax classification produces a probability distribution for classification, ensuring the model can accurately distinguish between the target classes. This arrangement ensures efficient learning, robust generalization, and effective decision-making.

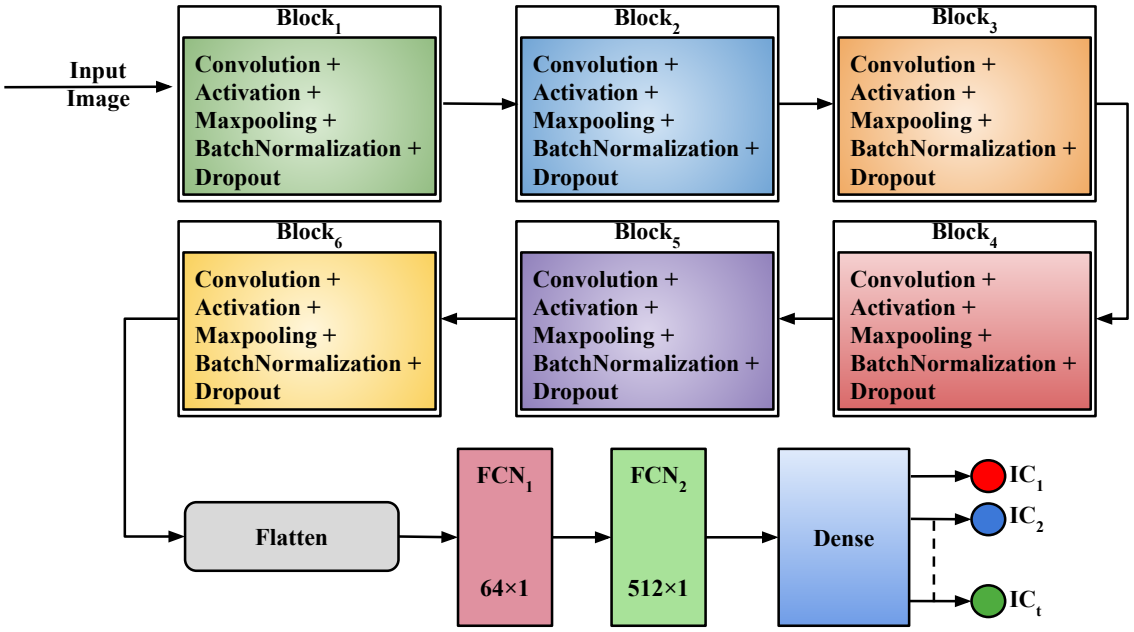


FIGURE 4.8: Proposed CNN_2 architecture.

4.2.6 Proposed CNN_2 System

The CNN_2 architecture is composed of six convolutional blocks and three fully connected layers for classification. The $Block_1$ starts with 3×3 convolution with 32 filters, followed by batch normalization, ReLU activation, 2×2 max pooling to downsample the spatial dimensions, and dropout for regularization. $Block_2$ and $Block_3$ follow the same pattern but increase the number of convolutional layers to deepen feature extraction with 32 and 64 filters, respectively, keeping activation, pooling, batch normalization, and dropout at each step to keep training stable and avoid overfitting. $Block_4$ and $Block_5$ increase the filter numbers to 96 for a more detailed representation of features and keep the same layer order. $Block_6$ uses a final 3×3 convolution with 64 filters, normalization, activation, pooling, and dropout. The output is flattened and fed to the dense layers, where three fully connected layers are applied in sequence, each with 256 neurons, followed by batch normalization, ReLU activation, and dropout for deeper abstraction and regularization. The last output layer is a dense layer with t units for classification of t classes. The total number of parameters is around 3.1 million, with more than 3 million trainable parameters, ensuring that this architecture is resilient for challenging classification tasks with strong feature discriminability. The design of the proposed CNN_2 architecture has

TABLE 4.2: Description of the CNN_2 architecture.

Layers	**OutputShape**	**ImageSize**	**Parameters**	**Layers**	**OutputShape**	**ImageSize**	**Parameters**
$Block_1$							
Conv2D(3×3)@32	(n,n,32)	(96, 96, 32)	896				
ActivationReLU	(n,n,32)	(96, 96, 32)	0				
Maxpool2D(2×2)	(n_1, n_1, 32), $n_1 = n/2$	(48,48,32)	0				
BatchNorm	(n_1, n_1, 32)	(96, 96, 32)	128				
Dropout	(n_1, n_1,32)	(48,48,32)	0				
$Block_2$				**$Block_4$**			
Conv2D (3×3)@32	(n_1, n_1,32)	(48,48,32)	((3×3×3) + 1)×32 =9248	Conv2D (3×3)@96	(n_3, n_3,96)	(12,12,96)	((3×3×64)+1) ×96=55392
Activation ReLU	(n_1, n_1,32)	(48,48,32)	0	Activation ReLU	(n_3, n_3,96)	(12,12,96)	0
Maxpool2D (2×2)	(n_2, n_2,32) $n_2 = n_1/2$	(24, 24, 32)	0	Maxpool2D (2×2)	(n_4, n_4,96) $n_4 = n_3/2$	(6,6,96)	0
Batch Norm	(n_2, n_2,32)	(24, 24, 32)	4×32 = 128	Batch Norm	(n_4, n_4,96)	(6,6,96)	4×96=384
Dropout	(n_2, n_2,32)	(24, 24, 32)	0	Dropout	(n_4, n_4,96)	(6,6,96)	0
$Block_3$				**$Block_5$**			
Conv2D (3×3)@64	(n_2, n_2,64)	(24, 24, 64)	((3×3×32)+1) × 64=18496	Conv2D (3×3)@96	(n_4, n_4,96)	(6,6,96)	((3×3×96)+1) ×96=83040
Activation ReLU	(n_2, n_2,32)	(24, 24, 64)	0	Activation ReLU	(n_4, n_4,96)	(6,6,96)	0
Maxpool2D (2×2)	(n_3, n_3,64) $n_3 = n_2/2$	(12, 12, 64)	0	Maxpool2D (2×2)	(n_5, n_5,96) $n_5 = n_4/2$	(3,3,96)	0
Batch Norm	(n_3, n_3,64)	(12, 12, 64)	4×64=256	Batch Norm	(n_5, n_5,96)	(3,3,96)	4×96=384
Dropout	(n_3, n_3,64)	(12, 12, 64)	0	Dropout	(n_5, n_5,64)	(3,3,96)	0
$Block_6$							
Conv2D(3×3)@64	(n_5, n_5,64)	(3,3,64)	55360				
ActivationReLU˜	(n_5, n_5,64)	(3,3,64)	0				
Maxpool2D(2×2)	(n_6, n_6,64), $n_6 = n_5/2$	(1,1,64)	0				
BatchNorm	(n_6, n_6,64)	(1,1,64)	256				
Dropout	(n_6, n_6 ,64)	(1,1,64)	0				
Layer	**Output Shape**	**Image Size**	**Parameter**				
Flatten	(1, $n_6 \times n_6 \times 64$)	1, 64	0				
Dense	(1, 256)	(1, 256)	(1 + 64)× 256 =16640				
Batch Normalization	(1, 256)	(1, 256)	1024				
Activation Relu	(1, 256)	(1, 256)	0				
Dropout	(1, 256)	(1, 256)	0				
Dense	(1, 256)	(1, 256)	(256+1)×256 =65792				
Batch Normalization	(1, 256)	(1, 256)	1024				
Activation Relu	(1, 256)	(1, 256)	0				
Dropout	(1, 256)	(1, 256)	0				
Dense	(1, 3)	(1, 3)	(256+1)×3 =771				
Total Parameters for The Input Image Size			310247				
Total Number of Trainable Parameters:			308455				
Non-trainable params:			1792				

been provided in Fig. 4.8. The detailed description of this architecture with input-output, hidden layers, output shapes of the convoluted image, input image size, and parameters generated at each layer has been shown in Table 4.2, respectively, for better understanding and clarity of the model.

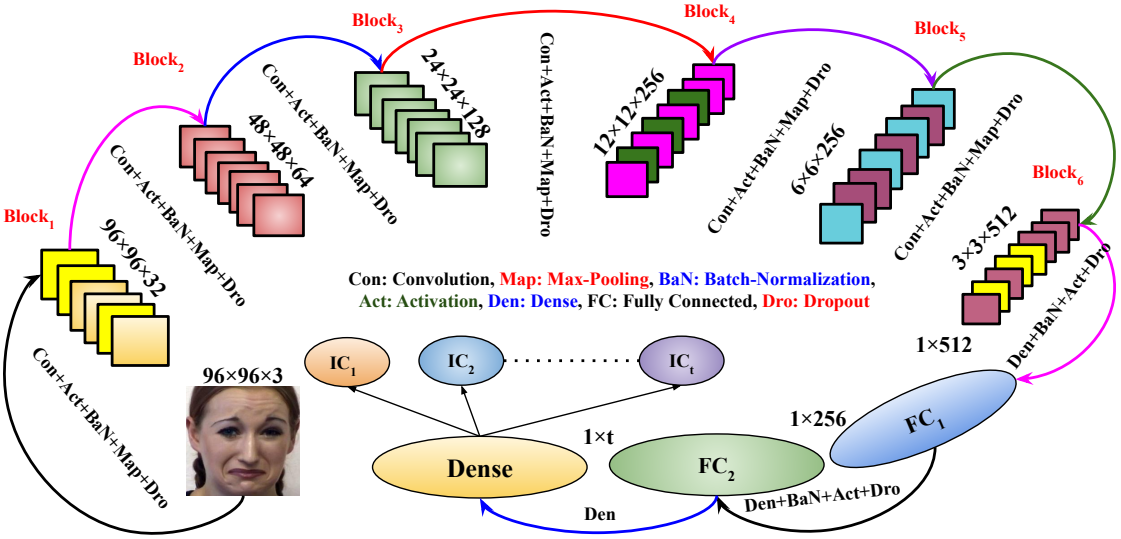


FIGURE 4.9: Proposed CNN_3 architecture.

A key difference between CNN_1 and CNN_2 is that, unlike CNN_2, CNN_1 does not include dropout layers in its CNN blocks, making it more prone to overfitting. Additionally, CNN_2 consists of more CNN blocks than CNN_1, which enhances its feature extraction capabilities; early blocks detect edges, intermediate blocks recognize features related to textures, deeper blocks identify the features of object parts, and the deepest blocks capture full object-level features. This also allows for more complex nonlinear transformations and allowing a deeper decision boundary that contributes to better classification performance.

4.2.7 Proposed CNN_3 System

The CNN_3 architecture consists of six blocks, similar to the CNN_2 model, but each block includes a sequence of layers: a Convolutional Layer, an Activation Layer, a BatchNormalization Layer, a MaxPooling Layer, and a Dropout Layer. After passing through these blocks, the output is subjected to a Flattened Layer, followed by two Fully Connected Layers, each containing a Dense Layer and a Dropout Layer. The final layer in the architecture is an output layer, which also includes a Dense Layer. The 'Adam' optimizer is employed during the compilation process to optimize the entire framework. Proposed CNN_3 architecture is shown in Fig. 4.9, and the corresponding parameter list is given in Table 4.3.

Various specifications of these models, along with details of the input-output hidden layers, provide a clearer understanding and transparency of the model structure. Once training with the CNN_3 framework is complete, the trained model f_{CNN_3} is obtained. CNN_3 is considered better than CNN_2 mainly due to the more optimal ordering of its layers, particularly the placement of Batch Normalization before Max-Pooling. In CNN_3, Batch Normalization is applied immediately after the activation function, helping to stabilize and accelerate training by normalizing the activations before spatial downsampling occurs in MaxPooling. This sequence ensures that the inputs to the pooling layer maintain consistent distributions, allowing for more effective feature extraction and reducing the risk of information loss. In contrast, CNN_2 applies MaxPooling before Batch Normalization, which can diminish the regularizing and stabilizing benefits of normalization, since down-sampled data may already discard key spatial information. Thus, CNN_3 offers improved training dynamics and potentially better generalization performance.

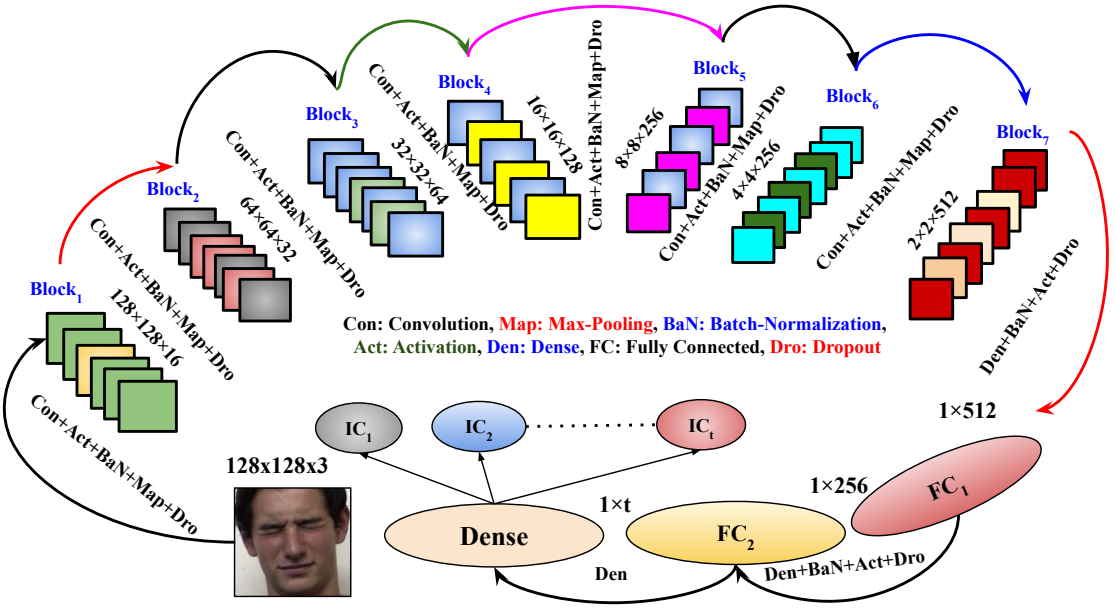


FIGURE 4.10: Proposed CNN_4 architecture.

TABLE 4.3: Parameters list of CNN_3 architecture.

Layers	Outputshape	ImageSize	Parameters
Block1			
Conv2D(3x3)@32	(n, n, 32)	(96, 96, 32)	896
Activation ReLU	(n, n, 32)	(96, 96, 32)	0
BatchNorm	(n, n, 32)	(96, 96, 32)	4x64 = 256
Maxpool2D (2x2)	(n1, n1, 32), n1=n/2	(48, 48,32)	0
Dropout	(n1, n1, 32)	(48, 48,32)	0
Block2			
Conv2D(3x3)@64	(n1, n1, 64)	(48, 48,64)	((3x3x32)+1)x 64=18496
Activation ReLU	(n1, n1, 64)	(48, 48,64)	0
BatchNorm	(n1, n1, 64)	(48, 48,64)	4x64=256
Maxpool2D (2x2)	(n2, n2, 64), n2=n1/2	(24, 24, 64)	0
Dropout	(n2, n2, 64)	(24, 24, 64)	0
Block3			
Conv2D (3x3)@128	(n2,n2,128)	(24,24,128)	((3x3x64)+1)x128=73856
Activation ReLU	(n2,n2,128)	(24,24,128)	0
BatchNorm	(n2,n2,128)	(24,24,128)	4x128=512
Maxpool2D (2x2)	(n3,n3,128), n3=n2/2	(12, 12, 128)	0
Dropout	(n3,n3,128)	(12, 12, 128)	0
Block4			
Conv2D (3x3)@256	(n3, n3, 256)	(12, 12, 256)	((3x3x128)+1)x256=295168
Activation ReLU	(n3, n3, 256)	(12, 12, 256)	0
BatchNorm	(n3, n3, 256)	(12, 12, 256)	4x256=1024
Maxpool2D (2x2)	(n4, n4, 256), n4=n3/2	(6, 6, 256)	0
Dropout	(n4, n4, 256)	(6, 6, 256)	0
Block5			
Conv2D(3x3)@256	(n4, n4, 256)	(6, 6, 256)	((3x3x256)+1)x256=590080
ActivationReLU	(n4, n4, 256)	(6, 6, 256)	0
BatchNorm	(n4, n4, 256)	(6, 6, 256)	4x256=1024
Maxpool2D(2x2)	(n5, n5, 256), n5=n4/2	(3, 3, 256)	0
Dropout	(n5, n5, 256)	(3, 3, 256)	0
Block6			
Conv2D(3x3)@512	(n5, n5,512)	(3, 3, 512)	((3x3x256)+1)x512=1180160
Activation Re LU-	(n5, n5,512)	(3, 3, 512)	0
BatchNorm	(n5, n5,512)	(3, 3, 512)	4x512=2048
Maxpool2D(2x2)	(n6, n6,512), n6=n5/2	(1,1,512)	0
Dropout	(n6, n6,512)	(1,1,512)	0
Layers	**Outputshape**	**ImageSize**	**Parameters**
Flatten	(1, n6 X n6 X 512)	(1,512)	0
Dense	(1,512)	(1,512)	(1 + 512)x 512=262656
BatchNorm	(1,512)	(1,512)	0
ActivationReLU	(1,512)	(1,512)	0
Dropout	(1,512)	(1,512)	0
Dense	(1,256)	(1,256)	(1 + 512)x 256=131328
BatchNorm	(1,256)	(1,256)	0
ActivationReLU	(1,256)	(1,256)	0
Dropout	(1,256)	(1,256)	0
Dense	(1,5)	(1,5)	(256+1)x5=1285
Total Parameters for The Input Image Size:			**2558917**
Total Number of Trainable Parameters:			**2556421**
Non-trainable params:			**2496**

TABLE 4.4: Parameters list of CNN_4 architecture.

Layers	Outputshape	ImageSize	Parameters
Block1			
Conv2D(3x3)@16	(n, n, 16)	(128, 128, 16)	448
Activation ReLU	(n, n, 16)	(128, 128, 16)	0
BatchNorm	(n, n, 16)	(128, 128, 16)	4x16 = 64
Maxpool2D (2x2)	(n1, n1, 16), n1=n/2	(64, 64, 16)	0
Dropout	(n1, n1, 16)	(64, 64, 16)	0
Block2			
Conv2D(3x3)@32	(n1, n1, 32)	(64, 64, 32)	((3x3x16)+1)x 32=4640
Activation ReLU	(n1, n1, 32)	(64, 64, 32)	0
BatchNorm	(n1, n1, 32)	(64, 64, 32)	4x32=128
Maxpool2D (2x2)	(n2, n2, 32), n2=n1/2	(32, 32, 32)	0
Dropout	(n2, n2, 32)	(32, 32, 32)	0
Block3			
Conv2D (3x3)@64	(n2,n2,64)	(32, 32, 64)	((3x3x32)+1)x64=18496
Activation ReLU	(n2,n2,64)	(32, 32, 64)	0
BatchNorm	(n2,n2,64)	(32, 32, 64)	4x64=256
Maxpool2D (2x2)	(n3, n3, 64), n3=n2/2	(16, 16, 64)	0
Dropout	(n3, n3, 64)	(16, 16, 64)	0
Block4			
Conv2D (3x3)@128	(n3,n3, 128)	(16, 16, 128)	((3x3x64)+1)x128=73856
Activation ReLU	(n3,n3, 128)	(16, 16, 128)	0
BatchNorm	(n3,n3, 128)	(16, 16, 128)	4x128=512
Maxpool2D (2x2)	(n4, n4, 128), n4=n3/2	(8, 8, 128)	0
Dropout	(n4, n4, 128)	(8, 8, 128)	0
Block5			
Conv2D(3x3)@256	(n4, n4, 256)	(8, 8, 256)	((3x3x128)+1)x256=295168
ActivationReLU	(n4, n4, 256)	(8, 8, 256)	0
BatchNorm	(n4, n4, 256)	(8, 8, 256)	4x256=1024
Maxpool2D(2x2)	(n5, n5, 256), n5=n4/2	(4, 4, 256)	0
Dropout	(n5, n5, 256)	(4, 4, 256)	0
Block6			
Conv2D(3x3)@256	(n5, n5 , 256)	(4, 4, 256)	((3x3x256)+1)x256=590080
Activation Re LU-	(n5, n5 , 256)	(4, 4, 256)	0
BatchNorm	(n5, n5 , 256)	(4, 4, 256)	4x256=1024
Maxpool2D(2x2)	(n6, n6 , 256), n6=n5/2	(2, 2, 256)	0
Dropout	(n6, n6 , 256)	(2, 2, 256)	0
Block7			
Conv2D(3x3)@512	(n6, n6 , 512)	(2, 2, 512)	((3x3x256)+1)x512=1180160
Activation Re LU-	(n6, n6 , 512)	(2, 2, 512)	0
BatchNorm	(n6, n6 , 512)	(2, 2, 512)	4x512=2048
Maxpool2D(2x2)	(n7, n7 , 512), n6=n5/2	(1, 1, 512)	0
Dropout	(n7, n7 , 512)	(1, 1, 512)	0
Layers	**Outputshape**	**ImageSize**	**Parameters**
Flatten	(1, n7 X n7 X 512)	(1,512)	0
Dense	(1,512)	(1,512)	(1 + 512)x 512=262656
BatchNorm	(1,512)	(1,512)	0
ActivationReLU	(1,512)	(1,512)	0
Dropout	(1,512)	(1,512)	0
Dense	(1,256)	(1,256)	(1 + 512)x 256=131328
BatchNorm	(1,256)	(1,256)	0
ActivationReLU	(1,256)	(1,256)	0
Dropout	(1,256)	(1,256)	0
Dense	(1,5)	(1,5)	(256+1)x5=1285
Total Parameters for The Input Image Size:			**2563173**
Total Number of Trainable Parameters:			**2560645**
Non-trainable params:			**2528**

4.2.8 Proposed CNN_4 System

The CNN_4 framework is proposed, which has seven blocks. These individual blocks also have a convolutional layer, an Activation layer, a BatchNormalization layer, a

MaxPooling layer, and a Dropout layer back to back, then three flatten layer,s and at the end the output layer. Each flattened layer appears after the combination of a dense layer, a batch normalization layer containing an activation function 'Relu', and a dropout layer. In the end, the CNN_4 architecture includes a dense layer, the final output layer. At the time of compilation of the proposed CNN_4 framework for optimization operation, the 'Adam' optimizer is used. After completion of the training process with the CNN_4 architecture, the trained f_{CNN_4} model is obtained. The proposed CNN_4 architecture is illustrated in Fig. 4.10 and the parameter list is given in Table 4.4.

The organization of the layer within the CNN blocks of CNN_3 and CNN_4 is the same; however, CNN_4 contains a greater number of CNN blocks compared to CNN_3. As a result, CNN_4 is able to extract deeper and more complex features than CNN_3. In this chapter, four CNN architectures have been proposed. These architectures have some differences from each other. The architectural comparison of different CNN architectures is showcased in Table 4.5.

4.3 Experiments and Results

This chapter performs experiments with images of the facial region of patients to detect pain intensity by analyzing expressions. The databases used in this chapter are as follows.

i. **UNBC Shoulder Pain Database** (ID_{UNBC}): This database comprises 129 participants (63 male and 66 female), where the participants have shoulder pain, and three physiotherapy clinics have identified their problems. The videos were recorded on the campus of McMaster University. During data acquisition, participants are assumed to have suffered from tendinitis, bursitis, rotator cuff injuries, arthritis, bone spurs, dislocation, subluxation, impingement syndromes, capsulitis, and these causes shoulder pain. Here, from each video, the images are extracted, where each image is assigned a label from the 'No pain' to 'High-intensity pain' classes. In this work, we have categorized

TABLE 4.5: Comparison of the CNN architectures as models for the $\text{DLPSA}_{\text{image}}$ system.

Model	**No. of Convolution Block**	**Layer Arrangement in each block**
$\mathbf{CNN}_1$	5	Convolution
		Activation
		Maxpooling
		BatchNormalization
$\mathbf{CNN}_2$	6	Convolution
		Activation
		Maxpooling
		BatchNormalization
		Dropout
$\mathbf{CNN}_3$	6	Convolution
		Activation
		BatchNormalization
		Maxpooling
		Dropout
$\mathbf{CNN}_4$	7	Convolution
		Activation
		BatchNormalization
		Maxpooling
		Dropout

TABLE 4.6: Description of the employed ID_{UNBC}.

3-Class problem		
Pain	**Class**	**Sample**
PI_0	IC_1	40029
PI_1	IC_2	2909
PI_2	IC_3	5460
2-Class problem		
Pain	**Class**	**Sample**
PI_0	IC_1	40029
PI_1	IC_2	8369

these images into three pain categories (already discussed in the previous chapters). The description of the samples of these three classes has been demonstrated in Table 4.6. Some images of this database are shown in Fig. 4.11.

FIGURE 4.11: Some sample images of ID_{UNBC} (3-Class).

TABLE 4.7: Dataset description of both VD_{BioVid} for Approach-1, Approach-2 and Approach-3

Approach	Number of Subject (Individuals)	Number of Videos	Number of frame from each Video	Number of class	Total Number of Images
Approach-1	5	20	14	5	7000
Approach-2	10	20	14	5	14000
Approach-3	20	20	14	5	28000

ii. **BioVid** (VD_{BioVid}): This dataset contains videos from 87 individuals, which are kept in Part A to Part E directories. In this study, only Part A directory of the VD_{BioVid} is considered. For Part A, the data is collected on three separate occasions, as outlined in Table 4.7. The database is classified into five distinct classes, including Baseline indicates no-pain (PI_0), and four levels of pain intensity: Pain Intensity-1 (PI_1), Pain Intensity-2 (PI_2), Pain Intensity-3 (PI_3), and Pain Intensity-4 (PI_4). The sample frames of the dataset have been demonstrated in Fig. 4.12. From this table, it has been observed that the videos of this dataset has been experimented using three approaches: (i) Approach-1, involving 5 subjects, 500 videos are used; (ii) Approach-2, with 10 subjects, 1,000 videos are used, (iii) Approach-3, which includes 20 subjects, 2,000 videos are considered. These videos are used to develop the proposed PSA_{image} system. For this purpose, only the image frames of Approach-1 videos have been considered. This subset serves as the image dataset for the experiments discussed in this chapter of the thesis.

In this work, during image preprocessing, the tree-structured part model is used to extract the face region and then normalize the image to F of size $200 \times 200 \times 3$ from

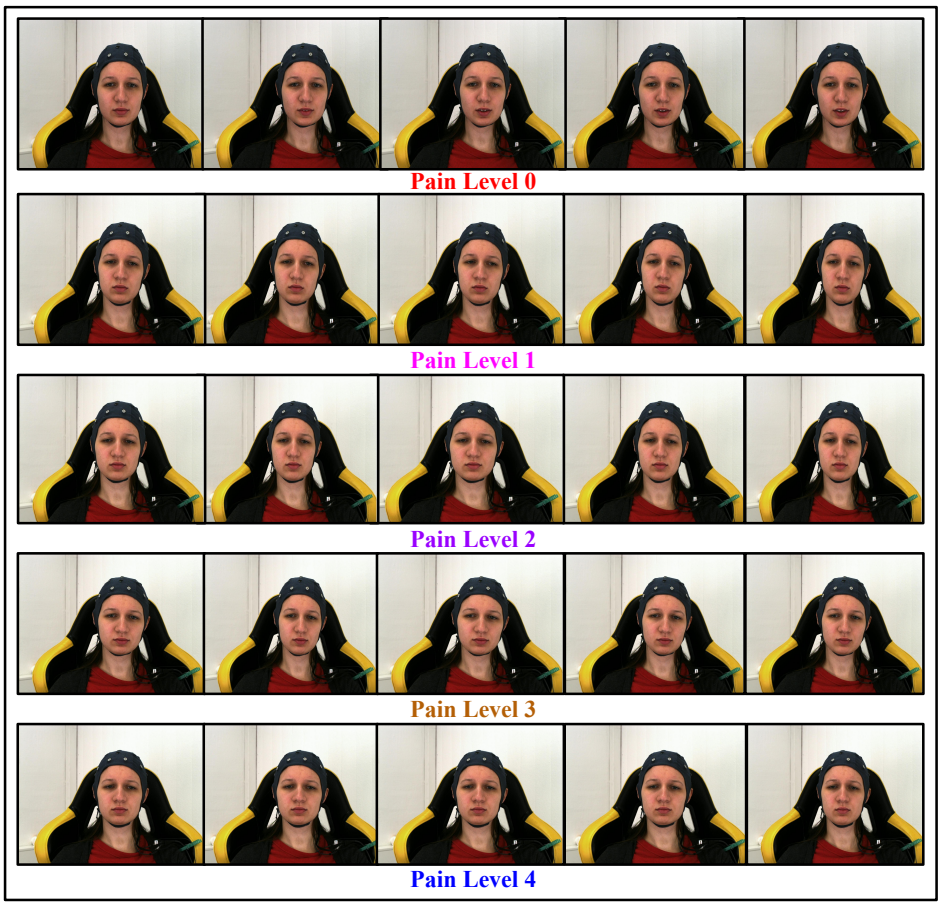


FIGURE 4.12: Sample video frames from VD_{BioVid} (5-Class).

TABLE 4.8: Feature dimension using different feature partitioning setup for $HANDPSA_{image}$ system.

Image Partitioning Setups	**Feature**	**Feature Dimension**	
		HOG	**LBP**
P_1	f_G	1×81	1×256
P_2	f_{HL}	1×162	1×512
P_3	f_{VL}	1×162	1×512
P_4	f_{LG}	1×324	1×1024

the input image. This work aims to identify the level of pain among two, three, and five classes.

The $HANDPSA_{image}$ system computes statistical features, HOG, and LBP to identify the level of pain. In Section 4.2.2, the variants of HOG and LBP have been demonstrated. This variation of features is summarized in Table 4.8.

Hence using these features, from the facial region F, $f_G^{HOG} \in R^{1\times 81}$ and $f_G^{LBP} \in R^{1\times 256}$ feature vectors are obtained. The feature vectors undergo classifiers such as LR [24], kNN [301], DT [302], RF, and SVM [303].

Each dataset is divided into 50% of data in the training set and 50% of data in the testing set. Then, a ten-fold cross-validation technique is used, and the average performance for testing data is reported for the proposed system. The performance of the proposed system for the ID_{UNBC} 2-class problem due to these classifiers has been shown in Table 4.9.

TABLE 4.9: Performance of $\text{HANDPSA}_{\text{image}}$ System for 2-class pain detection on $\text{ID}_{\textbf{UNBC}}$ using P_1.

Classifier	**HOG**				**LBP**			
	Accuracy	**F1-Score**	**Precision**	**Recall**	**Accuracy**	**F1-Score**	**Precision**	**Recall**
LR	74.28	0.7267	73.79	73.96	74.39	0.7266	73.82	73.92
kNN	71.33	0.7045	70.81	71.15	73.56	0.7285	73.11	73.28
DT	75.44	0.7413	74.67	74.87	73.62	0.7182	72.64	72.89
RF	74.45	0.7434	74.06	74.26	76.07	0.7597	75.48	75.74
SVM	75.85	0.7405	74.42	75.72	76.87	0.7551	75.31	76.14

Table 4.9 shows that the proposed $\text{HANDPSA}_{\text{image}}$ System has attained better performance for the Support SVM. The purpose of these feature extraction methods is to extract more and more local features such that the performance of the proposed system is increased. Also, from this table, we observe that the performance of SVM is better than all other classifiers. Therefore, for further experiments with the $\text{HANDPSA}_{\text{image}}$ System, we use SVM as the classifier. The performance due to these feature vectors for the ID_{UNBC} 2-class problem has been reported through the bar-graphs in Fig. 4.13, where the x-axis shows the partitioning scheme while the y-axis shows the accuracy of the SVM classifier.

It has been observed that the proposed system has achieved 78.34% accuracy using HOG, 80.29% accuracy using LBP feature for the ID_{UNBC} 2-class problem. The performance of these four feature schemes on the datasets ID_{UNBC} (3-class) and $\text{VD}_{\text{BioVid}}$ (5-class) is given in Table 4.10. From Fig. 4.13 and Table 4.10, it has been observed that for P_4, the performance is better for both HOG and LBP features when SVM is used as a classifier. Hence, P_4 statistical-based feature representation scheme with SVM as the classifier has been adopted to represent the $\text{HANDPSA}_{\text{image}}$ System.

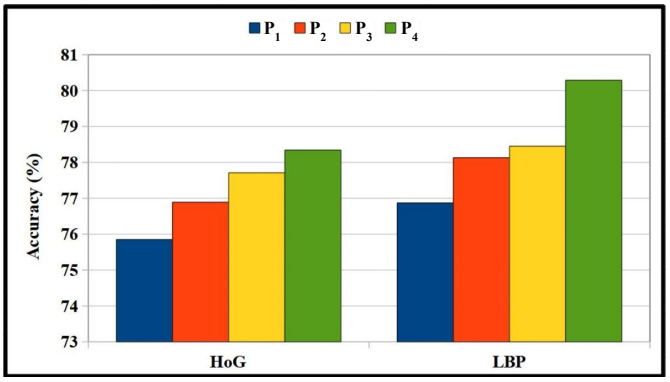


FIGURE 4.13: Performance of $\text{HANDPSA}_{\text{image}}$ System with SVM as classifier on ID_{UNBC} 2-class pain detection.

TABLE 4.10: Performance of the proposed $\text{HANDPSA}_{\text{image}}$ System on ID_{UNBC} (3-class) and $\text{VD}_{\text{BioVid}}$ (5-class).

	HOG				**LBP**			
ID_{UNBC} (3-class)								
Scheme	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Accuracy**	**F1-Score**	**Precision**	**Recall**
P_1	77.85	0.7533	76.69	77.13	78.71	0.7702	77.39	78.14
P_2	78.26	0.7629	77.03	77.57	79.16	0.7753	78.61	78.82
P_3	78.91	0.7704	77.38	78.27	79.68	0.7825	78.43	79.29
P_4	79.14	0.7751	78.64	78.46	80.08	0.7891	79.22	79.63
$\text{VD}_{\text{BioVid}}$ (5-class)								
Scheme	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Accuracy**	**F1-Score**	**Precision**	**Recall**
P_1	31.74	0.3054	30.31	31.24	32.14	0.3152	31.28	31.86
P_2	32.37	0.3122	31.54	31.82	32.79	0.3212	31.91	32.42
P_3	32.89	0.3179	31.73	32.16	33.57	0.3286	32.73	33.36
P_4	33.46	0.3258	32.17	32.73	34.48	0.3319	33.67	33.52

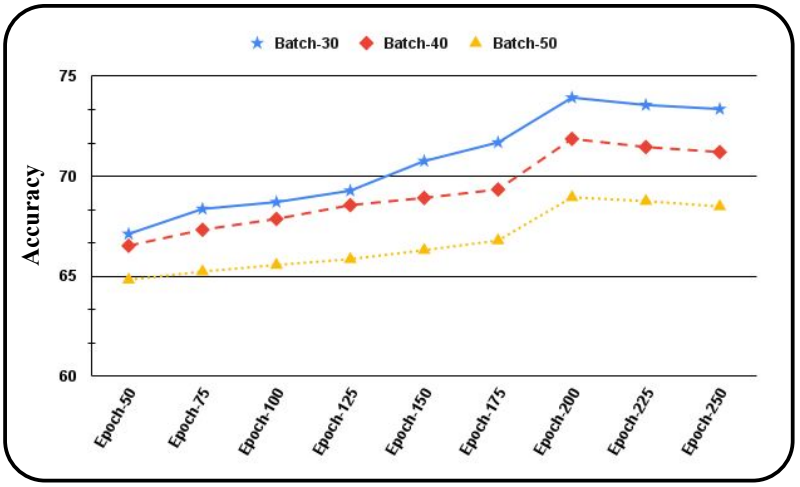


FIGURE 4.14: Effectiveness of batch vs epochs for the proposed $\text{DLPSA}_{\text{image}}$ System on ID_{UNBC}.

In the $\text{DLPSA}_{\text{image}}$ System, the input image is preprocessed, and the facial region is extracted by the TSPM model. The extracted face region F is normalized. The size of the face region is 48×48 while the batch size and the number of epochs vary. Fig. 4.14 demonstrates the effectiveness of batch sizes and the number of epochs over the performance of the proposed image-based sentiment analysis system for the ID_{UNBC}. From Fig. 4.14, it has been observed that the performance improves with increasing epochs, and for the batch size 30, the performance is better. For further experiments, we have employed a batch size of 30 on training samples with 200 epochs for learning the parameters of the proposed CNN_1 architecture.

With the above setting, the performance of the proposed $\text{DLPSA}_{\text{image}}$ System with CNN_1 has been shown in Table 4.11. Hence, from the performance, it has been observed that the proposed system has achieved performance using CNN_1.

TABLE 4.11: Performance of the proposed $\text{DLPSA}_{\text{image}}$ System using CNN_1.

Dataset Used	**Problem Type**	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Training time**	**Testing time**
ID_{UNBC}	3-Class	80.61	0.7946	79.64	79.43	41.29 sec.	0.03 sec.
$\text{VD}_{\text{BioVid}}$	5-Class	35.21	0.3346	34.21	34.74	38.41 sec.	0.03 sec.

Now, the original image F is resized to 96×96 image size. Then the resized face images from the training data set to fed through the proposed CNN_2 architecture Fig. 4.8), where the input to the architecture is $96 \times 96 \times 3$ image. The CNN_2 architecture is trained in such a way that it would perform both feature computation and pain classification tasks. For a better understanding of the functionality of these architectures, at first, we have started the experiment by training the CNN architecture with training samples. In contrast, the performance of the trained CNN_2 model is evaluated using the remaining testing samples. Both Fine-tuning and Transfer Learning techniques have been employed here to improve the performance of the proposed system. Hence, the proposed $\text{DLPSA}_{\text{image}}$ System performance has been demonstrated in Table 4.12. These performances have been reported in terms of accuracy, F1-score, training time, and testing time, respectively.

From Table 4.12, it has been observed that the proposed system attains 82.07% accuracy for ID_{UNBC} 3-class problem, and 37.48% accuracy for $\text{VD}_{\text{BioVid}}$ 5-class problems. The CNN_2 architecture performs better than the CNN_1 because of a more optimized and considered placement of layers within every convolutional block. Both

TABLE 4.12: Performance of the proposed $\text{DLPSA}_{\text{image}}$ System using CNN_2.

Dataset Used	Problem Type	Accuracy	F1-Score	Precision	Recall	Training time	Testing time
ID_{UNBC}	3-Class	83.33	0.8207	82.58	82.89	41.29 sec.	0.03 sec.
$\text{VD}_{\text{BioVid}}$	5-Class	37.48	0.3522	36.28	36.84	26.31 sec.	0.03 sec.

architectures employ convolution, activation, pooling, and normalization, but the CNN_2 wisely incorporates Dropout layers in every block, increasing regularization and avoiding early overfitting within the network. Compared to that, the CNN_1 architecture has no dropout inside the blocks, which may affect CNN_1 with overfitting. These design options in the CNN_2 model enhance learning dynamics, feature richness, and generalization towards better performance.

Using the CNN_3 architecture, here also the images are resized to fixed dimensions of $96 \times 96 \times 3$ to obtain normalized facial expressions, denoted as F. The proposed CNN_3 architecture (illustrated in Fig. 4.3) is applied to these normalized images. This framework is trained to compute features and classify pain sentiment, with the trained model f_{CNN_3} evaluated on test samples to validate performance. The training-testing split follows a 50-50% ratio for both the ID_{UNBC} 3-class and $\text{VD}_{\text{BioVid}}$ 5-class datasets. The performance of CNN_3 is reported in Table 4.13.

TABLE 4.13: Performance of the proposed $\text{DLPSA}_{\text{image}}$ System using CNN_3.

Dataset Used	Problem Type	Accuracy	F1-Score	Precision	Recall	Training time	Testing time
ID_{UNBC}	3-Class	84.72	0.8273	83.46	83.27	44.32 sec.	0.03 sec.
$\text{VD}_{\text{BioVid}}$	5-Class	48.23	0.4581	46.39	46.72	38.41 sec.	0.03 sec.

The performance reported in Table 4.13 shows that CNN_3 delivers strong results for the 3-class ID_{UNBC}, attaining 84.72% accuracy and an excellent 0.8273 F1-score. These metrics demonstrate the effectiveness of the model in handling simpler $\text{PSA}_{\text{image}}$ System tasks, where it achieves both reliable classification and efficient learning. However, performance metrics decline noticeably when applied to the more challenging 5-class $\text{VD}_{\text{BioVid}}$, with accuracy falling to 48.23% and F1-score dropping to 0.4581. This performance gap suggests difficulties in managing finer-grained classification, potentially due to class imbalance or overlapping feature distributions. Notably, the model maintains consistently fast testing times of 0.03 seconds for both datasets, confirming its suitability for real-time applications.

These results collectively indicate that while CNN_3 performs well for simpler classification tasks, architectural improvements, such as advanced feature extraction methods or class-balancing techniques, could enhance its capability to handle more complex, multi-class problems. The CNN_3 achieves superior performance due to its optimized layer arrangement (Conv → Activation → BatchNorm → MaxPool → Dropout). The key advantage lies in the placement of the BatchNormalization before MaxPooling, which normalizes activations prior to downsampling, stabilizing gradients and preserving important features during pooling. This contrasts with CNN_1 (lacking Dropout entirely) and CNN_2 (where BatchNorm occurs after pooling), where unnormalized pooling can amplify noise and degrade feature quality. Additionally, strategic Dropout placement of CNN_3 after pooling effectively regularizes the already-normalized features, while similar Dropout of CNN_2 is less effective due to suboptimal BatchNorm positioning. The combined effect of proper normalization before dimensionality reduction and well-placed regularization gives CNN_3 better training stability, faster convergence, and improved generalization compared to both CNN_1 and CNN_2 architectures.

For CNN_4, the preprocessing phase similarly focuses on facial expression extraction but resizes images to $128 \times 128 \times 3$, producing normalized inputs F. The architecture (Fig. 4.4) processes these higher-resolution images, aiming to enhance feature learning precision. Like CNN_3, the trained model f_{CNN_4} is evaluated on a 50-50% split of the ID_{UNBC} and VD_{BioVid} to assess its functionality and performance. The performance of CNN_4 is reported in Table 4.14.

TABLE 4.14: Performance of the proposed $DLPSA_{image}$ System using CNN_4.

Dataset Used	**Problem Type**	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Training time**	**Testing time**
ID_{UNBC}	3-Class	85.29	0.8386	84.42	84.73	46.19 sec.	0.03 sec.
VD_{BioVid}	5-Class	48.45	0.4617	47.31	47.67	39.06 sec.	0.03 sec.

From Table 4.14, CNN_4 demonstrates strong performance on the simpler 3-class ID_{UNBC} dataset with 85.29% accuracy and 0.8386 F1-score, indicating effective feature learning and classification for coarse-grained pain sentiment analysis. However, its performance significantly declines on the more complex 5-class VD_{BioVid}, achieving only 48.45% accuracy and 0.4617 F1-score, which suggests challenges in handling fine-grained classification tasks with greater class overlap or data imbalance. While the model maintains efficient testing times (0.03 seconds) across both

TABLE 4.15: Summary of the performance of different schemes of the proposed $\text{DLPSA}_{\text{image}}$ System.

Methodology	**Accuracy (%) 3-Class Problem (ID_{UNBC})**	**Accuracy (%) 5-Class Problem ($\text{VD}_{\text{BioVid}}$)**
$\textbf{CNN}_1$	81.68	35.78
$\textbf{CNN}_2$	83.33	37.48
$\textbf{CNN}_3$	84.72	48.22
$\textbf{CNN}_4$	85.29	48.45

datasets, the stark contrast in results highlights the impact of problem complexity on performance, emphasizing the need for architectural improvements or additional techniques like attention mechanisms or data augmentation to enhance its capability for multi-class scenarios. The performance variation across CNNs under different dataset sizes can be attributed to their layer arrangements and capacity to balance feature extraction with generalization. CNN_4 excels with 20 subjects' images due to its deeper, structured architecture (16→32→64→128→256→512 filters), which efficiently captures complex patterns in large datasets without overfitting, thanks to consistent BatchNorm and Dropout. However, CNN_3 outperforms with smaller datasets because its irregular filter progression (64→96→64) and moderate parameter count (475,875) prevent over-parameterization, making it more adaptable to limited data. In contrast, the larger capacity of CNN_4 (2.56M parameters) overfits on smaller datasets. Thus, CNN_3 strikes a better balance for images of a smaller number of subjects, whereas CNN_4 leverages its depth for images of a larger number of subjects. From all the above experimental results for various CNN architectures, it has been observed that the CNN_3 and CNN_4 outperform other CNN_2 and CNN_1 architectures due to proper layer organization and depth of CNN blocks. The comparative performance analysis reveals a clear progression in effectiveness across the CNN architectures. CNN_1 establishes baseline performance with 81.68% accuracy on the 3-class ID_{UNBC} dataset and 35.78% on the 5-class $\text{VD}_{\text{BioVid}}$, while CNN_2 shows moderate improvement (83.33% and 37.48% respectively). The most significant leap occurs with CNN_3, which achieves 84.72% (ID_{UNBC}) and 48.22% ($\text{VD}_{\text{BioVid}}$) accuracy, demonstrating enhanced capability in handling both simpler and more complex classification tasks. CNN_4 further refines this performance, reaching 85.29% and 48.45% accuracy, marking it as the most effective architecture. This consistent improvement

across versions highlights the impact of architectural refinements, with the most notable gains appearing in the challenging 5-class problem, where both CNN_3 and CNN_4 show approximately 10% accuracy improvement over CNN_2, suggesting their superior feature extraction and classification capabilities for complex, fine-grained tasks. The performance trends indicate that while all architectures handle the 3-class problem reasonably well, the later versions (CNN_3 and CNN_4) demonstrate particular effectiveness in addressing the difficulties inherent in multi-class classification. The summary of the results of all the CNN architectures is mentioned in Table 4.15 with respect to 3-Class ID_{UNBC} and 5-Class VD_{BioVid}.

The Fig. 4.15 and Fig. 4.16 contain confusion matrices which represent the performance for ID_{UNBC} and VD_{BioVid} for 3-class and 5-class classification problems, respectively. From different applied classifiers in $HANDPSA_{image}$ System, SVM, along with P_4 feature representation of LBP, has been considered as it produces better results. In case of $DLPSA_{image}$ System for feature representation and classification, CNN_4 has been considered.

The confusion matrices in Fig. 4.15 show how well the SVM and CNN_4 models categorized data into three groups: IC_1, IC_2, and IC_3. The SVM model has done a good job identifying IC_1 cases, correctly predicting 15,837 of them, but it often confused them with IC_2 (3,084 times) and IC_3 (1,093 times). It also did reasonably well with IC_3, getting 2,331 right, though it struggled with IC_2, correctly identifying only 1,188 cases. On the other hand, the CNN_4 model was more accurate with IC_1, correctly classifying 17,791 instances, but it had more trouble with IC_2 and IC_3, with just 781 and 2,153 correct predictions, respectively. Both models often mixed up IC_2 and IC_3, but this was more noticeable in the results of CNN_4. In short, SVM performed better for IC_3, while CNN_4 showed higher accuracy for NAG but faced challenges with IC_2 and IC_3.

The confusion matrices in Fig. 4.16 compare how well the SVM and CNN_4 models classify data into five groups: IC_1, IC_2, IC_3, IC_4, and IC_5. The SVM model had moderate success, making the most correct predictions for IC_1 (241) and IC_5 (237), but often got confused, especially between BL and IC_4 (127 times) and BL and IC_5 (132 times). The CNN_4 model performed better overall, with strong results for IC_3 (370 correct) and IC_4 (409 correct), but it also made mistakes, such as mixing up BL with IC_5 (126 times) and IC_4 with IC_5 (157 times). While CNN_4 generally

Actual \ Predicted	IC_1	IC_2	IC_3
IC_1	79.13	15.41	5.46
IC_2	14.72	81.71	3.58
IC_3	13.04	1.58	85.38

Confusion matrix for SVM

Actual \ Predicted	IC_1	IC_2	IC_3
IC_1	88.89	6.47	4.64
IC_2	17.06	53.71	29.23
IC_3	6.92	14.21	78.86

Confusion matrix for CNN_4

FIGURE 4.15: Confusion matrices in percentage for the proposed $HANDPSA_{image}$ System and $DLPSA_{image}$ System on ID_{UNBC}.

gave more accurate results than SVM, both models struggled to clearly separate categories that share similar features.

Actual \ Predicted	IC_1	IC_2	IC_3	IC_4	IC_5
IC_1	34.43	17.86	10.71	18.14	18.86
IC_2	19.00	35.29	16.00	18.00	11.71
IC_3	14.00	17.43	36.43	15.86	16.29
IC_4	16.71	11.00	19.71	33.43	19.14
IC_5	11.57	19.14	18.29	17.14	33.86

Confusion matrix for SVM

Actual \ Predicted	IC_1	IC_2	IC_3	IC_4	IC_5
IC_1	49.00	11.00	13.29	8.71	18.00
IC_2	16.14	43.57	13.14	15.43	11.71
IC_3	15.14	8.29	52.86	13.14	10.57
IC_4	10.86	14.00	9.14	58.43	7.57
IC_5	10.29	17.00	12.00	22.43	38.29

Confusion matrix for CNN_4

FIGURE 4.16: Confusion matrices in percentage for the proposed $HANDPSA_{image}$ System and $DLPSA_{image}$ System on VD_{BioVid}.

The performance of the proposed system is evaluated against different SoA methods, and it is reported in Table 4.16. To ensure consistency and fairness in evaluation, we implemented notable deep learning architectures such as VGG16 [15], ResNet50 [17], and Inception-v3 [304], along with approaches proposed by Werner et al. [65], Lucey et al. [239], Yuille [305], Simos [306], and Traue [65] and assessed their performance using the same training and testing protocols employed by the proposed system. It is important to note that the performance of our hand-crafted system is better than

TABLE 4.16: Performance comparison for the proposed PSA_{image} System of this Chapter.

Method	**Accuracy (%)**	
	ID_{UNBC}	**VD_{BioVid}**
Vgg16 [15]	76.84	22.34
ResNet50 [17]	79.32	19.56
Inception-v3 [304]	79.64	23.18
Werner et al. [65]	75.52	24.01
Yuille [305]	79.15	25.39
Simos [306]	75.67	24.14
Traue [65]	63.12	31.51
Lucey et al. [239]	81.84	24.78
$HANDPSA_{image}$ System	**80.08**	**34.48**
$DLPSA_{image}$ System (CNN_3)	**84.72**	**48.22**
$DLPSA_{image}$ System (CNN_4)	**85.29**	**48.45**

the performance of the SoA methods for both datasets (except Lucey et al. [239] on ID_{UNBC}). Also, note that $DLPSA_{image}$ System with CNN_3 (and CNN_4) shows superior performance compared to the SoA methods.

4.4 Conclusions

This chapter discusses image-based $\text{PSA}_{\text{image}}$ System. The implementation of this system begins with the preprocessing technique applied to the input image, which extracts the facial portion as a region of interest. The extracted facial region then undergoes two systems: $\text{HANDPSA}_{\text{image}}$ and $\text{DLPSA}_{\text{image}}$. In the $\text{HANDPSA}_{\text{image}}$ System, the preprocessed facial image undergoes a feature extraction followed by classification. In this system, two feature extraction techniques, namely the Local Binary Pattern (LBP) and the Histogram of Oriented Gradients (HOG), are utilized to extract features from the facial region. The extracted features then undergo a classification task. On the other hand, the preprocessed image, in the $\text{DLPSA}_{\text{image}}$ system, is processed by a deep model to identify the level of pain. In this system, four CNN architectures, such as CNN_1, CNN_2, CNN_3, and CNN_4, have been designed and implemented. Both the $\text{HANDPSA}_{\text{image}}$ and the $\text{DLPSA}_{\text{image}}$ systems have been tested on ID_{UNBC} (UNBC Shoulder Pain Database) and $\text{VD}_{\text{BioVid}}$ (BioVid heat pain database). Through experiments and results, it has been observed that both CNN_3 and CNN_4 have achieved similar performance. So, perhaps both or either of these could be considered $\text{DLPSA}_{\text{image}}$ systems. These performances have obtained outstanding performance compared to all other competing methods, even surpassing the $\text{HANDPSA}_{\text{image}}$ system. ***In this chapter, the CNN_4 based $\text{DLPSA}_{\text{image}}$ System is considered the prediction model for the image-based pain sentiment analysis ($\text{PSA}_{\text{image}}$) System.*** This system will mitigate the limitations of the PSA_{text} and $\text{PSA}_{\text{audio}}$ Systems, providing valuable insights into variations in pain levels.

This methodology enhances the robustness of the system, as facial expressions are less susceptible to subjectivity or deliberate manipulation compared to text or audio. However, image-based analysis has limitations, primarily its static nature, which prevents the assessment of temporal dynamics, such as the duration and progression of pain expressions. To overcome this limitation, the next chapter will extend the framework by incorporating video-based dynamic signals. This advancement will enable continuous monitoring of pain intensity, temporal analysis of expression duration, and improved robustness through spatiotemporal feature extraction. By

integrating video data, the system will capture the evolving nature of pain experiences, bridge the gap left by static image analysis, and provide a more comprehensive understanding of patient conditions.

Chapter 5

Video-Based Pain Sentiment Analysis

In this chapter, we have presented a video-based pain analysis (PSA_{video}) system to address various challenges highlighted in the previous chapter. The work of this chapter focuses on analyzing the facial expressions of patients captured in individual frames to continuously monitor the progression of their pain over time. The images are inherently discrete, making it difficult to capture subtle changes and trends. To overcome this issue, it is essential to shift from images to videos that allow us to analyze dynamic behavior, offering a richer and more continuous perspective. By working with video, we can capture significant spatio-temporal measurements of facial actions in much finer detail. This enables more accurate and minute tracking of pain expressions as they evolve. Video processing utilizes dynamic data to identify, interpret, and respond to real-world signals for real-time monitoring of patients. PSA_{video} system has improved accuracy by providing more contextual information than standard sensors.

In this chapter, the assessment of pain is entirely based on facial expressions. Frames containing these facial expressions are captured at regular intervals, ensuring that minimal changes in facial expressions over short time gaps are preserved and can be utilized effectively for the experiment. The facial region of each patient is detected and cropped from these frames, after which the frames undergo preprocessing. The features are then extracted from the preprocessed frames, and these features

are concatenated to create the final representation of the features. In the previous chapter, we noted that deep learning methods provide significantly better results than hand-crafted methods and therefore, in this work, for classification, fully connected networks are used, making the entire process from feature extraction to feature concatenation and classification an integrated procedure, called *PainCapsule*. To further enhance the accuracy of the classification, the proposed *PainCapsule* is passed through an attention network, which effectively handles long-range dependencies within the feature map, leading to another unified methodology called *PainAttentionCapsule*. To further improve the classification results, LSTM networks are applied to the features of *PainAttentionCapsule*, effectively addressing temporal dependencies within the feature map. The block diagram of the proposed $\text{PSA}_{\text{video}}$ System is illustrated in Fig. 5.1.

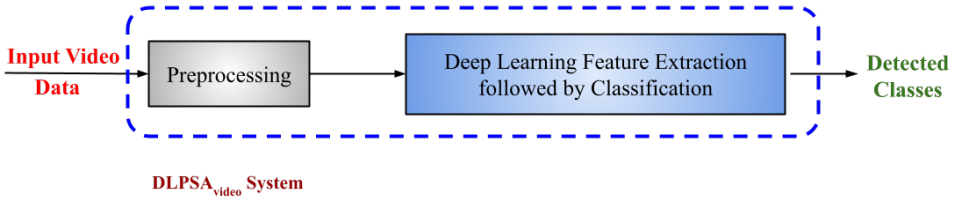


FIGURE 5.1: Block Diagram of the proposed $\text{PSA}_{\text{video}}$ system.

This chapter is organized as follows. Section 5.1 discusses literature review, Section 5.2 demonstrates the implementation of the $\text{PSA}_{\text{video}}$ system. Section 5.3 presents the experiments and the analysis of the results. Section 5.4 concludes the chapter.

5.1 Literature Review

Recent developments in video-based emotion analysis have transitioned from traditional techniques to advanced deep learning models, significantly enhancing performance and applicability. The foundational contributions of Ekman and Friesen [307] introduced the Facial Action Coding System (FACS), a pivotal tool to recognize emotions through facial muscle movements known as action units. Building on this, Zeng et al. [308] offered a comprehensive overview of early visual emotion recognition methods, identifying key challenges in feature extraction. The advent of CNNs marked a breakthrough, as demonstrated by Kahou et al. [309], whose EmoNets

framework enabled end-to-end emotion recognition directly from video data. This momentum continued with Mollahosseini et al. [310], who introduced AffectNet, one of the largest and most diverse databases for facial expression analysis. More recently, transformer-based models, such as the one proposed by Arriaga et al. [311], have shown improved capabilities in capturing temporal patterns, which is crucial for accurate emotion classification. Benchmark datasets, such as CK+ [312] and AFEW [313], have played a vital role in evaluating these methods in real-world conditions. The latest advances, including the work by Zhao et al. [314], emphasize the power of multimodal fusion, combining facial expressions, body movements, and contextual cues to achieve robust and reliable emotion analysis.

Ghazal et al. [315] introduced a method that utilizes Principal Component Analysis (PCA) to reduce the size of the dataset and extract significant facial features, followed by the application of a CNN for face identification. In their survey, Rasha M. et al. [316] reviewed various pain detection techniques, including multi-domain neural networks, CNN-LSTMs, and additive attributes. They also proposed a method for comparing feature extraction techniques such as Skin Conductance Response and Skin Conductance Level, using relevant features such as root mean square, mean values of local maxima and minima, as well as the mean absolute value for pain detection [317]. Patrick et al. [318] introduced a pain detection approach that combines both unimodal and multimodal concepts. Eshan et al. [319] compared a Random Forest classifier using Facial Activity Descriptors with the performance of a deep learning-based pain detection model. Kornprom et al. [320] presented a pain assessment method along with a pain intensity metric, using tools such as the Visual Analog Scale (VAS), Observer Rating Scale (ORS), Affective Motivation Scale (AMS), and Sensory Scale (SS). Leila et al. [321] developed a model for automatic pain detection using the VGG-Face model on the UNBC dataset. Jiang et al. [99] conducted experiments using the GLA-CNN approach for pain detection. Lastly, Ehsan et al. [322] proposed a pain classification system based on video data from the X-ITE dataset.

Ullah et al. [323] addressed several challenges faced during the implementation of Multimodal Sentiment Analysis (MSA) using text, image, audio and video data regularly posted on social media. The survey also highlighted existing and emerging

difficulties and opportunities in MSA research. Similarly, Soleymani et al. [324] discussed the challenges of MSA in various domains, including spoken reviews, video blogs, images, human-machine interaction, and human-human interaction systems, while also pointing out the opportunities in these areas. Rao et al. [325] employed speech-based LSTM features, kNNs, Bayesian networks, HMMs, and ANN to extract features from facial expressions and acoustic data, such as Gaussian mixture models and Mel frequency cepstral coefficients, using the RAVDESS audio dataset. Caschera et al. [80] employed a multimodal approach to classify emotions from speech and text in videos, demonstrating the automatic extraction of emotional information across various modalities and domain of interaction. Abdu et al. [326] presented a survey on multimodal video sentiment analysis using deep learning approaches, specifically highlighting multimodal sentiment analysis systems with a multimodal multi-utterance-based architecture.

Recent progress in video-based pain sentiment analysis has led to more intelligent systems that use 3D-CNNs to capture how pain unfolds in space and time in videos [19], while newer models like Vision Transformers (ViTs) with space-time attention go a step further by understanding long-term patterns in video frames better than traditional CNNs [327]. The UNBC-McMaster Shoulder Pain Archive remains the primary dataset for training and testing these models, thanks to its detailed frame-level AU labels [328]. However, datasets like EmoPain [329] and BioVid [257] now include additional signals, such as body movement and physiological data, for a more comprehensive analysis. To ease the burden of labeling every video frame, weakly supervised methods utilize multiple-instance learning to identify the most critical pain frames [330], and contrastive learning tools like SimCLR enhance the reliability of the model by comparing different augmented clips of the same pain sequence [261]. Graph neural networks (GNNs) help by mapping how facial features relate and change during pain using learnable connections [260], and neural ODEs allow tracking of pain levels over time, even when video frames are unevenly spaced [290].

However, models trained in controlled lab environments often do not perform well in real-world scenarios [271], which is why datasets like Aff-Wild2 Pain [272] have been created to reflect more natural settings. Cutting-edge hybrid models that combine CNNs and transformers [331] deliver top performance by integrating detailed feature

extraction with a global understanding of time, and test-time adaptation techniques enable these models to adjust to new environments on the fly [288]. Few-shot learning methods, such as prototypical networks [291], help detect rare pain expressions using very little labeled data, and federated learning enables different hospitals to collaborate on model training without sharing patient data [268]. To address data limitations, diffusion models generate realistic pain videos for training [269]. In contrast, causal discovery methods focus the models on real pain signals and disregard distracting factors such as individual identity [282]. Finally, the OMG-Pain Challenge encourages researchers to monitor how pain evolves in real-life conditions [283]. Multimodal transformers are now being used to integrate signals from facial expressions, voice, and body movement through cross-modal attention [284].

Bargshady et al. [332] present a vision-transformer-based model (ViViT) that processes facial expression video sequences to recognize acute pain, outperforming several conventional deep-learning baselines on standard pain datasets. Holden et al. surveys AI techniques [333] for animal pain assessment, linking species-specific facial action changes to algorithmic pipelines adapted from human facial-expression analysis. Holden et al. [334] described extensions of computer-vision-based facial pain detection systems toward postoperative and clinical monitoring scenarios, emphasizing real-time deployment issues such as robustness to pose, lighting, and occlusions. Tan et al. [335] demonstrates deep-learning–based pain detection from facial expressions in adult patients in a clinical setting. Zhang et al. [336] evaluates a deep learning facial pain recognition model for routine clinical assessment and decision support.

5.2 Proposed $\text{PSA}_{\text{video}}$ Systems

This section provides an in-depth examination of the implementation of the proposed $\text{PSA}_{\text{video}}$ System. Facial expressions play a key role in assessing pain levels, as individual responses to pain can vary greatly. Although there are numerous video-based pain detection algorithms, there is still room for improvement in pain recognition techniques. Mobile applications relying on the cloud that use streaming data and live video processing are gaining popularity. These applications typically

consist of two main components: the front end, which runs on the mobile device, and the back end, hosted in the cloud. The cloud enhances data processing capabilities and computational power, enabling our proposed method to run complex applications efficiently on devices with limited resources. In this chapter, we present a $\text{PSA}_{\text{video}}$ system, which is designed to estimate the intensity of pain based on facial expressions. The proposed system is presented in Fig. 5.2. The block diagram shows that an end-to-end system is proposed to predict pain intensity from patient videos. It begins with image/Video Data Preprocessing, followed by feature representation to extract relevant characteristics. The *PainCapsule* model is built using a combination of pre-trained models and an end-to-end deep learning model, where an ensemble of deep learning models is used to improve prediction accuracy by fusing features from multiple models. The final prediction model categorizes pain into different pain classes based on increasing intensities. This comprehensive system provides an automated and accurate solution for assessing pain levels, which aids in clinical diagnosis and treatment planning.

In this work, the separation between the end-to-end model and the deep learning models is based on the training strategy and usage, rather than the underlying learning paradigm. The proposed end-to-end model and ensemble models are trained from scratch using our dataset, with feature extraction and classification jointly optimized in a single learning process. In contrast, the pre-trained deep learning models rely on weights learned from large external datasets and are subsequently fine-tuned or used as feature extractors within our framework. Therefore, the distinction is made to emphasize the difference between end-to-end training from scratch and transfer learning–based approaches, not to imply that pre-trained models are outside the scope of deep learning.

5.2.1 Video preprocessing

This section focuses primarily on capturing facial expressions from video data (actually from a sequence of frames). However, noise introduced during image acquisition can degrade important textural details crucial to recognizing pain. This noise may lead to inaccurate assessments, potentially compromising the experimental results. To efficiently extract facial expressions, employs a tree-structured part model [296].

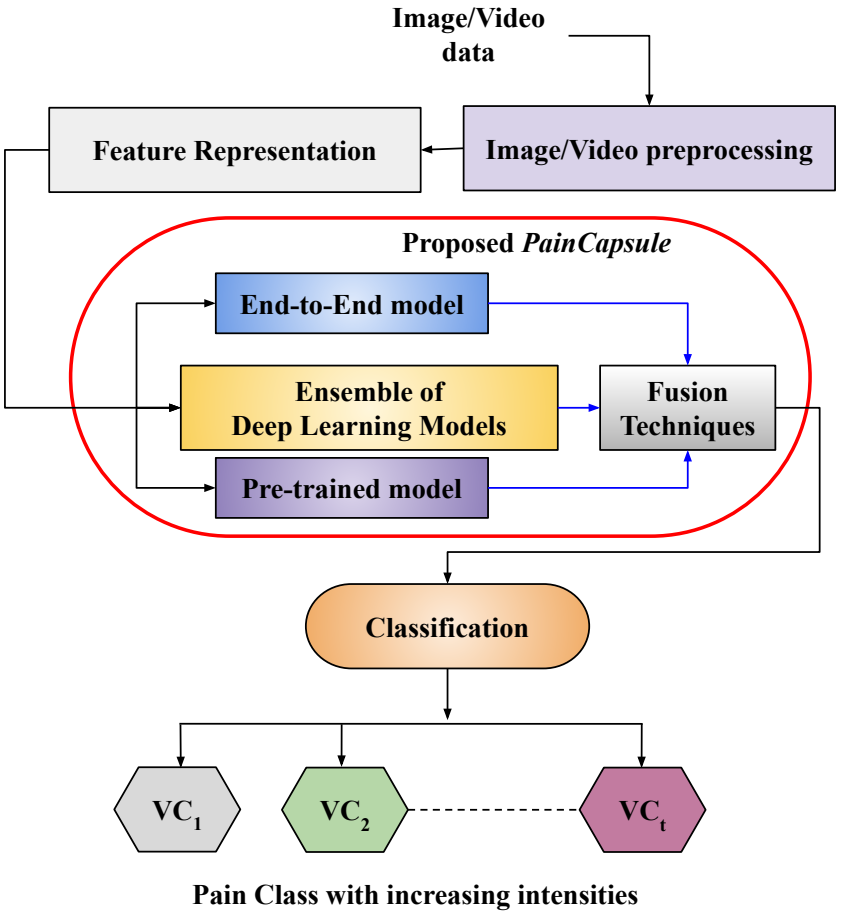


FIGURE 5.2: Block Diagram of the proposed PSA_{video} System using *PainCapsule* method.

To better understand the impact of frame preprocessing, such as facial expression detection, a study has been conducted using video frames from two different subjects, which compares the global motion changes and affects the facial region in the frames before and after applying facial region detection. The preprocessing performed here is similar to the previous Chapter 4, the comparison of frame differences of $subject_1$ curves before and after face detection reveals significant insights for video analysis research. The whole frame analysis in Fig. 5.3 demonstrates relatively low and uniform change scores, suggesting it primarily captures global motion and compression artifacts with limited sensitivity to facial movements. In contrast, the facial region analysis in Fig. 5.4 shows dramatically higher change scores with distinct. These individual patterns for each video indicate the successful isolation of meaningful facial dynamics. This sharp contrast validates that the isolation of the

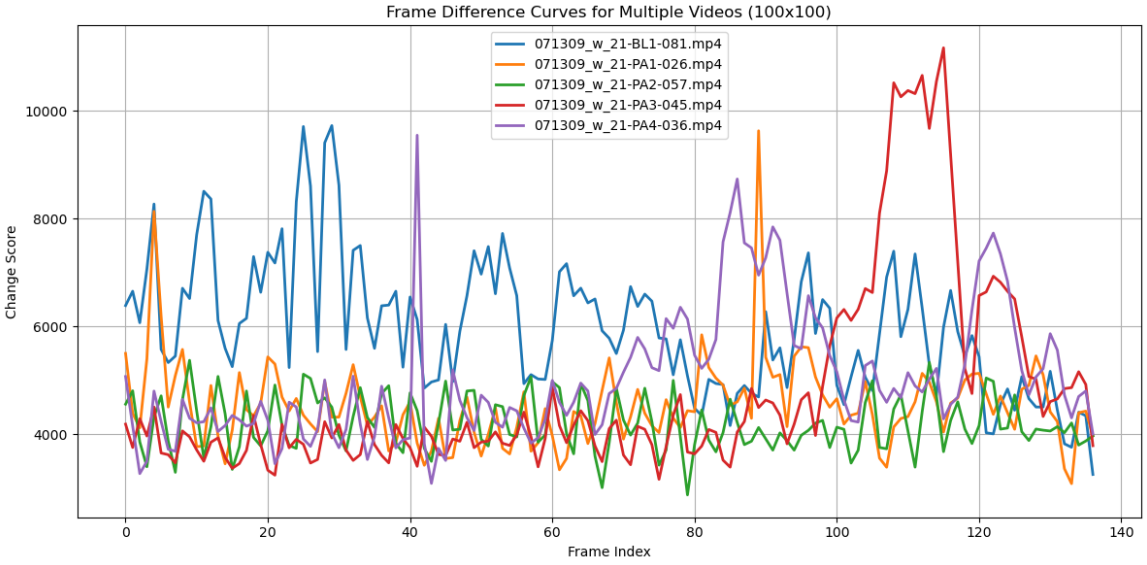


FIGURE 5.3: Detailed analysis of face global motion before facial expression detection for $subject_1$ of $\text{VD}_{\text{BioVid}}$.

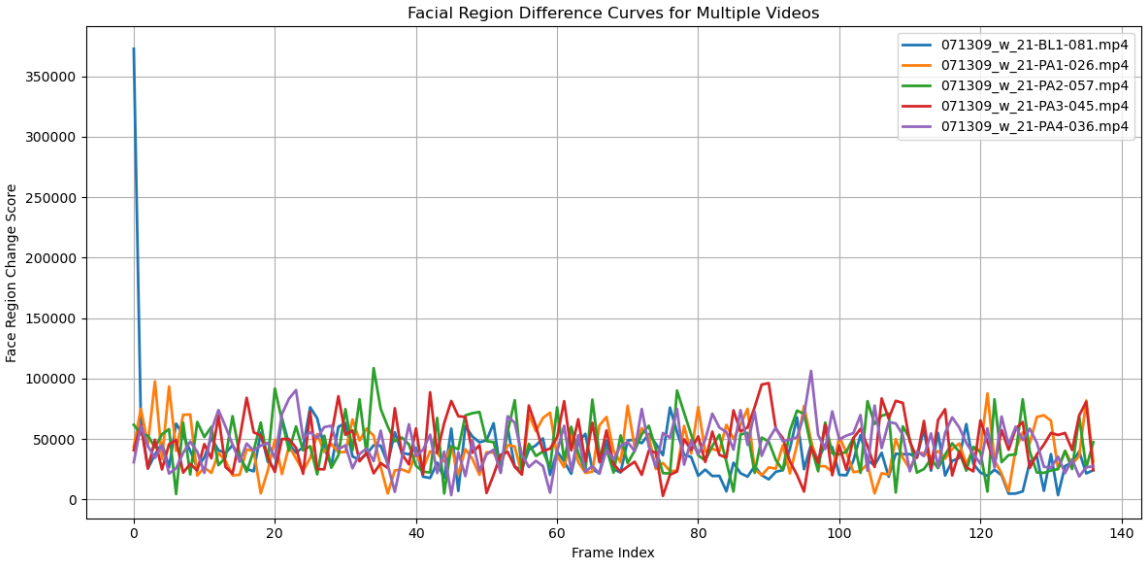


FIGURE 5.4: Detailed analysis of face global motion after facial expression detection for $subject_1$ of $\text{VD}_{\text{BioVid}}$.

facial region significantly enhances movement detection sensitivity by eliminating background noise and amplifying relevant facial signals. The differentiated curves in the facial analysis, showing periodic patterns in some videos and sporadic spikes in others, suggest that this method could enable precise characterization of individual facial behavior patterns, micro-expressions, or biometric signatures. These findings strongly support the incorporation of face detection as a preprocessing step in facial

movement research, as it transforms generic change detection into a powerful tool for analyzing subtle subject-specific facial dynamics that are otherwise obscured in whole-frame analysis.

The change between frames is calculated with the help of the Sum of Absolute Difference (SAD) by comparing each frame with the next one to measure how much the frames are changing over time. First, both frames are usually converted to grayscale and optionally resized (e.g., to 100×100 pixels) to simplify computation. For each pair of consecutive frames (Fi) and (Fi+1), the pixel-wise difference is computed using the formula $D(x, y) = |F_i(x, y) - F_{i+1}(x, y)|$, where (x, y) represents each pixel location. All these absolute differences are then summed to produce a single change score for that transition, given by $ChangeScore_i = \sum_{x,y} D(x, y)$. A low score means the two frames are similar and contain little movement, while a high score indicates significant motion or abrupt change. Plotting these scores over time provides a clear view of where and how strongly the frame variations occur within a video.

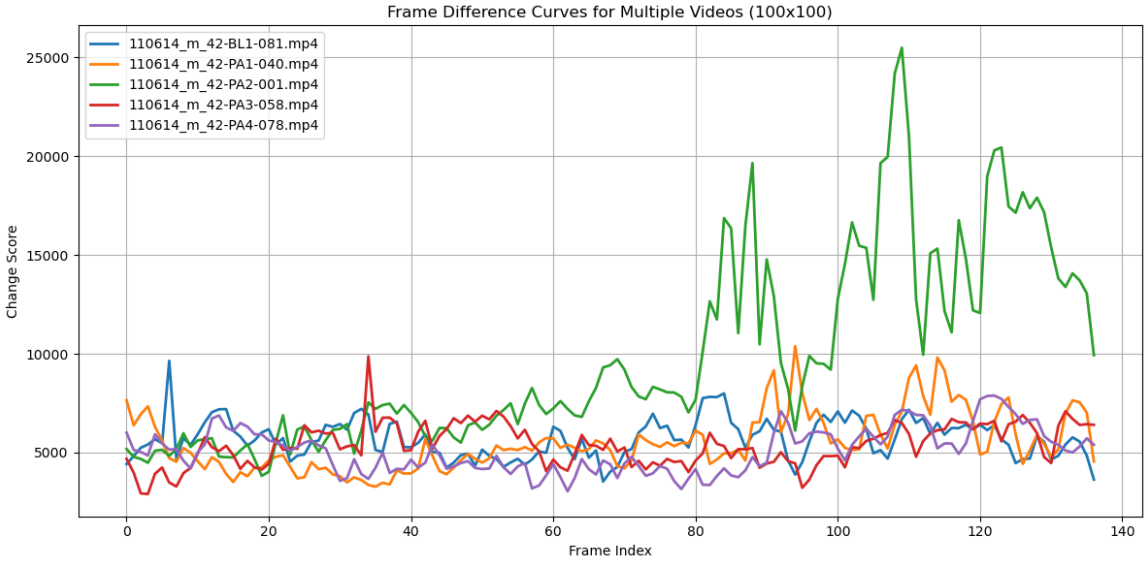


FIGURE 5.5: Detailed analysis of face global motion before facial expression detection for $subject_2$ of $\text{VD}_{\text{BioVid}}$.

The comparative analysis of frame difference curves before and after face detection $Subject_2$ in these videos reveals critical methodological insights for facial movement research. The whole frame analysis in Fig. 5.5 shows moderate change scores with relatively uniform fluctuations across all videos, suggesting that it captures

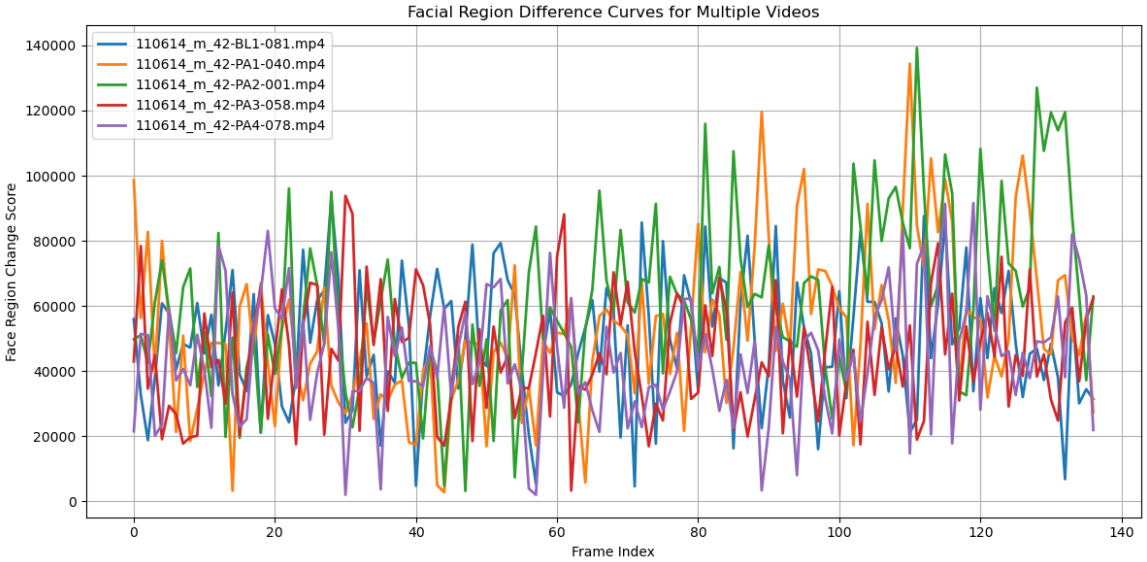


FIGURE 5.6: Detailed analysis of face global motion after facial expression detection for $subject_2$ of $\text{VD}_{\text{BioVid}}$.

general motion but lacks specificity for facial dynamics. In contrast, the facial region analysis in Fig. 5.6 shows significantly amplified change scores with distinct patterns for each video, demonstrating that face detection effectively isolates and magnifies facial movement signals while suppressing irrelevant background noise. The enhanced sensitivity reveals previously obscured individual differences in facial behavior, particularly notable in PI_3 and PI_4, which show pronounced spikes, suggesting that this approach can better capture micro-expressions or subject-specific movement patterns. These results reinforce that facial region isolation is essential for the precise analysis of facial dynamics. Fig. 5.7 shows the facial region detection frame-wise in the video sequence, which shows that the consistent detection of facial region would enhance this preprocessing step for facial behavior research.

5.2.2 $\text{DLPSA}_{\text{video}}$ System

This chapter utilizes two light CNNs (CNN_3 and CNN_4 of Chapter 4) that act as end-to-end deep learning models. Additionally, to get the benefits of real-time pattern identifications, some pretrained CNN models have also been employed, which are trained by the real-time ImageNet object classes. The combined framework of end-to-end and pretrained CNN architectures identifies the macro and micro-facial

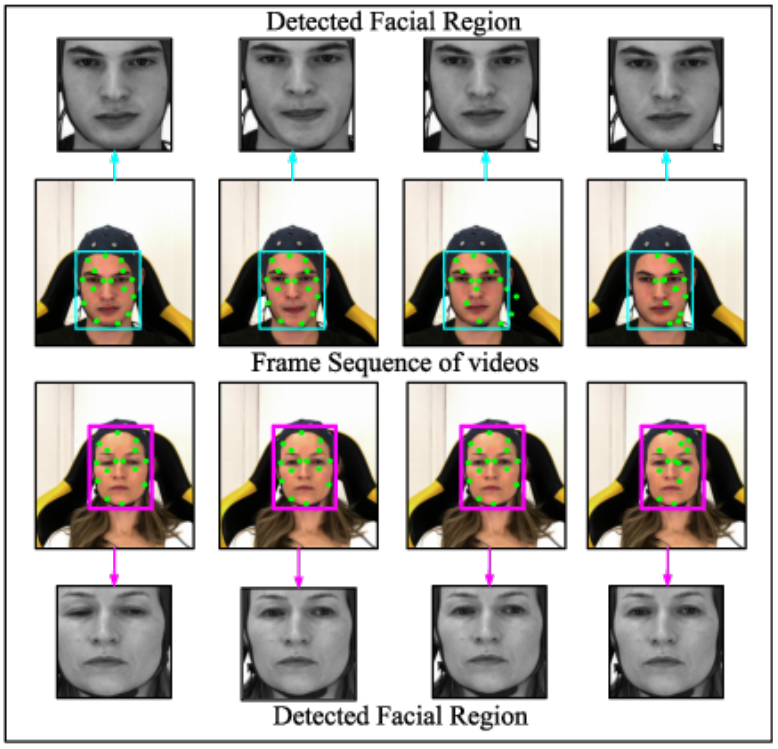


FIGURE 5.7: Results of frame preprocessing of PSA_{video} System.

expression patterns for the proposed pain detection system. The basic operations inside these CNN architectures are composed of the Convolutional layer (CL), Max-pooling layer (MPL), BatchNormalization (BN), Activation layer (AL), and Dense layer (DeL). Once convolution is completed, the network performs pooling operations, sharing weights to obtain the learning parameters over the network [337]. Several techniques are integrated to improve the performance of CNN model and address the challenges of overfitting, class imbalance, regularization, transfer learning, and fine-tuning. In addition, batch normalization, regularization, label smoothing, and various optimization tools are used to improve the robustness of the proposed CNN architecture. The proposed method emphasizes efficient extraction and utilization of facial features for optimal classification of pain detection while ensuring network stability and resilience through regularization and normalization techniques. Hence, the proposed system uses two end-to-end CNN architectures and five pretrained CNN models for feature extraction. Once each CNN model extracts the features fused at the feature level, they are processed for further processing. Here, in the successful execution of (CNN_3 and CNN_4 of Chapter 4), the proposed Capsule architecture has been proposed. Together with CNN_3 and CNN_4, the architecture of

the proposed capsule model includes five different pretrained CNN models, such as VGG16, InceptionV3, MobileNetV2, DenseNet, and EfficientNet. The following are the pretrained models, which are discussed as follows:

- **VGG16:** This architecture is based on the CNN framework proposed at the University of Oxford by the Visual Geometry Group [15]. The architecture comprises 16 layers organized systematically, including 13 convolutional layers followed by a pooling layer and three fully connected layers. Each convolutional layer has a kernel with dimension 3×3. It is trained with the Imagenet [338] dataset, which contains more than a million images. This model is considered here a pre-trained model, which computes f_{vgg16} features from each input image.

- **InceptionV3:** This architecture was proposed by Szegedy et al. in 2015 [17]. It is mainly a kind of CNN framework. Some methodologies are used to improve the performance of the model, such as factorized convolution, label smoothing, batch normalization, and an auxiliary classifier. The model is trained with the Imagenet dataset and is also used as a pre-trained model, which computes the $f_{inceptionv3}$ features from each input image.

- **MobileNetV2:** In the year 2018 Sandler et al. proposed MobileNetV2 [339]. It is a highly efficient and lightweight convolutional neural network framework. The model introduces the concept of linear bottlenecks with inverted residuals, which decreases the computation cost and the number of parameters while maintaining accuracy. The model is remarkably efficient with limited computational resources for tasks such as object recognition and image classification. This model is used here as a pre-trained model, which computes $f_{mobileNetv2}$ features from each input image.

- **DenseNet:** In 2017, the model was introduced by Huang et al. [340]. Mainly, it is a kind of deep learning-based framework in which each layer is connected to every other layer in a feed-forward manner, ensuring that information flows sequentially from the input to the output layer. The use of a dense connectivity mechanism helps reduce problems with respect to vanishing gradient descent and reuse of features, resulting in the model gaining more perfection with some specific parameters. Robust performance of the model was obtained in the case

of different tasks, such as image classification, using various parameters and high-level computational efficiency. This model is used here as a pre-trained model, which computes $f_{denseNet}$ features from each input image.

- **EfficientNet:** In 2019, this model was introduced by Tan et al. [341]. It is based on the CNN framework, which balances the scaling of the width of the model, depth, and resolution to achieve enhanced performance with very few parameters and reduced computational cost. EfficientNet uses the concept of compound scaling to scale all dimensions within a network uniformly. This model is also used here as a pre-trained model, which computes $f_{efficientNet}$ features from each input image.

5.2.2.1 Proposed *PainCapsule* model

The above-discussed two end-to-end CNN architectures and the pre-trained CNN architectures are used to build a unified framework as the deep learning-based feature representation technique to form a *PainCapsule* (Fig. 5.8). In this *PainCapsul*, the trained features from CNN_3, CNN_4, and those mentioned above, five pre-trained models such as f_{vgg16}, $f_{inceptionv3}$. $f_{mobileNetv2}$, $f_{denseNet}$, and $f_{efficientNet}$, are fused. Figure 5.8 illustrates the block diagram of the proposed pain recognition system using *PainCapsule*. In the final part of *PainCapsule*, a 1-dimensional CNN architecture has been proposed. In this architecture, the convolution operation is performed using a $t \times t$ mask, followed by the application of the rectified linear unit (ReLU) activation to the feature maps. Then, max-pooling layers are employed to extract the important features from these rectified maps. To ensure feature consistency, the important features pass through a batch normalization layer. In the final stage, two fully connected layers are added to flatten the features, preparing them for further processing and classification. Hence, the final classification is obtained on the testing data using three fully connected layers and a final output Dense layer. The fully connected layer consists of a Dense BatchNormalization, Activation, and Dropout layer. The details of the parameters of *PainCapsule* are given in Table 5.1.

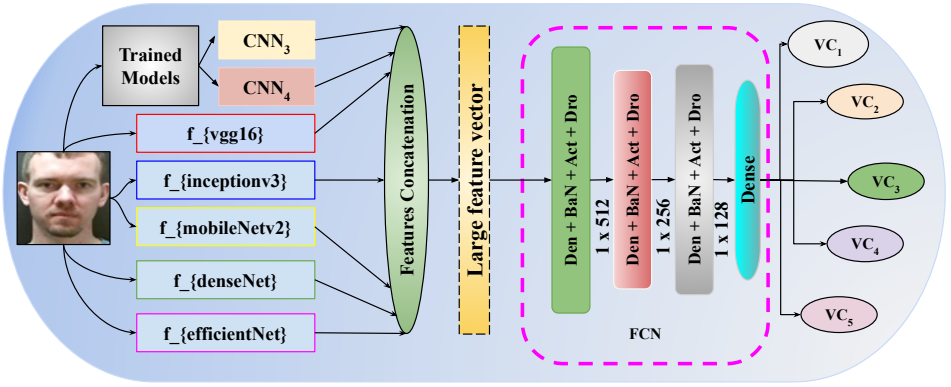


FIGURE 5.8: Proposed *PainCapsule* architecture for $\text{DLPSA}_{\text{video}}$ System.

TABLE 5.1: List of parameters required of the Proposed *PainCapsule* framework.

Layers	Outputshape	Image Size	Parameters
Dense	(1,512)	(1,512)	(1 + 17640) × 512=9032192
BatchNorm	(1,512)	(1,512)	4 × 512=2048
Activation Re LU-	(1,512)	(1,512)	0
Dropout	(1,512)	(1,512)	0
Dense	(1,256)	(1,256)	(1 + 512) × 256=131328
BatchNorm	(1,256)	(1,256)	4 × 256=1024
Activation Re LU-	(1,256)	(1,256)	0
Dropout	(1,256)	(1,256)	0
Dense	(1,128)	(1,128)	(1 + 256) × 128=32896
BatchNorm	(1,128)	(1,128)	4 × 128=512
Activation Re LU-	(1,128)	(1,128)	0
Dropout	(1,128)	(1,128)	0
Dense	(1,5)	(1,5)	(256+1) × 5=1285
Total Parameters for The Input Image Size:			9200645
Total Number of Trainable Parameters:			9198853
Non-trainable params:			1792

5.2.2.2 Proposed *PainAttentionCapsule* model

The attention layer has become a crucial component in deep learning architectures, particularly in CNNs. Given that CNNs consist of multiple blocks, each with several filter banks, they are capable of extracting more discriminative features in the derived feature maps. Introducing an attention layer enhances these features by focusing on the most discriminative ones. Similarly to how the human brain prioritizes the most important parts first, the attention layer directs attention to the most

crucial features. The Attention Model (AM), introduced by Bahdanau et al. [101], has since evolved into a key concept in neural network research. Over time, attention mechanisms have gained significant popularity in various domains, including Natural Language Processing (NLP) [342], speech recognition [343], and numerous computer vision applications [344]. This illustrates the exceptional versatility of attention mechanisms, reinforcing their importance and continued relevance in driving advancements in AI.

In our proposed *PainAttentionCapsule* model, the attention layer is introduced at the feature concatenation layer of *PainCapsule*, followed by a fully connected layer and an output layer. Mathematically, the attention layer is defined as: $\text{Attn}(A, B, C) = \text{softmax}\left(\frac{ABT}{\sqrt{d_b}}\right)$, where A, B, and C represent the query, key, and value matrices, respectively, and d_b is the dimension of the key vectors. The Softmax function normalizes the attention scores in this model. This equation computes attention scores by taking the dot product of the query matrix (A) and the key matrix (B), scaling the result by d_b, and then applying the Softmax function. The resulting attention scores are used to weight the value (C) matrices. This attention layer is integrated with the CNN architecture described above. The block diagram illustrating the CNN architecture with the attention layer is shown in Fig. 5.9.

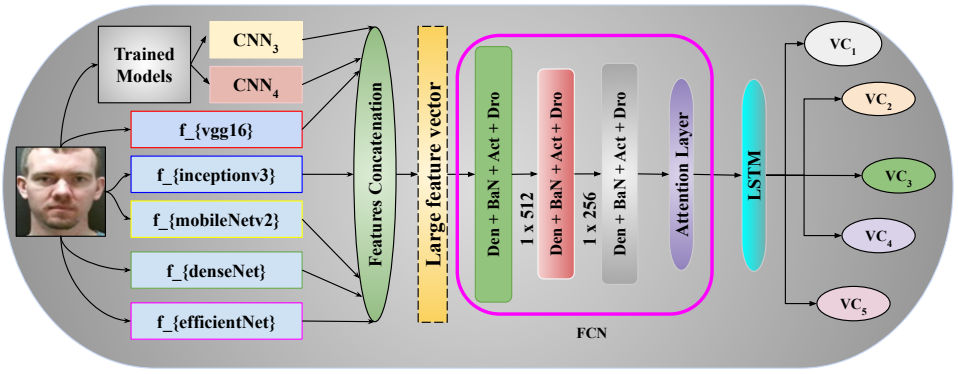


FIGURE 5.9: Proposed *PainAttentionCapsule* architecture for $\text{DLPSA}_{\text{video}}$ System.

The effectiveness of the proposed system is influenced by various components, including the CNN_3, CNN_4, and *PainCapsule* frameworks. Enhancing the performance and efficiency of these models is crucial, achieved through improved feature abstraction schemes for the proposed $\text{DLPSA}_{\text{video}}$ system. Here, each video V_i contains M frames. During pain motion video acquisition, pain expressions appear after a few

frames per second. It has been observed that the initial and final frames do not capture pain expressions effectively. Consequently, the primary pain expressions are captured between frames m_k and m_{M-K}, where $k \in \{0, 1, \cdots, M\}$ represents the frame index. For experimental purposes, features based on the *PainAttentionCapsule* are extracted from 14 frames between m_k and m_{M-K} at an interval of t. Each frame yields a feature vector of dimension $R^{1\times512}$, resulting in a feature matrix for each video V_i represented as $v \in R^{14\times512}$.

5.3 Experiments and Results

In this chapter, the experiments have been conducted using two video datasets, which are as follows:

(i) BioVid ($\text{VD}_{\text{BioVid}}$): In this chapter, the BioVid dataset $\text{VD}_{\text{BioVid}}$ is considered a video dataset. The description of $\text{VD}_{\text{BioVid}}$ is already discussed in Chapter 4. This dataset contains 5 different classes starting with PI_0 to PI_4 (discussed earlier in Section 4.3).

(ii) MIntPAIN ($\text{VD}_{\text{MIntPAIN}}$): The second dataset also contains the video data of 20 individuals. In the second dataset, for each individual, two trials were conducted to capture the data, and in those two trials, 40 sweeps of pain simulation were obtained. For each sweep, the data is captured; the first one is for No-pain or Pain Intensity-0 (PI_0), and the second trial is for capturing the data of Pain Intensity-1 (PI_1), Pain Intensity-2 (PI_2), Pain Intensity-3 (PI_3) and Pain Intensity-4 (PI_4) (discussed earlier). The statistics of this dataset are given in Table 5.2. The training and testing split for the database is $50-50\%$. The sample frames of the dataset are shown in Fig. 5.10.

TABLE 5.2: Dataset description of both $\text{VD}_{\text{BioVid}}$ and $\text{VD}_{\text{MIntPAIN}}$ for Approach-1, Approach-2 and Approach-3.

Approach	Number of Subject (Individuals)	Number of Videos	Number of frame from each Video	Number of class	Total Number of Images
Approach-1	5	20	14	5	7000
Approach-2	10	20	14	5	14000
Approach-3	20	20	14	5	28000

FIGURE 5.10: Sample video of $\text{VD}_{\text{MIntPAIN}}$ (5-Class).

By grouping PI_1 with PI_2 as *LowPain*, and PI_3 with PI_4 as *HighPain*, and retaining (PI_0) as it represents a modified version of both $\text{VD}_{\text{BioVid}}$ and $\text{VD}_{\text{MIntPAIN}}$ a combined dataset has been prepared. Fig. 5.11 and Fig. 5.12 illustrate sample video frames from the 3-class versions of the $\text{VD}_{\text{BioVid}}$ and $\text{VD}_{\text{MIntPAIN}}$, respectively.

This work incorporates five pre-trained CNN architectures: VGG16, InceptionV3, MobileNetV2, DenseNet, and EfficientNet. These architectures are used in their pre-trained form, meaning they have already been trained on the extensive ImageNet dataset, which contains over 14 million images across more than 20,000 categories. Through this prior training, they have learned to recognize a wide range of patterns, textures, shapes, and object features. As pretrained models, they come with optimized weights and parameters suitable for general visual recognition tasks, allowing us to build on their learned features instead of starting from randomly initialized weights. Again, the pre-trained CNN models used as feature extractors can often

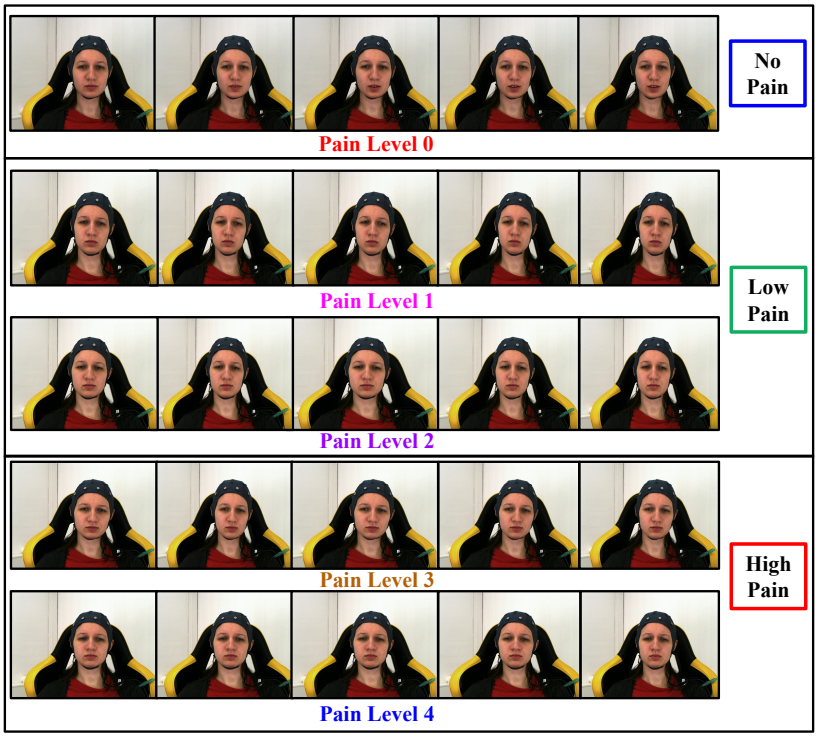


FIGURE 5.11: Sample video frames from VD_{BioVid} (3-Class).

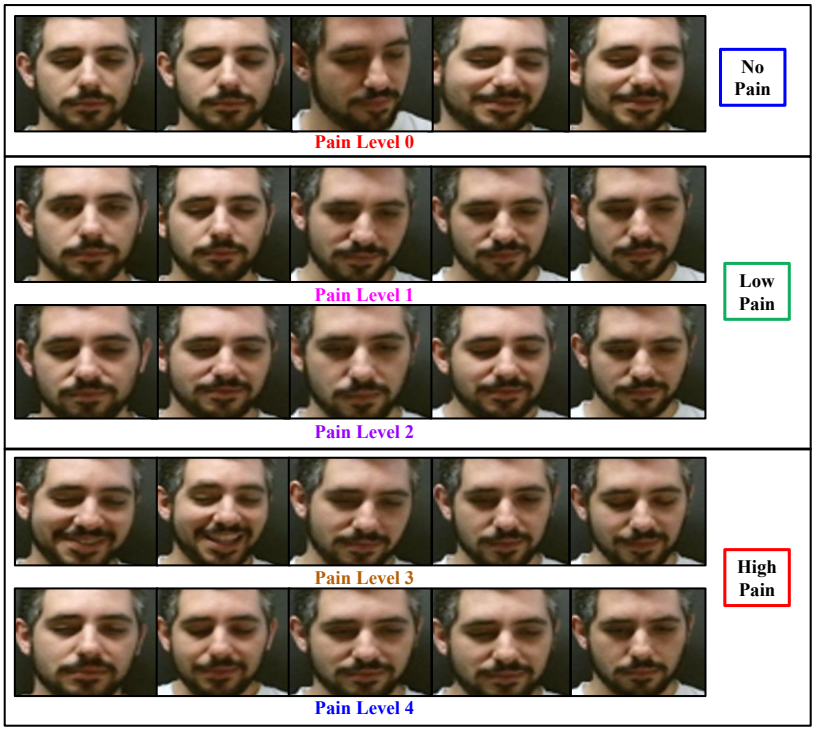


FIGURE 5.12: Sample video frames from $VD_{MIntPAIN}$ (3-Class).

yield better feature representations for each image than training a CNN model from scratch in an end-to-end fashion. Having been trained on large, diverse datasets like ImageNet, these models have already learned to capture a rich array of image characteristics, including edges, textures, and shapes. This results in a robust and well-generalized feature set that can be effectively applied to various tasks, often achieving high performance with minimal additional training.

In contrast, end-to-end training from scratch typically requires substantial data and computational resources and may fall short of the feature depth and quality provided by pre-trained models, especially on smaller or more specialized datasets. Hence, based on these facts, these pre-trained models are considered here as a feature extractor that extracts features in the form of a feature vector corresponding to each image or frame of the video sequence. The feature vectors are obtained from the pre-trained models as follows: $f_{vgg16} \in R^{1\times4096}$ from VGG16, $f_{inceptionV3} \in R^{1\times2048}$ from InceptionV3, $f_{mobileNetV2} \in R^{1\times1024}$ from MobileNetV2, $f_{denseNet} \in R^{1\times1024}$ from DenseNet, and $f_{efficientNet} \in R^{1\times1024}$ from EfficientNet. To ensure the discriminative power of the features extracted from these trained models, individual experiments are conducted using the features of each model with a neural network classifier. The performance of these models is illustrated in Fig. 5.13. From this figure, it has been observed that the performance due to the used pre-trained CNN models lies in the $32-43\%$ range of accuracies. Among these, it is also seen that $f_{modelNetv2}$ achieves better performance than the other pre-trained models.

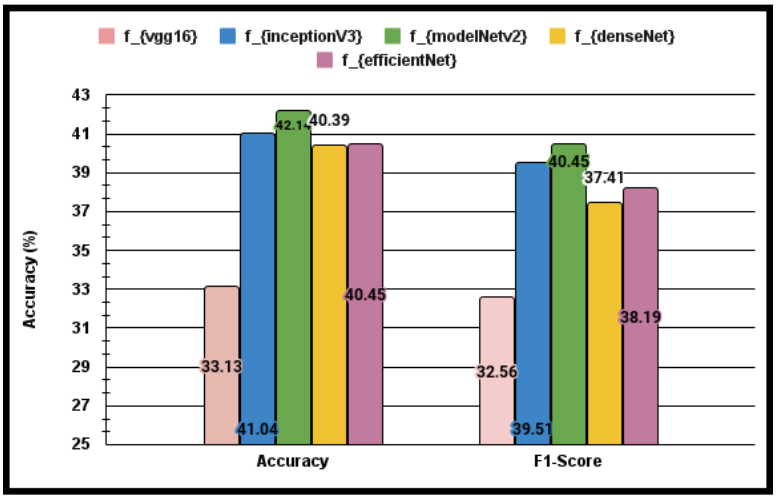


FIGURE 5.13: Performance for $\text{DLPSA}_{\text{video}}$ system using pretrained models for $\text{VD}_{\text{BioVid}}$ (Approach-1).

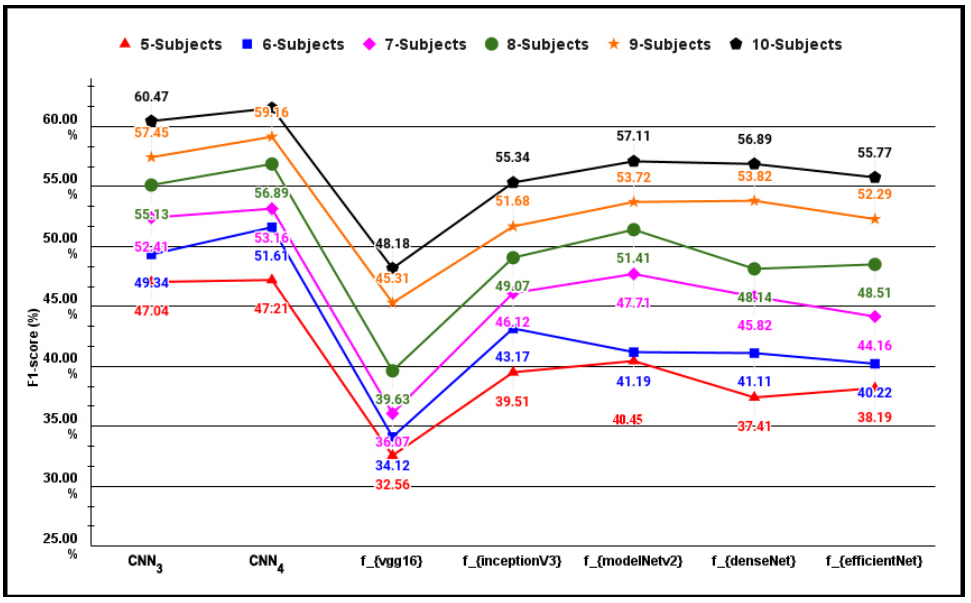


FIGURE 5.14: Performance for $\text{DLPSA}_{\text{video}}$ System using end to end CNN models for $\text{VD}_{\text{BioVid}}$ (Approach-2).

This experiment will justify the performance of the proposed *PainCapsule* model, which has been implemented by employing both end-to-end and pretrained CNN models. Here, based on the findings from end-to-end and pretrained CNN models, the performance of the proposed *PainCapsule* framework is evaluated. In this setup, two end-to-end trained architectures, CNN_3, and CNN_4, are utilized alongside five pre-trained CNN architectures: $f_{vgg16} \in R^{1\times4096}$ from VGG16, $f_{inceptionV3} \in R^{1\times2048}$ from InceptionV3, $f_{mobileNetV2} \in R^{1\times1024}$ from MobileNetV2, $f_{denseNet} \in R^{1\times1024}$ from DenseNet, and $f_{efficientNet} \in R^{1\times1024}$ from EfficientNet. These seven models are used as feature extractors for each image or video frame. From the CNN_3 and CNN_4 models, the feature vectors $f_{CNN_3} \in R^{1\times512}$ and $f_{CNN_4} \in R^{1\times512}$ are extracted from each facial image. Then, an FCN architecture is employed after the feature-level fusion to address the 3- or 5-class pain detection problem using the frameworks illustrated in Table 5.1. This integrated framework, which combines multiple feature vectors from the end-to-end and pretrained CNN models, offers a greater potential to capture more discriminative features and accurately classify them into predefined categories. This comprehensive approach is referred to as the Capsule framework and, specifically, as *PainCapsule* when applied to pain-level detection. Fig. 5.14, and Fig. 5.15 show the individual performance of end-to-end and pre-trained CNN models using facial expressions from 5-10 subjects (*Approach 1* to *Approach 2*) pain

detection problem using both VD_{BioVid} and $VD_{MIntPAIN}$ respectively.

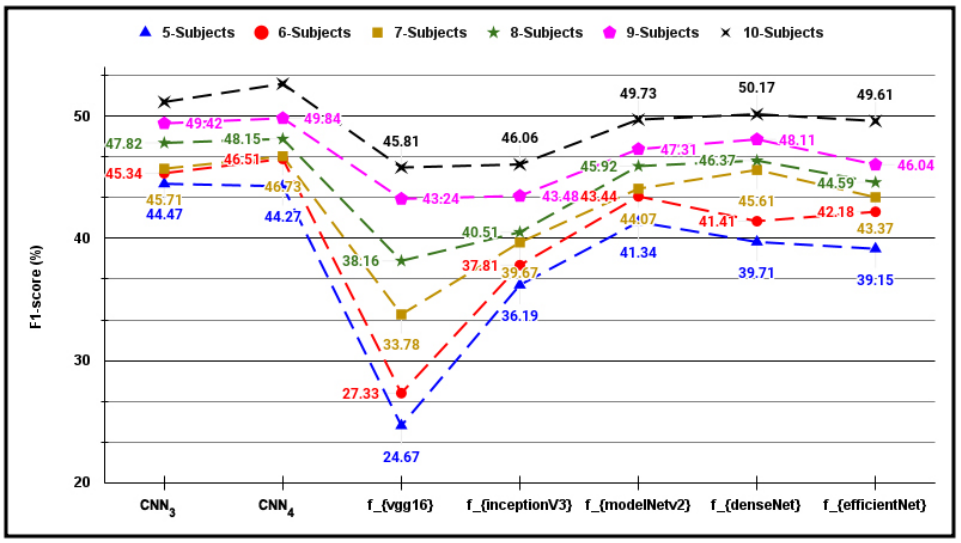


FIGURE 5.15: Performance for $DLPSA_{video}$ System using end to end CNN models for $VD_{MIntPAIN}$ (Approach-2).

From these figures, it has been observed that the performance of the end-to-end trained models outperforms the pretrained ImageNet CNN models. This improvement is primarily due to the superior ability of the end-to-end models to extract and discriminate texels in facial images accurately. These models are specifically trained using multi-resolution and manually adjusted CNN blocks, such as Convolution, Max-pooling, batchNormalization, and Dropout layers, allowing them to capture finer details in the facial expressions. In contrast, the pre-trained ImageNet models extract features based on generalized patterns like shapes, blobs, and edges, which are more suitable for object detection rather than the subtle features needed for facial expression analysis. To utilize the strengths of both approaches, features extracted from the end-to-end models and the ImageNet-trained CNN architectures are combined using a feature fusion technique. The performance of the *PainCapsule* framework using the VD_{BioVid} is summarized in Table 5.3, showing results for different approaches: Approach 1 (considering image samples from 5 subjects), Approach 2 (10 subjects), and Approach 3 (20 subjects). Similarly, the performance of the *PainCapsule* framework on the $VD_{MIntPAIN}$ is summarized in Table 5.4.

The experiment using the *PainAttentionCapsule* model is organized into three distinct stages. The first stage focuses on the extraction of features using a CNN. A large collection of frames from training videos is fed into CNN, where the network

TABLE 5.3: Performance of the $DLPSA_{video}$ system using VD_{BioVid} with different approaches.

Approach	Models	Accuracy	F1-Score	Precision	Recall
Approach 1	End-to-end	48.34	47.13	47.39	47.67
	Trained-network	39.43	37.62	37.47	38.56
	PainCapsule	51.56	49.85	49.65	50.71
Approach 2	End-to-end	61.47	59.72	60.22	60.35
	Trained-network	58.11	56.51	57.34	57.16
	PainCapsule	63.61	62.29	62.17	62.47
Approach 3	End-to-end	64.55	63.67	64.36	63.83
	Trained-network	60.51	59.62	59.28	59.74
	PainCapsule	65.75	64.38	64.46	64.62

TABLE 5.4: Performance of the $DLPSA_{video}$ system using $VD_{MIntPAIN}$ with different approaches.

Approach	Models	Accuracy	F1-Score	Precision	Recall
Approach-1	End-to-end	46.81	45.53	45.23	47.64
	Trained-network	43.56	41.13	42.19	43.17
	PainCapsule	46.12	45.72	45.87	46.46
Approach-2	End-to-end	50.45	48.71	48.76	49.38
	Trained-network	45.71	43.89	43.28	44.53
	PainCapsule	51.89	49.35	48.74	50.26
Approach-3	End-to-end	53.35	51.89	52.62	52.21
	Trained-network	47.89	45.72	46.33	46.72
	PainCapsule	53.71	52.06	52.41	52.94

is trained to learn spatial patterns by adjusting the weights within its convolutional filters. The *PainAttentionCapsule* model (Fig. 5.9) is used in this stage, acting as a feature extractor to identify relevant features that indicate varying levels of pain intensity from each video frame. In the second stage, the extracted features of each frame are passed into an LSTM network. The LSTM is designed to capture temporal dependencies across the sequential frames, enabling the model to understand how pain expressions evolve. By processing the sequential data, the LSTM ensures that the dynamic aspects of pain intensity are incorporated into the feature representation. The final stage is dedicated to classifying the test videos into one of five pain intensity levels. Using the aggregated features from the LSTM, the system predicts the appropriate pain class for each test video. This three-stage approach, combining spatial feature extraction, temporal modeling, and classification, ensures

a comprehensive understanding of pain intensity, enhancing the accuracy and robustness of the system in detecting pain levels. The feature matrices for each video

TABLE 5.5: The LSTM architecture summarization and its parameters for the $\text{DLPSA}_{\text{video}}$ System.

Layer	Output Shape	Parameters
LSTM	(None, 256)	787,456
Dropout	(None, 256)	0
Dense	(None, 128)	32,896
Dense	(None, 5)	645
Total parameters:		820,997
Trainable parameters:		820,997
Non-trainable parameters:		0

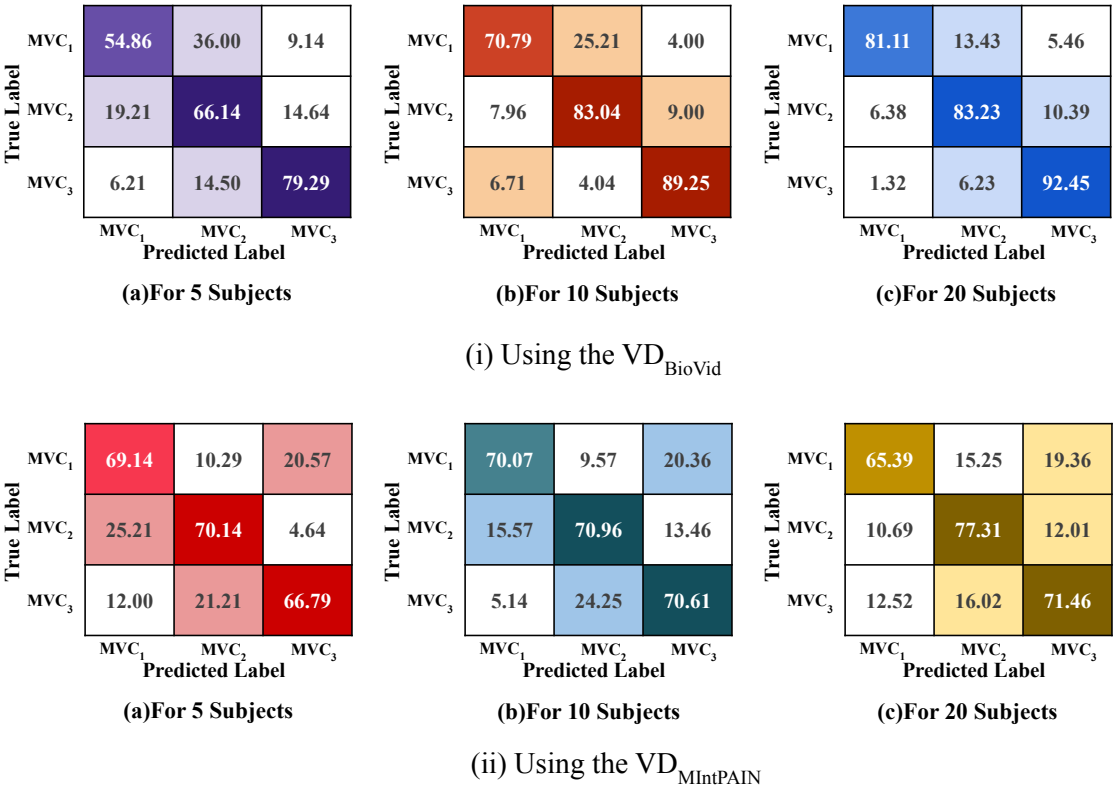


FIGURE 5.16: Demonstration of confusion matrix in percentage as the performance measure for the $\text{DLPSA}_{\text{video}}$ system for 3-Class problem.

are processed through an LSTM architecture to classify pain intensity into five levels, allowing the detection of pain intensity for each patient from V_i. A 50-50% training-testing split is applied in each approach. For Approach-1, 250 videos are used for

TABLE 5.6: Performance of the LSTM model using VD_{BioVid} and $VD_{MIntPAIN}$ for 3-class problem.

3-class Problem				
No. of Subject	**Accuracy**	**F1-Score**	**Precision**	**Recall**
VD_{BioVid}				
5	69.14	66.6	66.48	66.76
10	83.07	81.01	81.07	81.03
20	86.49	85.75	86.05	85.6
$VD_{MIntPAIN}$				
5	68.61	67.21	67.53	68.69
10	70.64	69.93	69.57	70.55
20	72.53	70.84	70.61	71.38

training and 250 for testing. In Approach-2, 500 videos are used for training and 500 for testing. Similarly, in Approach-3, 1,000 videos are used for training and 1,000 for testing. The LSTM architecture and its parameters are illustrated in Table 5.5. The experiment with this LSTM architecture is performed in two ways. In the first experiment, a 3-class system has been introduced, which combines $VC_2 + VC_3$ as Low Pain (MVC_2) and $VC_4 + VC_5$ as High Pain (MVC_3), while VC_1 remains the same and is renamed as MVC_1. For VD_{BioVid}, accuracy improves from 69.14% (5 subjects) to 86.49% (20 subjects), with consistent gains in precision, recall, and F1-score, demonstrating better performance with more subjects. The $VD_{MIntPAIN}$ shows slightly lower but stable performance (68.61% to 72.53% accuracy), with low Pain consistently being the best-classified category (F1-score: 66.62–85.75% in VD_{BioVid}, 67.21–70.84% in $VD_{MIntPAIN}$). Fig. 5.16 and Table 5.6 reveal that high pain classification improves with larger datasets, while MVC_1 remains the most challenging to distinguish, particularly from high pain to low pain. Both databases exhibit similar trends, with VD_{BioVid} generally outperforming $VD_{MIntPAIN}$, likely due to differences in data collection or labeling protocols. From the consistent structure, we are able to directly compare between databases and subject groups, highlighting the robustness of the 3-class simplification for pain assessment, while the second experiment is performed considering actual 5-Classes, the confusion matrix in Fig. 5.17 and Table 5.7 describes that the performance with VD_{BioVid} in terms of accuracy is increased from 54.31% to 65.64% and for $VD_{MIntPAIN}$ is from 56.46% to 58.81% with the increased number of subjects from 5 to 20. From these confusion matrices, it has been observed that, like the $DLPSA_{image}$ System, the $DLPSA_{video}$ System has also achieved

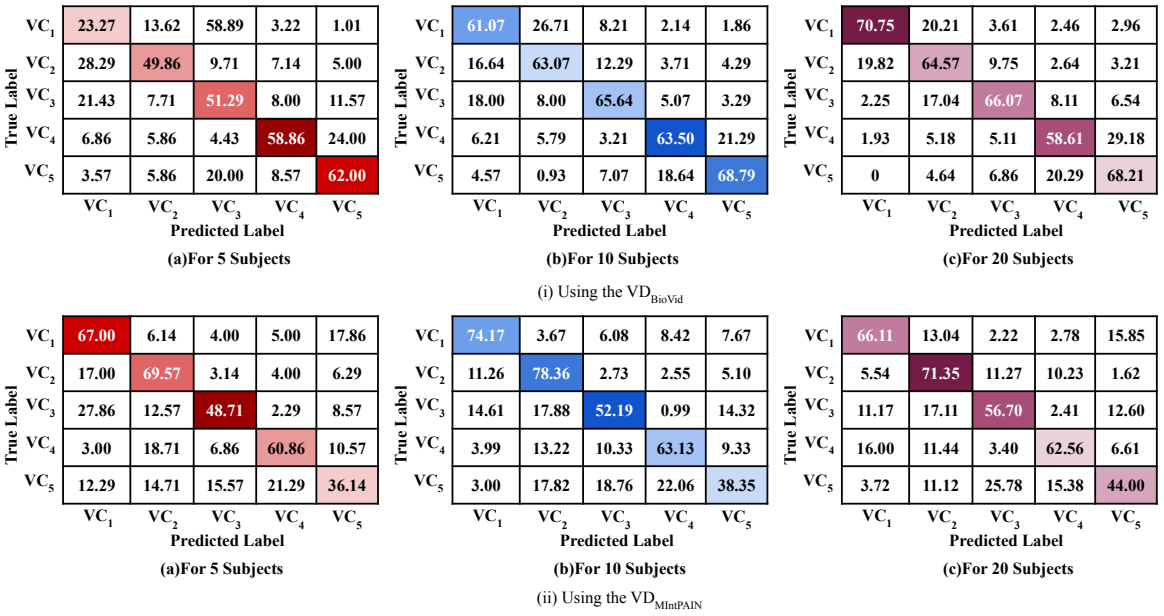


FIGURE 5.17: Demonstration of confusion matrix in percentage as the performance measure for the DLPSA$_{video}$ system for 5-Class problem.

better performance, and it is due to the proposed *PainAttentionCapsule* model.

TABLE 5.7: Performance of the LSTM model using VD$_{BioVid}$ and VD$_{MIntPAIN}$ for 5-class problem.

5-class problem				
No. of Subject	**Accuracy**	**F1-Score**	**Precision**	**Recall**
VD$_{BioVid}$				
5	54.31	53.87	55.49	54.32
10	64.41	63.92	63.69	64.43
20	67.83	67.29	67.08	67.54
VD$_{MIntPAIN}$				
5	56.46	55.95	56.55	56.17
10	57.47	56.92	58.81	57.29
20	58.81	57.85	59.49	58.52

The performance of the proposed system is compared with various state-of-the-art techniques for 5-Class VD$_{BioVid}$ and VD$_{MIntPAIN}$. Some of these existing methods, like Xiang et al. [305], Dimitra et al. [306], and Werner et al. [65] methods deal with hand-crafted and some of them deal with deep learning based feature representation approaches. These methods enlisted here are considered famous feature representation approaches for different applications based on computer vision like

scene understanding, object segmentation, object recognition, biometrics, etc. In this chapter, these SoA methods are executed from their description available in the papers with video samples of the 5-Class $\text{VD}_{\text{BioVid}}$ and $\text{VD}_{\text{MIntPAIN}}$ and testing is done by maintaining a similar training-testing protocol like the proposed system. Comparisons are conducted based on five different multiclass pain levels, such as VC_1, VC_2, VC_3, VC_4, and VC_5. The comparative performance of these methods is reported in Table 5.8 for the 5-Class $\text{VD}_{\text{BioVid}}$ and $\text{VD}_{\text{MIntPAIN}}$, and it is observed that the proposed system outperformed other competing methods. It compares how well these methods perform when looking at pain in pairs (like VC_1 vs. VC_2) and in a full multiclass setup. Older or more traditional methods, such as Yuille [305], Simos [306], Traue [65], Zhi [345], and Haque [346] generally perform well. However, the proposed models, especially *PainCapsule* and *PainAttentionCapsule* do a much better job. In fact, *PainAttentionCapsule* stands out as the best performer, reaching the highest accuracy: 67.83 on $\text{VD}_{\text{BioVid}}$ and 58.16 on $\text{VD}_{\text{MIntPAIN}}$, beating *PainCapsule*, which is second with scores of 65.75 and 53.23, respectively. These results highlight the effectiveness of the proposed *PainCapsule* and *PainAttentionCapsule* in improving pain classification performance.

TABLE 5.8: Performance comparison of the proposed $\text{DLPSA}_{\text{video}}$ system of this chapter.

$\text{VD}_{\text{BioVid}}$					
Method	VC_1 **vs.** VC_2	VC_1 **vs.** VC_3	VC_1 **vs.** VC_4	VC_1 **vs.** VC_5	**Multiclass**
Yuille [305]	26.81	29.83	37.42	42.17	23.39
Simos [306]	23.52	25.19	27. 13	35.51	21.14
Traue [65]	54.10	57.29	65.11	71.78	31.51
Zhi [345]	32.73	33.82	36.17	36.68	31.29
Haque [346]	33.28	34.47	36.21	38.86	32.05
PainCapsule	59.32	65.51	68.03	76.55	65.75
PainAttentionCapsule	61.15	67.33	70.14	78.13	67.83
$\text{VD}_{\text{MIntPAIN}}$					
Method	VC_1 **vs.** VC_2	VC_1 **vs.** VC_3	VC_1 **vs.** VC_4	VC_1 **vs.** VC_5	**Multiclass**
Yuille [305]	24.76	28.04	35.26	41.16	22.18
Simos [306]	21.79	23.56	26.37	34.21	20.27
Traue [65]	52.46	55.84	63.41	69.26	30.62
Zhi [345]	31.32	33.16	35.28	37.14	29.92
Haque [346]	32.54	33.87	35.71	38.29	30.77
PainCapsule	59.51	55.17	58.42	56.62	53.71
PainAttentionCapsule	52.09	56.14	59.17	57.17	58.81

5.4 Conclusions

This chapter implements the video-based PSA_{video} System. Here, facial expressions also play a key role in assessing pain levels. The implementation of this system begins with the preprocessing of video frames, where a frame is an input image that undergoes the image pre-processing task to extract the facial portion as a region of interest. The extracted facial region undergoes the $DLPSA_{video}$ System. This system utilizes two end-to-end CNN models (CNN_3 and CNN_4 of Chapter 4), and five different pre-trained CNN models, such as VGG16, InceptionV3, MobileNetV2, DenseNet, and EfficientNet. Then, these end-to-end CNN architectures and the pre-trained CNN architectures are used to build a unified framework as a deep learning-based feature representation technique to form a *PainCapsule*. Again, the concept of attention layer has been introduced with *PainCapsule* model to build the *PainAttentionCapsule* model. Both models are built and experimented under $DLPSA_{video}$ system and tested on VD_{BioVid} (BioVid), and $VD_{MIntPAIN}$ (Multimodal Intensity Pain (MIntPAIN)) datasets. Through experiments and results, it has been observed that the *PainAttentionCapsule* based $DLPSA_{video}$ system has achieved outstanding performance compared to all other competing methods. In this chapter, the *PainAttentionCapsule* based $DLPSA_{video}$ System is considered as the prediction model for the video-based pain sentiment analysis. This system will mitigate the limitations of PSA_{image} systems, providing valuable insights on variations in pain levels. This comprehensive system provides an automated and accurate solution for assessing pain levels, which aids in clinical diagnosis and treatment planning.

This chapter shows that both novel approaches outperform conventional techniques, establishing video-based analysis as the most effective solution for accurate pain sentiment assessment. Considering all the previous chapters and the current chapter, it is observed that pain sentiment analysis provides better and more authentic analysis about the pain of different patients. To further improve the performance of the pain detection models presented in this thesis, additional modalities may be incorporated. This integration aims to build a multimodal pain detection framework, which will be explored in detail in the next chapter.

Chapter 6

Multimodal-Based Pain Sentiment Analysis

In this chapter, Multimodal-Based Pain Sentiment Analysis (MPSA) Systems have been proposed with different combinations of multimodal data. MPSA systems play a vital role in assessing pain, which is both subjective and complex. These systems combine multiple data sources, such as facial expressions, vocal tones, and textual descriptions, to improve pain detection. The MPSA systems integrate data from various sources and offer a comprehensive approach to customize this data to detect pain sentiment. When the data comes from a single source, there exists correlation within different modalities of data. In the case of a real problem, when the data comes from a single subject, the post-classification fusion ensure more accurate evaluation. Although the multi-modal inputs originate from different sources, the post-classification fusion remains meaningful because the modalities are complementary and temporally aligned representations of the same underlying event or subject state. In real-world scenarios, heterogeneous sensors (e.g., text, audio, and visual-based sources) are commonly deployed in parallel, and each modality captures distinct aspects of the same phenomenon. Post-classification fusion allows each modality-specific model to independently learn discriminative patterns within its own feature space, while the fusion stage aggregates their class-level confidence

scores to produce a more robust and reliable final decision. This strategy is particularly suitable when the modalities have different sampling rates, noise characteristics, or missing data, as it avoids early-stage feature incompatibility and reduces error propagation. Therefore, even though the inputs are acquired from different sources, their fusion at the decision level reflects realistic multi-sensor deployments and improves generalization in practical applications. This holistic analysis allows the system to categorize the levels of pain intensity. In healthcare, an MPSA system follows a comparable approach, particularly valuable in e-healthcare systems, that includes many medical services, including hospitals, clinics, doctors, nurses, telemedicine, medical devices, and health insurance services [79]. These electronic healthcare systems utilize speech recognition, textual recognition, and video facial expression analysis to assess a patient's pain condition remotely by offering a more efficient and cost-effective way to assess and address patient needs. Gaur et al. [347] proposed a method that shows integration of multimodal data helps to reduce the dependency on self-reports, especially for patients who are unable to communicate, such as those with severe cognitive or physical impairments. By incorporating diverse data modalities, these MPSA systems improve reliability, as according to Werner et al. [348], relying on a single data type can be unreliable due to individual differences or situational factors. The challenging issues of these MPSA systems are (i) Data synchronization, and (ii) Modalities fusions. Here, in order to ensure accurate and coherent fusion, data gathered from many modalities, including text, audio, images, and video, are semantically aligned. This process is the data synchronization. This alignment improves the overall performance and reliability of the multimodal pain analysis by allowing the system to interpret pain-related cues consistently across various data sources.

Following data synchronization, fusion techniques are employed to integrate the outputs of individual unimodal systems, namely, text, audio, image, and video, into a unified multimodal pain detection system. In pre-classification fusion, features extracted from each modality are concatenated into a high-dimensional vector, which is then passed through a fully connected neural network for joint feature learning and classification. This approach requires building a new prediction model based on the combined features from all modalities. In contrast, post-classification fusion operates on the outputs of independently trained unimodal models and combines their predictions at the score or decision level without the need for retraining or

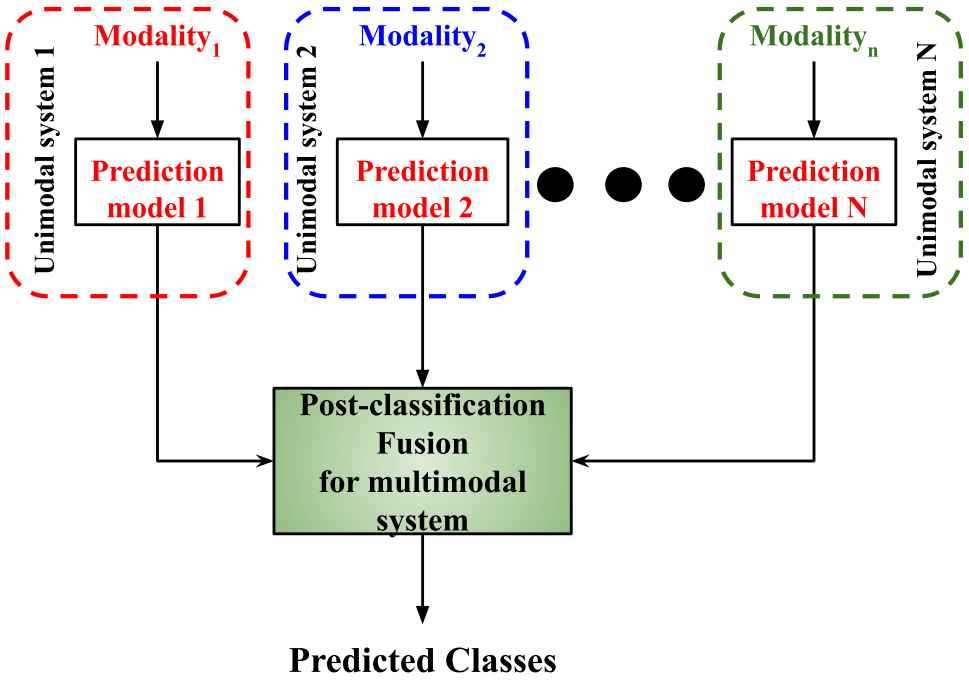


FIGURE 6.1: A block diagram of the proposed MPSA system.

redesigning a new model [349]. Empirical evidence suggests that post-classification fusion often achieves better performance in multimodal systems due to its ability to retain the strengths of specialized models and reduce feature-space complexity [350, 351]. Accordingly, this study adopts post-classification fusion techniques for the proposed MPSA system. Here, two post-classification fusion techniques have been implemented, (i) score-level fusion and (ii) decision-level fusion [352, 353]. A comprehensive discussion of the data synchronization and fusion methodologies is presented in this chapter, and the overall block diagram of the MPSA system is illustrated in Fig. 6.1.

The organization of this chapter is as follows. Section 6.1 discusses the existing literature on multimodal pain analysis. In Section 6.2, our MPSA systems have been discussed. The experimental results have been demonstrated in Section 6.3. Finally, Section 6.4 is the concluding section of this chapter.

6.1 Literature Review

Multimodal data integrates information from diverse sources such as text (e.g., patient self-reports), audio (e.g., vocal stress patterns), visual inputs (e.g., facial expressions), and physiological signals (e.g., heart rate). This integration offers a more comprehensive and reliable understanding of complex phenomena, such as pain, which is inherently subjective and emotion-laden [354]. In multimodal pain assessment, combining emotional cues from text, vocal features from speech, and facial expressions from video enables a more objective and accurate evaluation of pain levels. Such systems address the limitations of unimodal methods, such as self-report biases, and promote personalized healthcare, for example, by differentiating between acute physical pain and emotional distress or tailoring interventions based on richer patient profiles [355]. Werner et al. [356] have provided a comprehensive survey on automatic pain recognition systems, underlining the critical role of multimodal approaches in detecting nuanced emotional states. Behavioral indicators like grimacing, eye closure, and wincing are highly correlated with procedural pain in clinical settings [357], and facial expressions have been identified as strong markers of negative emotions relevant to pain evaluation [358]. In real-world applications, sentiment analysis tools are increasingly used to remotely monitor patients' facial expressions, providing clinicians with enhanced insight into their emotional and physical well-being [79]. Overall, pain assessment remains a cornerstone of clinical care, essential for both treatment planning and evaluation of therapeutic outcomes [116].

Traditionally, self-assessment tools such as verbal numerical rating scales or visual analogue scales have served as the gold standard for pain reporting [359]. However, several vulnerable patient populations—including those who are physically immobile, terminally ill [360], critically ill but communicative [66], or psychologically distressed [361]—often face significant challenges in consistently self-reporting their pain. Patients with cancers of the head, neck, or brain may especially require reliable technological support to facilitate pain detection and reporting, thereby easing the burden on healthcare professionals. In response, sentiment analysis systems have been adapted to support various medical and psychological conditions such as Parkinson's disease and depression, employing facial expression analysis to detect emotional states and intent. These systems process multimodal data—including

text, images, and video—using neural network-based models to identify affective patterns and classify emotional states [52]. However, traditional natural language processing (NLP) techniques frequently fall short in capturing the complex, multimodal nature of pain, which is rarely conveyed through text alone. Pain expression often involves a combination of vocal tone, facial micro-expressions, gestures, and physiological signals. Consequently, integrating textual, auditory, and visual inputs enables a more holistic and accurate analysis of pain-related sentiment. Among these modalities, visual indicators—particularly facial expressions and gestures—serve as some of the most direct and observable signs of pain [358]. Addressing these needs, we propose a multimodal sentiment-based pain analysis framework [32] that aims to detect, classify, and interpret emotional and behavioral manifestations of pain in a clinically meaningful way.

Contemporary healthcare systems increasingly rely on multimodal physiological signals for more accurate and comprehensive pain analysis [82]. Textual data contributes both as a channel for explicit pain reporting—via descriptive terms such as 'sharp' or 'throbbing'—and as an implicit behavioral marker through linguistic style variations, such as changes in pronoun usage or verb tense, which have been shown to correlate with pain intensity and chronicity. Audio signals, including vocal and respiratory biomarkers, offer real-time indicators of physiological stress responses. Features such as pitch variability, rhythm, jitter, shimmer, and speech pause durations provide measurable evidence of nociceptive processing through vocal cord and respiratory patterns [83]. Visual data in the form of static images captures discrete behavioral cues like facial expressions and facial action units (AUs); for instance, AU4 (brow lowering) and AU6/7 (orbital tightening) provide standardized evidence of pain-induced muscle contractions [84]. Video-based modalities enhance this further by capturing dynamic and spatiotemporal patterns of facial behavior, including micro-expression timing (onset, apex, offset) and coordinated facial muscle movements that static images fail to convey—especially useful in diagnosing pain-related movement disorders [19]. Importantly, pain expression varies significantly across demographic and clinical contexts, as observed by Williams et al. [362]. Liu et al. [363] demonstrated the broad applicability of multimodal pain assessment in areas such as postoperative monitoring, chronic pain management, and mental health evaluation. Fusion-based machine learning models, as discussed by Poria et al. [364], further improve classification accuracy by utilizing the complementary strengths of multiple

modalities. Moreover, Chen et al. [365] emphasized that the objectivity inherent in multimodal systems helps mitigate bias, enabling fairer and more reliable pain assessment across diverse patient populations.

Recent research has increasingly focused on utilizing deep learning techniques to integrate diverse pain-related modalities for more accurate and robust assessment. For example, Zhang et al. [254] proposed an intelligent system that employs attention mechanisms to selectively focus on salient features within facial video streams and physiological signals for pain intensity estimation. Similarly, Gruss et al. [366] developed a multimodal framework that effectively integrates various temporal data streams for pain recognition, significantly outperforming unimodal systems in consistency and reliability. Practical applications of these systems are emerging in clinical environments. Thiam et al. [367] introduced a multimodal post-operative pain monitoring tool that analyzes facial expressions and biosignals, yielding pain predictions closely aligned with clinical assessment. For chronic pain management, Lu et al. [368] explored a wearable sensor-based system combined with mobile reporting, enabling remote monitoring and intervention. Schmidt et al. [369] highlighted issues of bias and fairness in pain datasets, advocating for more inclusive and ethically designed AI systems. Li et al. [370] emphasized the necessity of implementing robust privacy-preserving mechanisms in AI-driven healthcare applications. Looking forward, future directions may include the development of lightweight, edge-compatible architectures [371] and explainable AI models [372] to improve transparency and clinician trust. Additionally, integrating multimodal pain sentiment analysis with electronic health records (EHRs) [373] holds promise for delivering highly personalized, real-time clinical care. Overall, multimodal pain analysis stands as a comprehensive and transformative approach that bridges technology.

6.2 MPSA Systems

The proposed MPSA systems [135] represent an emerging interdisciplinary domain that integrates affective computing, medical informatics, and multimodal machine learning to assess pain-related sentiment and emotional states [374]. These systems process heterogeneous data modalities, including text-based inputs (e.g., pain diaries

and social media posts), audio-based features (e.g., speech pitch and vocal tone), and video-based cues (e.g., facial Action Units from the Facial Action Coding System). Unlike traditional unimodal methods, multimodal approaches accommodate the multifaceted and subjective nature of pain by capturing complementary affective information in distinct channels [375]. Recent advances in deep learning, particularly attention-based fusion networks, enable the effective integration of temporal and semantic features from different modalities [376]. Despite their promise, these systems face notable challenges, such as modality alignment, handling missing data, and interpretability within clinical contexts [377]. Nonetheless, MPSA systems show strong potential in applications including chronic pain monitoring, post-surgical pain management, and mental health assessment [367]. Furthermore, ethical considerations related to patient privacy and algorithmic bias in pain assessment continue to warrant focused research [378]. The implementation of the proposed MPSA system requires data synchronization as well as post-classification fusion, which are discussed below.

6.2.1 Data Synchronization

In this chapter, the proposed Multimodal Pain Sentiment Analysis (MPSA) System is developed using sample datasets from three modalities: text, audio, and video. These datasets, sourced from varied environments and domains, represent pain-related content and are described in detail in their respective chapters. In practical scenarios, it is often unrealistic to assume the availability of all modalities for a given subject or condition. This creates inherent challenges in synchronizing such heterogeneous data within a unified framework. The complexity is further amplified by the modality-specific difficulties in data acquisition and recognition, such as linguistic ambiguity in text, variability in vocal signals, and occlusions or lighting inconsistencies in video, which have been comprehensively discussed in Chapter 1. Acknowledging these challenges, this chapter introduces the development of an MPSA system that addresses the multimodal synchronization problem.

Data synchronization is a crucial step that ensures different types of data, such as text, audio, and video, are aligned in time, allowing them to be analyzed together [45,

377]. Fig. 6.2 illustrates the data synchronization technique employed for the proposed MPSA system. This figure illustrates the process of creating a multimodal dataset by synchronizing and combining two unimodal datasets. Each unimodal dataset contains samples from the same set of classes (pain levels). Through data synchronization, the samples corresponding to the same class from both datasets (e.g., h_1 number of samples from Dataset 1 and k_1 from Dataset 2 for the i^{th} class) are aligned. These synchronized samples are further employed in multimodal systems, where they are typically concatenated to form a unified representation, e.g., $h_1 \times k_1$ samples belong to the i^{th} class. This results in a multimodal dataset where each class instance is enriched with information from both modalities, enabling more robust and accurate classification by utilizing the complementary strengths of each data type.

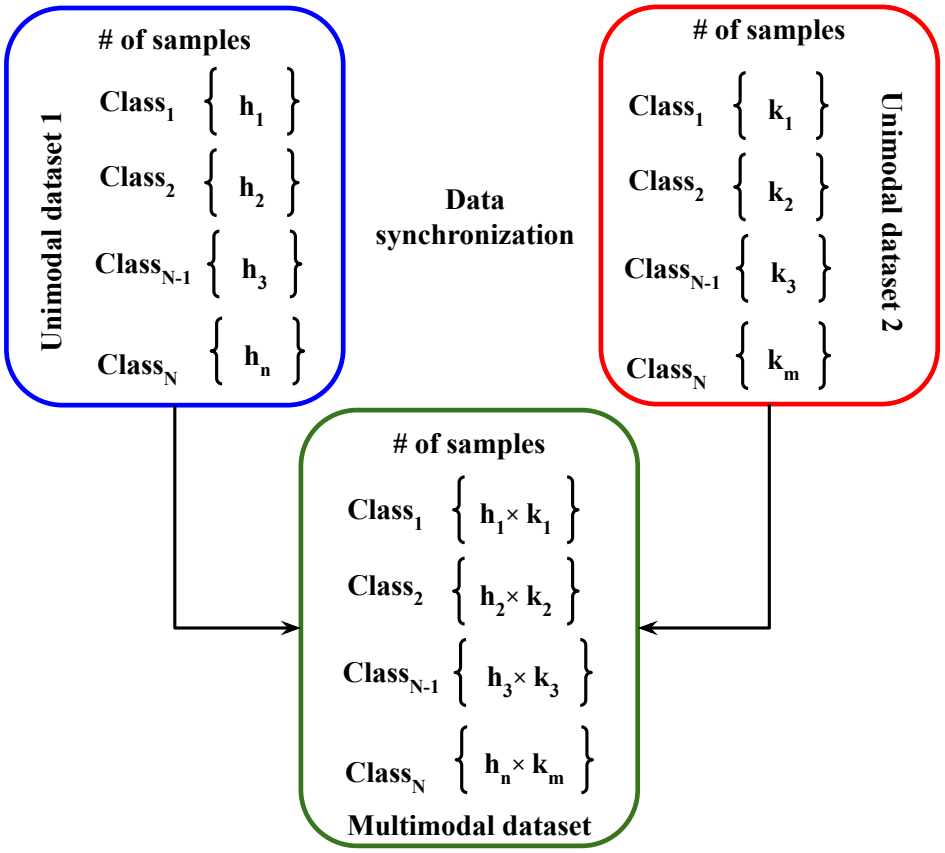


FIGURE 6.2: Data synchronization model employed in this work.

For an example, a video is synchronized based on its frame rate, audio is divided into short time windows that match the video frames, and text features are linked

to speech segments or specific time intervals [379, 380]. Methods like Dynamic Time Warping (DTW), interpolation, and attention-based alignment are used to fix timing mismatches between these data types [381]. These methods also help deal with problems like missing data or different sampling speeds.

6.2.2 Modalities Fusion

This chapter focuses on the modalities of text data, audio data, and video samples. Image data has not been separately considered, as video frames inherently include image content. Moreover, for real-time applications, these selected modalities are more suitable and representative for dynamic pain detection systems. Since for the proposed MPSA system, the post classification fusion techniques have been employed, so, the best prediction models obtained for the considered modalities from the corresponding chapters have been considered here for the experimentation. The list of these prediction models have been reported in Table 6.1.

TABLE 6.1: List of best prediction model considered for the MPSA system.

Modality	System	Best prediction model	Declared model
Text	DLPSA$_{text}$	f_{bilstm}	f_{text}
Audio	DLPSA$_{audio}$	DLPSA$_{audio}$	f_{audio}
Video	DLPSA$_{video}$	$PainAttentionCapsule$	f_{video}

So, the f_{text}, f_{audio}, and f_{video} trained models are used to obtain the classification scores when testing these models with the corresponding test data types. Here, suppose a text data t undergoes to f_{text}, it predicts $TC_1, \cdots, TC_n$ scores of n-class pain problem. Similarly for an audio test data a, the f_{audio} derives $AC_1, \cdots, AC_n$ scores of n-class pain problem, and for a video data v, f_{video} produces $VC_1, \cdots, VC_n$ scores of n-class pain problem. So, these scores are fused together to make the final decisions.

- **Sum Score Level Fusion (SSLF):** In SSLF, at first, the classification score of each model is obtained, then scores of the corresponding classes are added. Here, the SSLF for the three MPSA Systems is defined as:

  i) $TA_i = (TC_i + AC_i), \ \forall i$

ii) $AV_i = (AC_i + VC_i),\ \forall i$

iii) $TAV_i = (TC_i + AC_i + VC_i),\ \forall i$

- **Product Score Level Fusion (PSLF):** In PSLF, at first, the classification score of each model is obtained, then the scores of the corresponding classes are multiplied. Here, the PSLF for the three MPSA Systems is defined as:

  i) $TA_i = (TC_i \times AC_i),\ \forall i$

  ii) $AV_i = (AC_i \times VC_i),\ \forall i$

  iii) $TAV_i = (TC_i \times AC_i \times VC_i),\ \forall i$

- **Weighted Sum Product Score Level Fusion (WSSLF):** In WSSLF, at first, the classification score of each model is obtained, then the scores of the corresponding classes are weighted-multiplied summed. Here, the WSSLF for the three MPSA Systems is defined as:

  i) $TA_i = (w_1 \times TC_i + w_2 \times AC_i),\ \forall i,\ 0 \leq w_1, w_2 \leq 1,\ w_1 + w_2 = 1$

  ii) $AV_i = (w_1 \times AC_i + w_2 \times VC_i),\ \forall i,\ 0 \leq w_1, w_2, \leq 1,\ w_1 + w_2 = 1$

  iii) $TAV_i = (w_1 \times TC_i + w_2 \times AC_i + w_3 \times VC_i),\ \forall i,\ 0 \leq w_1, w_2, w_3 \leq 1,\ w_1 + w_2 + w_3 = 1$

- **Decision-level fusion (DLF):** It is also called a late fusion technique used in multimodal systems where the outputs from individual models—each trained on separate data modalities such as text, audio, or video—are combined to make a final decision. In this chapter, the considered modalities include text data, audio signals, and video sequences. Accordingly, the models f_{text} for DLPSA$_{text}$, f_{audio} for DLPSA$_{audio}$, and f_{video} for DLPSA$_{video}$ are trained separately on their respective data types. For a given test sample, each model outputs a binary decision corresponding to the presence (B_1) or absence (B_0) of pain-related sentiment.

  Let us consider pain level is 'Low Pain (PI$_1$)', a text sample t passed through f_{bilstm} yields a binary outcome TB_0 or TB_1 with respect to the pain level 'Low Pain (PI$_1$)'; an audio sample a evaluated with f_{audio} produces AB_0 or AB_1 with respect to the pain level 'Low Pain (PI$_1$)'; similarly, a video sample v processed by f_{video} results in VB_0 or VB_1 with respect to the pain level 'Low

TABLE 6.2: Types of MPSA systems.

Fused System	**Involved Modalities**
MPSA_{TA} System	DLPSA_{text} + DLPSA_{audio}
MPSA_{AV} System	DLPSA_{audio} + DLPSA_{video}
MPSA_{TAV} System	DLPSA_{text} + DLPSA_{audio} + DLPSA_{video}

Pain (PI_1)'. These modality-specific binary decisions are then integrated using decision-level fusion techniques, such as majority voting or rule-based logic, to arrive at the final prediction outcome of the proposed MPSA system. The majority voting is one of the most widely used decision-level fusion techniques in multimodal systems due to its simplicity, robustness, and interpretability. In the context of pain sentiment detection, where binary classification is performed across heterogeneous modalities (text, audio, video), majority voting allows the system to arrive at a final decision based on the consensus of individual model predictions. This method is particularly effective when the modalities contribute comparably and may occasionally be affected by noise or missing information. Majority voting helps mitigate the influence of any one unreliable modality, making it a suitable choice for clinical and real-world applications [353]. So, in this thesis, the OR-ing-based technique has been adopted. This method does not sum class scores; instead, it relies on individual binary predictions to form a consensus.

In this chapter, three types of MPSA systems have been proposed, which are listed in Table 6.2. To design a multimodal system, PSA_{image} System (Chapter 4) is not considered; instead, we consider video. Since video-based pain detection using facial expressions is more suitable for real-time applications than image-based approaches. Video enables the capture of dynamic changes in facial muscle movements associated with pain expressions, which static images may fail to detect. Moreover, video-based recognition offers a more practical solution for real-world scenarios such as clinical monitoring, assistive technologies, and remote healthcare systems. These multimodal systems are discussed as follows

1. **MPSA_{TA} System:** In MPSA_{TA}, fusion is performed between the DLPSA_{text}

System and the DLPSA$_{audio}$ System using some fusion approaches. As unimodal systems, both DLPSA$_{text}$ and DLPSA$_{audio}$ have certain limitations. Individuals who are illiterate may find it difficult to express their pain through written comments, but can communicate effectively through vocal expressions. Conversely, individuals with speech impairments may not be able to convey their pain vocally but can describe their experience through written text. To address these complementary limitations and improve the overall system performance, the DLPSA$_{text}$ System is fused with the DLPSA$_{audio}$ System. The details of DLPSA$_{text}$ and DLPSA$_{audio}$ Systems have already been discussed in Chapters 2 and 3, respectively. These chapters cover the complete pipeline, including data preprocessing, feature extraction, classification techniques, and performance evaluations. In MPSA$_{TA}$, the best performing models f_{text} and f_{audio} have been employed for the above score-level and decision-level fusions. The block diagram of this system has been shown in Fig. 6.3.

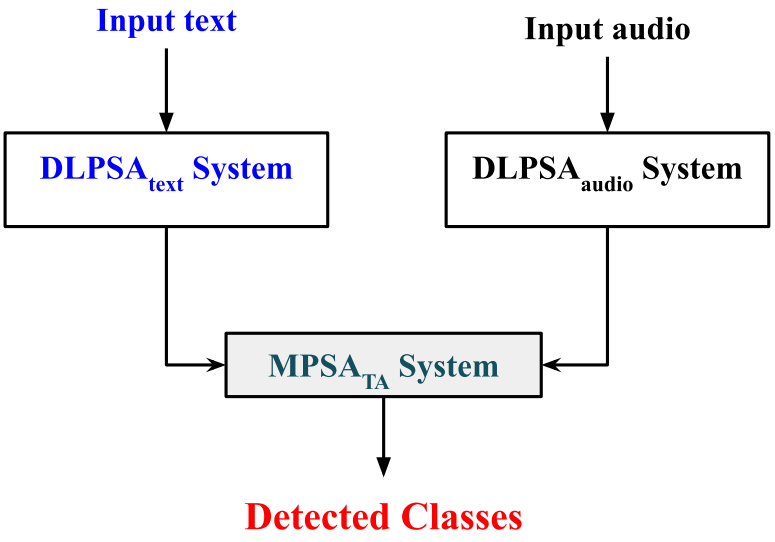


FIGURE 6.3: Block Diagram of the proposed MPSA$_{TA}$ system.

2. **MPSA$_{AV}$ System:** The MPSA$_{TA}$ system is capable of detecting pain from both text and audio inputs; however, it has certain limitations. Elderly patients often face difficulties in expressing their pain through writing or may lack the vocal strength required to convey emotions effectively. Moreover, it is sometimes challenging to distinguish genuine vocal expressions of pain from acted ones. To overcome these issues, facial expressions in the form of video data have been incorporated through the DLPSAS$_{video}$ System, resulting in an

enhanced framework referred to as MPSA$_{AV}$. Chapter 5 provides a detailed overview of the DLPSAS$_{video}$ System, covering the entire pipeline including data preprocessing, feature extraction, classification, and experimental evaluation. Several models are assessed, and the best-performing model is selected for integration into MPSA$_{AV}$ to improve its accuracy and reliability in pain assessment. The block diagram of this system has been shown in Fig. 6.4.

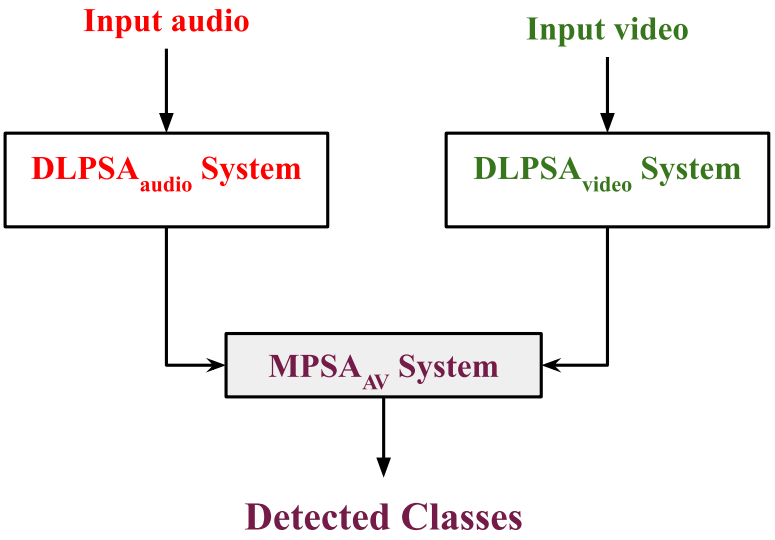


FIGURE 6.4: Block Diagram of the proposed MPSA$_{AV}$ system.

3. **MPSA$_{TAV}$ System:** This system offers advantages over MPSA$_{TA}$ and MPSA$_{AV}$ by incorporating three different data modalities, thereby enabling more comprehensive pain level identification. MPSA$_{TAV}$ has been developed by integrating the MPSA$_{TA}$ and MPSA$_{AV}$. Among the various models evaluated, the best-performing model has been selected and incorporated into MPSA$_{TAV}$ to enable more accurate and temporally-aware pain severity assessment. The block diagram of this system has been shown in Fig. 6.5.

In this thesis, three multimodal PSA systems have been proposed, such as MPSA$_{TA}$System (Multimodal system combination of Text and Audio), MPSA$_{AV}$System (Multimodal system combination of Audio and Video) and MPSA$_{TAV}$System (Multimodal system combination of Text, Audio and Video) (refer to Table 6.2 of the revised thesis). From these multimodal systems it has been observed that the performance of MPSA$_{AV}$System is better than MPSA$_{TA}$System, and the performance of MPSA-TAV System is much better than MPSA$_{AV}$System and MPSA$_{TA}$System. There

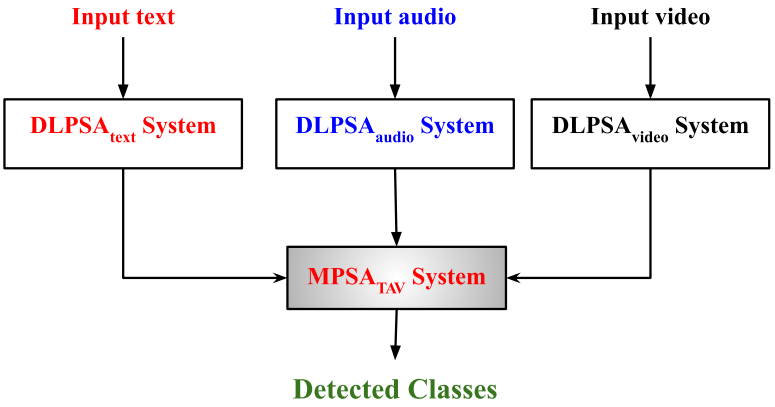


FIGURE 6.5: Block Diagram of the proposed MPSA$_{TAV}$ system.

is cumulative performance improvement has been seen from MPSA$_{TA}$System to MPSA$_{AV}$System to MPSA$_{TAV}$System. While the use of Multimodal system combination of Text and Video has a drop-off performance in the cumulative performance growth. At the same time, the possibility of getting data of same patient is more challenging than the considered other three multimodal systems.

PSA$_{audio}$Systems has been used as the bridge between PSA$_{text}$Systems and PSA$_{video}$Systems, it has also been observed that the fusion combination with only Text and Video reduces the performance; thus, this combination has not been included.

6.3 Experiments and Results

By using the above fusion techniques, the following multimodal systems have been experimented and validated.

6.3.1 MPSA$_{TA}$ System

In the MPSA$_{TA}$ system, the prediction model f_{text} of PSA$_{text}$ is employed to compute classification scores for test samples from TD_{aggr} (3-class problem with class distributions: TC_1: 2120, TC_2: 1354, TC_3: 2526) and TD_{llm} (5-class problem with TC_1–TC_5: 100 samples each). Accordingly, f_{text} produces class-wise score vectors (t_1, t_2, t_3) for each 3-class sample and $(t_1, t_2, t_3, t_4, t_5)$ for each 5-class sample. Similarly, the prediction model f_{audio} of PSA$_{audio}$ is used for audio-based pain sentiment

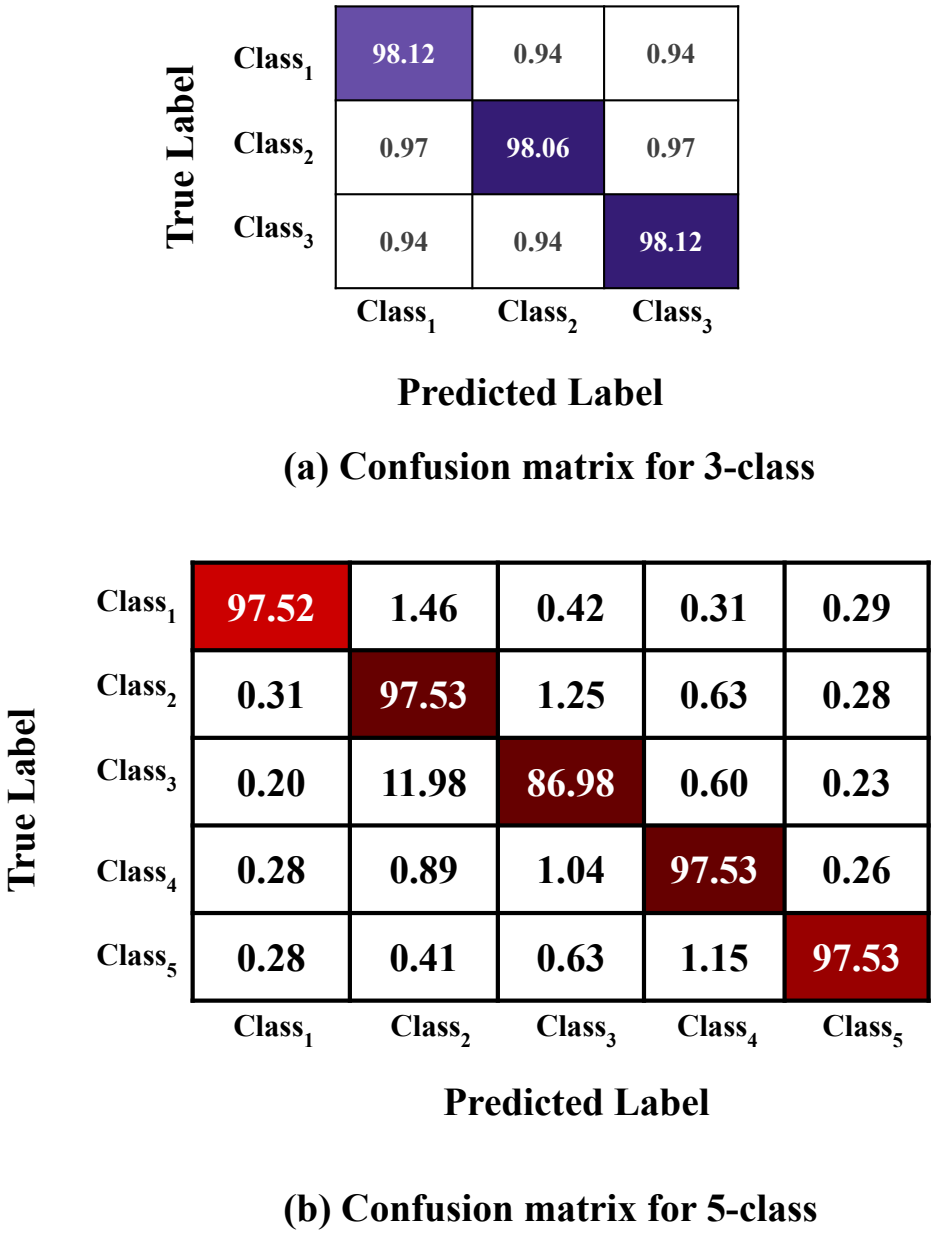
classification over AD_{VIVAE} (3-class: AC_1: 265, AC_2: 136, AC_3: 141) and $\text{AD}_{\text{RAVDESS}}$ (5-class: AC_1: 48, AC_2–AC_4: 192, AC_5: 96). The model f_{audio} generates corresponding class score vectors (a_1, a_2, a_3) and $(a_1, a_2, a_3, a_4, a_5)$ for the respective classification tasks. During score-level fusion (SLF), the score vectors from text and audio modalities for a given sample belonging to the same pain class category are combined using fusion strategies such as SSLF, PSLF, and WSSLF, as discussed earlier. This approach is applied to both 3-class and 5-class problems. The detailed performance analysis of these SLF techniques under the MPSA_{TA} system is presented in Table 6.3. The performance metrics presented in this table clearly demonstrate the effectiveness of various pain sentiment analysis (PSA) systems across both 3-class and 5-class classification tasks. Among all systems, the proposed weighted score level fusion (WSSLF) technique consistently achieves the highest performance across all metrics, with an F1-score of 98.14% and 97.72%, and an accuracy of 98.90% and 97.79% for 3-class and 5-class tasks, respectively. This indicates the robustness and generalizability, confirming that the WSSLF technique provides more accurate and balanced predictions for the MPSA_{TA} system. Fig. 6.6 shows confusion matrices for 3-class and 5-class MPSA_{TA} system using WSSLF technique. The performance of the MPSA_{TA} system using the DLF technique has also been demonstrated in Table 6.3, which highlights that the **DLF** technique surpasses both unimodal approaches by achieving near-perfect precision, recall, F1-score, and accuracy values in both classification scenarios, with a perfect score of 99.99% in the 5-class task.

TABLE 6.3: Performance of MPSA_{TA} System.

System	Accuracy	F1-score	Precision	Recall	Accuracy	F1-score	Precision	Recall
	3-class				5-class			
PSA_{text}	68.59	68.34	67.71	69.52	73.22	74.77	72.58	75.31
$\text{PSA}_{\text{audio}}$	98.05	97.24	96.72	98.38	97.84	96.39	96.18	98.13
SSLF	95.24	94.69	94.88	94.5	97.52	96.96	97.15	96.77
PSLF	96.06	95.34	94.77	95.91	97.60	97.03	97.22	97.84
WSSLF	98.90	98.14	98.20	98.08	97.79	97.72	97.91	97.53
DLF	98.88	98.88	98.88	98.88	99.99	99.99	99.99	99.99

6.3.2 MPSA_{AV} system

In the MPSA_{AV} system, the audio-based prediction model f_{audio} from $\text{PSA}_{\text{audio}}$ is utilized to generate classification scores for test samples from the datasets AD_{VIVAE}



(a) Confusion matrix for 3-class

(b) Confusion matrix for 5-class

FIGURE 6.6: Illustration of 3-class and 5-class confusion matrices in percentage for the MPSA$_{TA}$ system using WSSLF technique.

and AD$_{\text{RAVDESS}}$. This model produces class score vectors of the form (a_1, a_2, a_3) and $(a_1, a_2, a_3, a_4, a_5)$ for the respective 3-class and 5-class classification tasks. Similarly, the video-based prediction model f_{video} from PSA$_{\text{video}}$ is applied to the VD$_{\text{BioVid}}$ dataset for pain sentiment classification, generating class-wise score vectors (v_1, v_2, v_3) for the 3-class setting (VC_1: 2800, VC_2: 5600, VC_3: 5600) and $(v_1, v_2, v_3, v_4, v_5)$ for

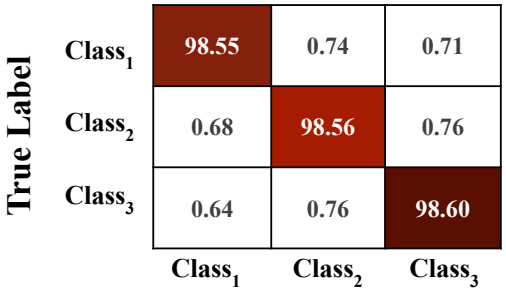
the 5-class setting (VC_1-VC_5: 2800). During the score-level fusion (SLF) stage, the corresponding score vectors from the audio and video modalities each representing the same pain class are integrated using fusion strategies such as SSLF, PSLF, and WSSLF, as previously described.

TABLE 6.4: Performance of MPSA$_{AV}$ System..

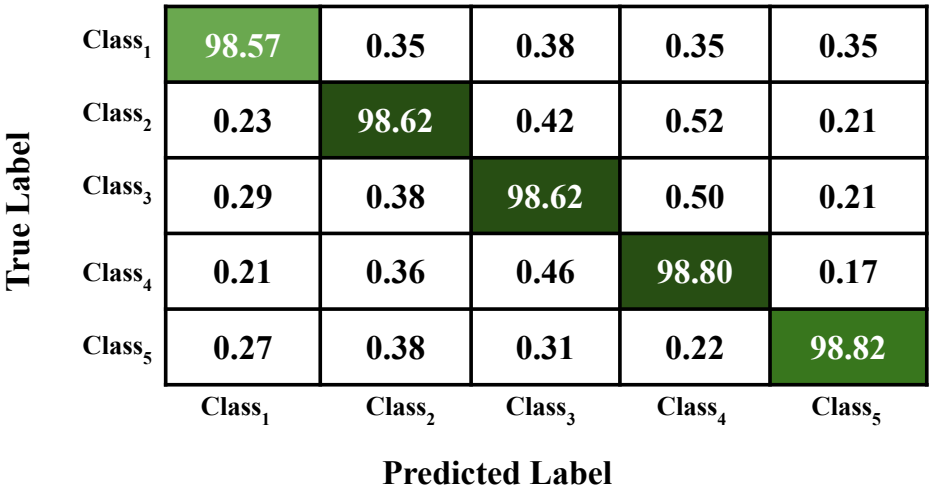
System	Accuracy	F1-score	Precision	Recall	Accuracy	F1-score	Precision	Recall
	3-class				5-class			
PSA$_{\text{audio}}$	98.05	97.24	96.72	98.38	97.84	96.39	96.18	98.13
PSA$_{\text{video}}$	86.49	86.19	86.54	86.00	65.64	65.51	66.41	65.63
SSLF	97.82	97.17	97.23	97.09	94.45	93.77	93.17	94.34
PSLF	98.14	97.39	97.45	97.33	95.22	94.50	95.94	95.06
WSSLF	99.13	98.62	98.68	98.56	98.26	97.03	96.45	98.61
DLF	99.99	99.99	99.99	99.99	97.66	97.66	97.66	97.66

Table 6.4 shows the results of SSFL, PSFL, and WSSFL techniques for the MPSA$_{AV}$ system. From this table, it has been observed that the performance comparison among the Score-Level Fusion (SLF) strategies SSLF, PSLF, and WSSLF within the proposed MPSA$_{AV}$ system highlights the progressive impact of increasingly sophisticated fusion mechanisms. The SSLF (Simple Score-Level Fusion) method yields competent results (Precision: 97.23%, Recall: 97.09%, F1-Score: 97.17%, Accuracy: 97.82%), but its lower test set performance (Accuracy: 94.45%) suggests limited adaptability to unseen data. The PSLF (Parametric Score-Level Fusion) improves upon this with higher generalization capability (Accuracy: 95.22%), likely due to its ability to weigh scores based on learned parameters. The best results are achieved by the WSSLF (Weighted Score-Level Fusion), which effectively balances the contribution of modalities based on their confidence or reliability, achieving top scores across both validation and test sets (Accuracy: 99.13% and 98.26%, respectively). These results justify the adoption of WSSLF as the most effective SLF technique, offering enhanced robustness and predictive reliability for the MPSA$_{AV}$ system. Table 6.4 shows the results of the MPSA$_{AV}$ system using the decision level fusion (DLF) technique. This table shows that in the 3-class scenario, the system achieves near-perfect performance with 99.99% in precision, recall, F1-score, and accuracy, indicating highly consistent and precise predictions. Even in the more challenging 5-class classification, the system maintains strong results with all metrics uniformly at 97.66%. These results demonstrate that the DLF-based MPSA$_{AV}$ framework effectively combines complementary information from audio and video modalities,

leading to enhanced robustness and generalization. The superior and stable performance across both classification tasks validates the efficacy of decision-level fusion in multimodal pain sentiment analysis. Fig. 6.7 shows confusion matrices for 3-class and 5-class MPSA$_{AV}$ system using WSSLF technique.



(a) Confusion matrix for 3-class



(b) Confusion matrix for 5-class

FIGURE 6.7: Illustration of 3-class and 5-class confusion matrices in percentage for the MPSA$_{AV}$ system using WSSLF technique.

6.3.3 MPSA$_{TAV}$ System

In the MPSA$_{TAV}$ system, the prediction model f_{text} of PSA$_{text}$, f_{audio} from PSA$_{\text{audio}}$, and f_{video} from PSA$_{\text{video}}$ are combined togather using score level fusion (SLF), and

decision level fusion (DLF). In case of SLF techniques for MPSA_{TA} and MPSA_{AV} systems, it has been observed that in both these systems, the weighted-sum score level fusion (WSSLF) technique outperforms. So, for the MPSA_{TAV} system, the performance of the proposed system concerning WSSLF and DLF techniques results have been demonstrated in Table 6.5. This table shows the performance results for the MPSA_{TAV} system using two distinct fusion strategies, WSSLF for score-level fusion and DLF for decision-level fusion, demonstrating the effectiveness of multimodal integration for pain sentiment classification. When employing WSSLF, the system achieves high precision, recall, F1-score, and accuracy across both 3-class (up to 99.67%) and 5-class (up to 99.56%) tasks, indicating reliable and consistent performance through the weighted combination of audio, video, and text modality scores. However, with DLF, MPSA_{TAV} attains near-perfect results, reaching 100% across all metrics for 3-class classification and 99.99% in the 5-class scenario. This substantial improvement highlights the superiority of decision-level fusion in capturing complex inter-modal dependencies by leveraging the final predictions from each unimodal classifier. Overall, the results confirm that both fusion approaches significantly enhance model performance, with DLF offering the most accurate and robust solution for multimodal pain sentiment analysis. Fig. 6.8 shows confusion matrices for 3-class and 5-class MPSA_{TA} system using WSSLF technique.

TABLE 6.5: Performance of MPSA_{TAV} System.

System	**Accuracy**	**F1-score**	**Precision**	**Recall**	**Accuracy**	**F1-score**	**Precision**	**Recall**
	3-class				**5-class**			
PSA_{text}	68.59	68.34	67.71	69.52	73.22	74.77	72.58	75.31
$\text{PSA}_{\text{audio}}$	98.05	97.24	96.72	98.38	97.84	96.39	96.18	98.13
$\text{PSA}_{\text{video}}$	86.49	86.19	86.54	86	65.64	65.51	66.41	65.63
WSSLF	99.35	99.21	99.53	99.67	99.16	98.79	99.31	99.56
DLF	100	100	100	100	99.99	99.99	99.99	99.99

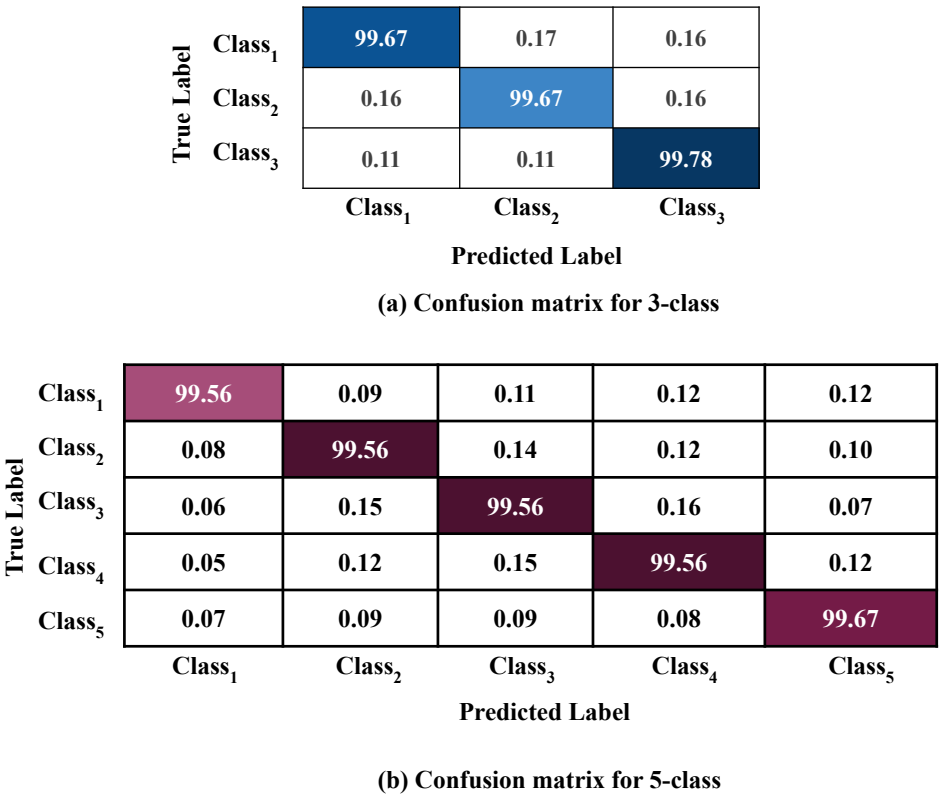


FIGURE 6.8: Illustration of 3-class and 5-class confusion matrices in percentage for the MPSA$_{TAV}$ system using WSSLF technique.

6.4 Conclusions

This chapter introduces Multimodal-Based Pain Sentiment Analysis (MPSA) Systems, which demonstrate improved performance over the previously developed unimodal PSA Systems. The development of these systems begins with the integration of three individual modalities: PSA$_{\text{text}}$, PSA$_{\text{audio}}$, and PSA$_{\text{video}}$. The rationale behind combining these modalities is that text inputs, audio features, and video cues capture complementary affective information across different channels, which is essential for addressing the multifaceted and subjective nature of pain. Based on this, three distinct MPSA systems—MPSA$_{TA}$, MPSA$_{AV}$, and MPSA$_{TAV}$—have been proposed. These systems are designed to effectively handle data synchronization and multimodal fusion. To ensure reliable and accurate fusion, the data from all considered modalities are semantically aligned, thereby enhancing the ability of the system to consistently interpret pain-related cues across varied input sources. This semantic

alignment improves both the reliability and performance of multimodal pain sentiment analysis. Given the dynamic nature of pain detection, post-classification fusion techniques have been employed in this chapter, specifically score-level fusion (SLF) and decision-level fusion (DLF). Fusion is performed using the best-performing prediction models from each unimodal system. Experimental results show that the Weighted Sum Score Level Fusion (WSSLF) technique provides the most accurate and balanced predictions across the MPSA$_{TA}$, MPSA$_{AV}$, and MPSA$_{TAV}$ systems. Among them, the MPSA$_{TAV}$ system consistently outperforms the others, benefiting from the integration of all three modalities—text, audio, and video. Similarly, under the decision level fusion strategy, MPSA$_{TAV}$ again achieves the highest performance. The comparative performance of the fusion techniques is further analyzed with respect to several key aspects, including input type, granularity, flexibility, interpretability, computational cost, and suitability. These findings underline the uniqueness and robustness of the proposed MPSA systems.

The implementation of the MPSA system presented in this chapter significantly enhances the overall performance and robustness of the pain sentiment analysis framework proposed in this thesis. By integrating the best-performing prediction models from the unimodal systems developed in previous chapters, this chapter constructs a comprehensive multimodal system. This system effectively addresses several critical challenges associated with pain sentiment analysis, such as modality-specific limitations, ambiguity in emotional expression, and inconsistencies across input types. Through rigorous experimentation on diverse and real-time PSA datasets, the MPSA system demonstrates superior accuracy and generalizability. These results validate the effectiveness of the multimodal approach in delivering a reliable and scalable solution for automated pain assessment. The conclusions and overall contributions of this thesis are summarized in the following chapter.

Chapter 7

Conclusions and Future Scope

This thesis offers a thorough exploration of pain sentiment analysis by systematically examining four modalities—text, audio, image, and video—to develop effective pain assessment systems. The process begins with text-based analysis, where conventional feature extraction techniques such as Bag of Words (BoW) and Term Frequency–Inverse Document Frequency (TF-IDF) are considered, as well as advanced deep learning models like LSTM and BiLSTM are considered. While text data provides rich subjective information about patients' pain experiences in their own words, it faces limitations in capturing nonverbal cues and is affected by cultural and linguistic variations in pain expression. Among the evaluated methods, the BiLSTM model ($\text{DLPSA}_{\text{text}}$ System) emerges as the most effective for text-based classification, highlighting the importance of capturing semantic meaning in pain descriptions.

To address the limitations of text-based analysis, this thesis proposes audio-based pain sentiment analysis, which leverages vocal biomarkers such as pitch modulation, disrupted speech rhythm, and voice instability. The integration of statistical, MFCC, and spectral features with fully connected network classifiers ($\text{DLPSA}_{\text{audio}}$ System) outperforms conventional machine learning methods, offering real-time, non-invasive capabilities crucial for telemedicine applications. However, audio analysis continues to face challenges in distinguishing genuine pain from deceptive vocal expressions, highlighting the need for additional modalities to enhance reliability.

The challenges of audio-based systems lead to the development of image-based analysis, where facial expressions serve as observable physiological indicators of pain. The combined use of various deep CNN architectures provides greater resistance to manipulation, though it comes at the cost of limited capacity to process the temporal evolution of pain. Further, to resolve the temporal limitations of static images while building, we strengthen the analysis system by considering video. Proposed *PainCapsule* and *PainAttentionCapsule* ($\text{DLPSA}_{\text{video}}$ System) models, improve the performance in comparison to $\text{PSA}_{\text{image}}$. This approach effectively captures spatiotemporal expressions of pain and offers the most comprehensive and realistic pain assessment among all the methods evaluated.

The progressive enhancements observed in text, audio, image and video-based analyses individually emphasize the importance of integrating diverse data types into a unified system through multimodal fusion techniques. Such integration significantly strengthens the robustness and suitability of the system. To achieve this, both score-level and decision-level fusion methods are applied to combine information from different modalities. Among the proposed systems, the multimodal pain sentiment analysis system MPSA_{TAV} consistently delivers the best performance by effectively utilizing the combined strengths of text, audio, and video inputs. The results highlight the distinctiveness and reliability of the proposed MPSA systems.

Ultimately, this thesis demonstrates that multimodal pain sentiment analysis offers one of the most reliable and comprehensive approaches to pain assessment. By addressing the limitations of individual data modalities through complementary information sources and innovative modeling techniques, this work establishes a strong foundation for future advancements in autonomous pain recognition systems. The contributions made in this study are significant for both the scientific field of affective computing and the practical domain of healthcare, paving the way for more objective and real-time pain assessment tools that can improve patient care and outcomes across diverse clinical settings.

7.1 Future Research Directions

Looking ahead, there are many promising directions for future research in multimodal pain sentiment analysis. Future work can explore better ways to detect and remove fake pain expressions by using additional signals like heart rate or skin responses. Creating larger and more diverse datasets that include people from different backgrounds and pain types can help make the models more reliable. Developing AI models that can clearly explain their decisions will also help gain trust in clinical use. Protecting patient information is another important issue, and future efforts will focus on improving data privacy. In addition, this multimodal approach can be extended to other areas such as emotion detection and mental health assessment, showing the wider usefulness of the methods developed in this research.

Bibliography

[1] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[2] B. Pang and L. Lee, "Opinion mining and sentiment analysis foundations and trends in information retrieval vol. 2," 2008.

[3] B. Liu, *Sentiment analysis and opinion mining*, vol. 5. Springer, 2012.

[4] E. Cambria and A. Hussain, "Affective computing and sentiment analysis," *IEEE Intelligent Systems*, vol. 31, no. 2, pp. 102–107, 2016.

[5] R. Potapova and V. Potapov, "Sentiment analysis of digital communication," *Bulletin of Moscow State Linguistic University. Humanities*, no. 4 (872), pp. 86–96, 2023.

[6] A. Yadav and D. K. Vishwakarma, "A comparative study on bio-inspired algorithms for sentiment analysis," *Cluster Computing*, vol. 23, no. 4, pp. 2969–2989, 2020.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *NAACL-HLT*, 2019.

[8] H. Qureshi and A. P. Agrawal, "Video based sentiment analysis," in *2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT)*, pp. 388–399, IEEE, 2022.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[11] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2000.

[12] M. El Ayadi, M. S. Kamel, and F. Karray, "A survey on feature extraction techniques for speech recognition," *Pattern recognition*, vol. 44, no. 3, pp. 585–601, 2011.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, IEEE, 2005.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[19] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, pp. 4489–4497, 2015.

[20] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1725–1732, 2014.

[21] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86, 2002.

[22] M. Mursalin, Y. Zhang, M. Islam, and P. Andersen, "Automated epilepsy detection using deep learning approach based on deep neural network combined with random forest classifier," *Healthcare technology letters*, vol. 4, no. 6, pp. 181–187, 2017.

[23] W. S. Noble, "What is a support vector machine?," *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[24] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*, vol. 398. John Wiley & Sons, 2013.

[25] J. R. Quinlan, *Induction of decision trees*, vol. 1. 1986.

[26] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[27] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Routledge, 1984.

[28] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Thirteenth annual conference of the international speech communication association*, 2012.

[29] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[30] B. Gambäck and U. K. Sikdar, "Using convolutional neural networks to classify hate-speech," in *Proceedings of the First Workshop on Abusive Language Online*, pp. 85–90, 2017.

[31] A. Joshi, P. Bhattacharyya, and M. J. Carman, "Harnessing context incongruity for sarcasm detection," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 757–762, 2017.

[32] M. Ebrahimi, A. H. Yazdavar, and A. Sheth, "Challenges of sentiment analysis for dynamic events," *IEEE Intelligent Systems*, vol. 32, no. 5, pp. 70–75, 2017.

[33] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[34] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams engineering journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[35] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 15–21, 2013.

[36] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language resources and evaluation*, vol. 39, no. 2, pp. 165–210, 2005.

[37] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, *et al.*, "Semeval-2016 task 5: Aspect based sentiment analysis," in *International workshop on semantic evaluation*, pp. 19–30, Springer, 2016.

[38] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," vol. 50, pp. 723–762, 2014.

[39] C. Banea, R. Mihalcea, and J. Wiebe, "Multilingual sentiment analysis on twitter," *Proceedings of the Workshop on Semantic Analysis in Social Media*, pp. 19–26, 2012.

[40] B. Schuller and A. Batliner, "Voice analytics: Sentiment analysis in speech," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 103–111, 2013.

[41] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," *IEEE ICASSP*, pp. 2227–2231, 2017.

[42] S. R. Livingstone and F. A. Russo, "The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english," *PloS one*, vol. 13, no. 5, p. e0196391, 2018.

[43] D. Ververidis and C. Kotropoulos, "Automatic speech emotion recognition," *Signal Processing*, vol. 86, no. 12, pp. 3519–3534, 2006.

[44] B. Schuller, S. Steidl, and A. Batliner, "The opensmile toolkit," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 154–156, 2018.

[45] S. Poria, E. Cambria, R. Bajpai, and A. Hussain, "A review on multimodal sentiment analysis," *IEEE Transactions on Affective Computing*, vol. 9, no. 2, pp. 228–250, 2017.

[46] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.

[47] G. Meena, K. K. Mohbey, S. Kumar, R. K. Chawda, and S. V. Gaikwad, "Image-based sentiment analysis using inceptionv3 transfer learning approach," *SN Computer Science*, vol. 4, no. 3, p. 242, 2023.

[48] D. C. Neth, *Facial configuration and the perception of facial expression.* PhD thesis, The Ohio State University, 2007.

[49] E. Cambria, D. Hazarika, S. Poria, A. Hussain, and R. Subramanyam, "Benchmarking multimodal sentiment analysis," in *International Conference on Computational Linguistics and Intelligent Text Processing*, pp. 166–179, Springer, 2017.

[50] Q. You, J. Luo, H. Jin, and J. Yang, "Robust image sentiment analysis using progressively trained and domain transferred deep networks," *arXiv preprint arXiv:1509.06041*, 2015.

[51] H. Li and H. Xu, "Video-based sentiment analysis with hvnlbp-top feature and bi-lstm," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9963–9964, 2019.

[52] S. Butler, J. Tanaka, M. Kaiser, and R. Le Grand, "Mixed emotions: Holistic and analytic perception of facial expressions," *Journal of Vision*, vol. 9, no. 8, pp. 496–496, 2009.

[53] E. Sariyanidi, H. Gunes, and A. Cavallaro, "Automatic analysis of facial affect: A survey of registration, representation, and recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 6, pp. 1113–1133, 2014.

[54] P. Ekman and D. Keltner, "Universal facial expressions of emotion," *California mental health research digest*, vol. 8, no. 4, pp. 151–158, 1970.

[55] L. S. Chen, T. S. Huang, T. Miyasato, and R. Nakatsu, "Multimodal human emotion/expression recognition," in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 366–371, IEEE, 1998.

[56] L. C. De Silva, T. Miyasato, and R. Nakatsu, "Facial emotion recognition using multi-modal information," in *Proceedings of ICICS, 1997 International Conference on Information, Communications and Signal Processing. Theme: Trends in Information Systems Engineering and Wireless Multimedia Communications (Cat.*, vol. 1, pp. 397–401, IEEE, 1997.

[57] S. Poria, E. Cambria, R. Bajpai, and A. Hussain, "Multimodal sentiment analysis: Addressing key issues and setting up the baselines," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 17–25, 2017.

[58] M. Cascella, D. Schiavo, A. Cuomo, A. Ottaiano, F. Perri, R. Patrone, S. Migliarelli, E. G. Bignami, A. Vittori, F. Cutugno, *et al.*, "Artificial intelligence for automatic pain assessment: research methods and perspectives," *Pain Research and Management*, vol. 2023, 2023.

[59] S. S. Rajagopalan, L.-P. Morency, T. Baltrusaitis, and R. Goecke, "Extending long short-term memory for multi-view structured learning," in *Computer*

Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14, pp. 338–353, Springer, 2016.

[60] T. C. Kietzmann, A. L. Gert, F. Tong, and P. König, "Representational dynamics of facial viewpoint encoding," *Journal of cognitive neuroscience*, vol. 29, no. 4, pp. 637–651, 2017.

[61] L. Watts, "Synchronous and asynchronous communication in distance learning: A review of the literature," *Quarterly Review of Distance Education*, vol. 17, no. 1, p. 23, 2016.

[62] S. M. Mohammad and S. Kiritchenko, "Even the abstract have character: Recognizing emotions and personality traits in the text of the abstracts of scientific articles," *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 343–353, 2018.

[63] R. Schwartz, L. Vasserman, S. Feldman, and J. Dodge, "Ethics in sentiment analysis," *AI and Ethics*, vol. 1, no. 1, pp. 1–13, 2019.

[64] W. Guo, J. Wang, and S. Wang, "Deep multimodal representation learning: A survey," *Ieee Access*, vol. 7, pp. 63373–63394, 2019.

[65] P. Werner, A. Al-Hamadi, K. Limbrecht-Ecklundt, S. Walter, S. Gruss, and H. C. Traue, "Automatic pain assessment with facial activity descriptors," *IEEE Transactions on Affective Computing*, vol. 8, no. 3, pp. 286–299, 2016.

[66] J.-F. Payen, O. Bru, J.-L. Bosson, A. Lagrasta, E. Novel, I. Deschaux, P. Lavagne, and C. Jacquot, "Assessing pain in critically ill sedated patients by using a behavioral pain scale," *Critical care medicine*, vol. 29, no. 12, pp. 2258–2263, 2001.

[67] A. C. Meltzer, J. M. Pines, L. M. Richards, P. M. Mullins, and M. Mazer-Amirshahi, "Text mining for pain assessment," *Academic Emergency Medicine*, vol. 23, no. 11, pp. 1202–1210, 2016.

[68] S. Hossain, S. Umer, R. K. Rout, and M. Tanveer, "Fine-grained image analysis for facial expression recognition using deep convolutional neural networks with bilinear pooling," *Applied Soft Computing*, vol. 134, p. 109997, 2023.

[69] P. D. Mehta, E. M. Roth, S. Thakur, D. Choi, and B. D. Darnall, "Natural language processing for clinical pain research: A scoping review," *Pain Reports*, vol. 5, no. 6, p. e860, 2020.

[70] K. M. Prkachin and K. D. Craig, "Nonverbal communication in pain," *Pain: Clinical Updates*, vol. 17, no. 2, pp. 1–6, 2009.

[71] E. Martin, M. d. M. d'Autume, and C. Varray, "Audio denoising algorithm with block thresholding," *Image Processing*, pp. 2105–1232, 2012.

[72] V. Giordano, A. Luister, C. Reuter, I. Czedik-Eysenberg, D. Singer, D. Steyrl, E. Vettorazzi, and P. Deindl, "Audio feature analysis for acoustic pain detection in term newborns," *Neonatology*, vol. 119, no. 6, pp. 760–768, 2022.

[73] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," *IEEE transactions on multimedia*, vol. 13, no. 2, pp. 303–319, 2010.

[74] C.-H. Lee, J.-L. Shih, K.-M. Yu, and H.-S. Lin, "Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features," *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 670–682, 2009.

[75] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *NeurIPS*, vol. 33, pp. 12449–12460, 2020.

[76] A. S. Cowen and D. Keltner, "Mapping 24 emotions conveyed by brief human vocalization," *American Psychologist*, vol. 74, no. 6, pp. 698–712, 2019.

[77] B. Schuller and A. Batliner, *Computational paralinguistics: Emotion, affect and personality in speech and language processing*. Wiley, 2018.

[78] Y.-I. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 97–115, 2001.

[79] G. Muhammad, M. Alsulaiman, S. U. Amin, A. Ghoneim, and M. F. Alhamid, "A facial-expression monitoring system for improved healthcare in smart cities," *IEEE Access*, vol. 5, pp. 10871–10881, 2017.

[80] M. C. Caschera, F. Ferri, and P. Grifoni, "Multimodal emotion recognition from speech and text in videos," *Multimedia Tools and Applications*, vol. 81, no. 3, pp. 1–23, 2022.

[81] P. Rodriguez, Y. Wang, and X. Huang, "Hybrid cnn-transformer for robust pain recognition from facial expressions," *IEEE Transactions on Affective Computing*, vol. 13, no. 2, pp. 789–802, 2022.

[82] J. W. Pennebaker, M. R. Mehl, and K. G. Niederhoffer, "Psychological aspects of natural language use: Our words, our selves," *Annual Review of Psychology*, vol. 54, no. 1, pp. 547–577, 2003.

[83] N. Cummins, S. Scherer, J. Krajewski, S. Schnieder, J. Epps, and T. F. Quatieri, "A review of depression and suicide risk assessment using speech analysis," *Speech Communication*, vol. 71, pp. 10–49, 2015.

[84] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[85] M. Shutaywi and N. N. Kachouie, "Machine learning in pain assessment," *Artificial Intelligence in Medicine*, vol. 118, p. 102156, 2021.

[86] W. Chen, D. Liu, Z. Pei, and J. Wang, "Deep learning for pain expression recognition," *IEEE Transactions on Affective Computing*, vol. 13, no. 1, pp. 412–425, 2020.

[87] T. Hadjistavropoulos, K. Herr, D. C. Turk, P. G. Fine, R. H. Dworkin, R. Helme, K. Jackson, P. A. Parmelee, T. E. Rudy, B. Lynn Beattie, *et al.*, "Pain assessment in elderly adults with dementia," *The Lancet Neurology*, vol. 10, no. 7, pp. 592–603, 2011.

[88] R. H. Dworkin, D. C. Turk, S. Peirce-Sandner, L. B. Burke, J. T. Farrar, I. Gilron, M. P. Jensen, N. P. Katz, S. N. Raja, B. A. Rappaport, *et al.*, "Using patient-reported outcomes to improve pain practice," *Pain Medicine*, vol. 9, no. 6, pp. 685–695, 2008.

[89] M. Cohen, J. Quintner, D. Buchanan, M. Nielsen, and L. Guy, "Ethical issues in pain management," *Pain Medicine*, vol. 22, no. 3, pp. 501–529, 2021.

[90] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: A statistical framework," in *International Journal of Machine Learning and Cybernetics*, vol. 1, pp. 43–52, Springer, 2010.

[91] K. Spärck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[92] G. Peeters, "The temporal statistics of musical signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 5, pp. 1222–1235, 2011.

[93] A. Lerch, "An introduction to audio content analysis: Applications in signal processing and music informatics," *John Wiley & Sons*, 2012.

[94] S. A. Medjahed, "A comparative study of feature extraction methods in images classification," *International journal of image, graphics and signal processing*, vol. 7, no. 3, p. 16, 2015.

[95] T. Kobayashi, "Bfo meets hog: feature extraction based on histograms of oriented pdf gradients for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 747–754, 2013.

[96] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[97] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[98] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sensing*, vol. 7, no. 11, pp. 14680–14707, 2015.

[99] J. Wu, Y. Shi, S. Yan, and H.-m. Yan, "Global-local combined features to detect pain intensity from facial expression images with attention mechanism1," *Journal of Electronic Science and Technology*, p. 100260, 2024.

[100] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[101] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[102] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[103] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[104] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[105] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, "Tensorflow distributions," *arXiv preprint arXiv:1711.10604*, 2017.

[106] D. Knox, S. Beveridge, L. A. Mitchell, and R. A. MacDonald, "Acoustic analysis and mood classification of pain-relieving music," *The Journal of the Acoustical Society of America*, vol. 130, no. 3, pp. 1673–1682, 2011.

[107] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006. Includes discussion on confusion matrices.

[108] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 3rd ed., 2009. Covers time and space complexity analysis in detail.

[109] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *International Conference on Learning Representations (ICLR)*, 2016. Discusses model storage efficiency in deep learning.

[110] V. Mayer-Schönberger and K. Cukier, *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. Houghton Mifflin Harcourt, 2013. Discusses challenges in large-scale data storage.

[111] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. Theoretical treatment of time/space complexity.

[112] H. H. Abu-Saad, "Challenge of pain in the cognitively impaired.," *Lancet (London, England)*, vol. 356, no. 9245, pp. 1867–1868, 2000.

[113] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *ACL*, 2018.

[114] P. Malhotra, Y. Singh, P. Anand, D. K. Bangotra, P. K. Singh, and W.-C. Hong, "Internet of things: Evolution, concerns and security challenges," *Sensors*, vol. 21, no. 5, p. 1809, 2021.

[115] S. Tian, W. Yang, J. M. Le Grange, P. Wang, W. Huang, and Z. Ye, "Smart healthcare: making medical care more intelligent," *Global Health Journal*, vol. 3, no. 3, pp. 62–65, 2019.

[116] K. Herr, P. J. Coyne, T. Key, R. Manworren, M. McCaffery, S. Merkel, J. Pelosi-Kelly, and L. Wild, "Pain assessment in the nonverbal patient: position statement with clinical practice recommendations," *Pain Management Nursing*, vol. 7, no. 2, pp. 44–52, 2006.

[117] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of lstm and bilstm in forecasting time series," in *2019 IEEE International conference on big data (Big Data)*, pp. 3285–3292, IEEE, 2019.

[118] R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri, "Benchmarking aggression identification in social media," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pp. 1–11, 2018.

[119] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," in *fifth international AAAI conference on weblogs and social media*, 2011.

[120] M. Dadvar, D. Trieschnigg, R. Ordelman, and F. de Jong, "Improving cyberbullying detection with user context," in *European Conference on Information Retrieval*, pp. 693–696, Springer, 2013.

[121] M. Dadvar, D. Trieschnigg, and F. de Jong, "Experts and machines against bullies: A hybrid approach to detect cyberbullies," in *Canadian Conference on Artificial Intelligence*, pp. 275–281, Springer, 2014.

[122] C. Van Hee, E. Lefever, B. Verhoeven, J. Mennes, B. Desmet, G. De Pauw, W. Daelemans, and V. Hoste, "Detection and fine-grained classification of cyberbullying events," in *Proceedings of the international conference recent advances in natural language processing*, pp. 672–680, 2015.

[123] E. Cambria, P. Chandra, A. Sharma, and A. Hussain, "Do not feel the trolls," *ISWC, Shanghai*, 2010.

[124] S. Kumar, F. Spezzano, and V. Subrahmanian, "Accurately detecting trolls in slashdot zoo via decluttering," in *Proceedings of the 2014 IEEE/ACM international conference on advances in social networks analysis and mining*, pp. 188–195, IEEE Press, 2014.

[125] T. Mihaylov, G. Georgiev, and P. Nakov, "Finding opinion manipulation trolls in news community forums," in *Proceedings of the nineteenth conference on computational natural language learning*, pp. 310–314, 2015.

[126] L. G. Mojica, "Modeling trolling in social media conversations," *arXiv preprint arXiv:1612.05310*, 2016.

[127] E. Greevy and A. F. Smeaton, "Classifying racist texts using a support vector machine," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 468–469, ACM, 2004.

[128] P. Burnap and M. L. Williams, "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making," *Policy & Internet*, vol. 7, no. 2, pp. 223–242, 2015.

[129] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate speech detection with comment embeddings," in *Proceedings of the 24th international conference on world wide web*, pp. 29–30, ACM, 2015.

[130] N. D. Gitari, Z. Zuping, H. Damien, and J. Long, "A lexicon-based approach for hate speech detection," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 4, pp. 215–230, 2015.

[131] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 759–760, International World Wide Web Conferences Steering Committee, 2017.

[132] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, "Detecting offensive language in social media to protect adolescent online safety," in *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pp. 71–80, IEEE, 2012.

[133] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proceedings of the 25th international conference on world wide web*, pp. 145–153, International World Wide Web Conferences Steering Committee, 2016.

[134] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[135] K. Dashtipour, M. Gogate, E. Cambria, and A. Hussain, "A novel context-aware multimodal framework for persian sentiment analysis," *arXiv preprint arXiv:2103.02636*, 2021.

[136] R. A. Sagum, "An application of emotion detection in sentiment analysis on movie reviews," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 3, pp. 5468–5474, 2021.

[137] F. Rustam, M. Khalid, W. Aslam, V. Rupapara, A. Mehmood, and G. S. Choi, "A performance comparison of supervised machine learning models for covid-19 tweets sentiment analysis," *Plos one*, vol. 16, no. 2, p. e0245909, 2021.

[138] T. B. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *NeurIPS*, 2020.

[139] D. Zimbra, A. Abbasi, D. Zeng, and H. Chen, "The state-of-the-art in twitter sentiment analysis," *ACM Transactions on Management Information Systems*, vol. 9, no. 2, 2018.

[140] A. Conneau, K. Khandelwal, *et al.*, "Unsupervised cross-lingual representation learning at scale," 2020.

[141] F. Barbieri, J. Camacho-Collados, *et al.*, "Tweeteval: Unified benchmark for twitter sentiment analysis," 2020.

[142] Y. Cai, H. Cai, and X. Wan, "Multi-modal sarcasm detection in twitter," *ACL*, 2019.

[143] L. A. M. Bostan and R. Klinger, "An analysis of annotated corpora for emotion classification in text," 2018.

[144] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," *KDD*, 2016.

[145] S. L. Blodgett, S. Barocas, *et al.*, "Language (technology) is power: A critical survey of "bias" in nlp," *ACL*, 2020.

[146] A. Al-Hassan and H. Al-Dossari, "Pain sentiment detection in clinical narratives using hybrid deep learning models," *Journal of Biomedical Informatics*, vol. 99, p. 103291, 2019.

[147] S. Ji, X. Li, Z. Huang, and J. Wu, "Bert for pain intensity classification in textual patient reports," *IEEE Access*, vol. 8, pp. 107840–107849, 2020.

[148] S. M. Sarsam, H. Al-Samarraie, and A. I. Alzahrani, "A lexicon-based approach for pain sentiment analysis," *Health Informatics Journal*, vol. 26, no. 3, pp. 1921–1934, 2020.

[149] J. Gratch, R. Artstein, and G. Lucas, "Multimodal pain assessment: Integrating text and acoustic-prosodic features," *IEEE Transactions on Affective Computing*, 2021.

[150] E. Sharma, S. Ghosh, and M. Choudhury, "Unsupervised detection of pain-related tweets using topic modeling," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. CSCW1, pp. 1–26, 2021.

[151] P. López-Úbeda, M. C. Díaz-Galiano, and T. Martín-Noguerol, "Cross-lingual pain sentiment analysis using xlm-r: A case study in spanish," *Journal of Biomedical Semantics*, vol. 13, no. 1, pp. 1–12, 2022.

[152] S. Deerwester, A. Carcone, and S. Naar, "Bias in pain sentiment annotation: Ethical implications and mitigation strategies," *Journal of Medical Ethics*, vol. 48, no. 6, pp. 403–410, 2022.

[153] W. Peng, R. Zheng, and Y. Chen, "Explainable ai for pain sentiment analysis in mental health forums," *Artificial Intelligence in Medicine*, vol. 135, p. 102471, 2023.

[154] C. P. Cheng, T. Owusu, P. Shekane, and A. M. Patel, "Sentiment analysis of pain physician reviews on healthgrades: a physician review website," *Regional Anesthesia and Pain Medicine*, vol. 49, no. 9, pp. 656–660, 2024.

[155] D. A. P. Nunes *et al.*, "Computational analysis of the language of pain: A systematic review," *arXiv preprint arXiv:2404.16226*, 2024.

[156] I. Aggarwal, S. Joseph, N. Jaganathan, *et al.*, "Sentiment analysis in healthcare: A comparison of vader, bert, and flair nlp models on patient reviews of pain management physicians," *Cureus*, 2024. Early online publication.

[157] A. Ghosh, S. Umer, B. C. Dhara, D. K. Jain, R. K. Rout, and A. Hussain, "A novel pain sentiment detection system utilizing a paincapsule model and textual facial patterns," *Neurocomputing*, p. 130907, 2025.

[158] R. Fang, E. Hosseini, R. Zhang, C. Fang, S. Rafatirad, H. Homayoun, *et al.*, "Survey on pain detection using machine learning models: Narrative review," *JMIR AI*, vol. 4, no. 1, p. e53026, 2025.

[159] R. Fernandez-Rojas, C. Joseph, N. Hirachan, B. Seymour, and R. Goecke, "The ai4pain grand challenge 2025: Advancing pain assessment with multimodal physiological signals," in *Companion Proceedings of the 27th International Conference on Multimodal Interaction*, pp. 147–152, 2025.

[160] K. Raiyani, T. Gonçalves, P. Quaresma, and V. B. Nogueira, "Fully connected neural network with advance preprocessor to identify aggression over facebook

and twitter," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pp. 28–41, 2018.

[161] C. Boulis and M. Ostendorf, "Text classification by augmenting the bag-of-words representation with redundancy-compensated bigrams," in *Proc. of the International Workshop in Feature Selection in Data Mining*, pp. 9–16, Citeseer, 2005.

[162] Y. Bengio, P. Simard, P. Frasconi, *et al.*, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[163] N. S. Samghabadi, D. Mave, S. Kar, and T. Solorio, "Ritual-uh at trac 2018 shared task: Aggression identification," *arXiv preprint arXiv:1807.11712*, 2018.

[164] R. Kumar, G. Bhanodai, R. Pamula, and M. R. Chennuru, "Trac-1 shared task on aggression identification: Iit (ism)@ coling'18," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pp. 58–65, 2018.

[165] S. Modha, P. Majumder, and T. Mandl, "Filtering aggression from the multilingual social media feed," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pp. 199–207, 2018.

[166] C. Orasan, "Aggressive language identification using word embeddings and sentiment features," Association for Computational Linguistics, 2018.

[167] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the ACL*, pp. 90–94, 2012.

[168] S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti, "Identifying helpful reviews based on customer voting," in *Proceedings of the COLING/ACL 2006*, pp. 597–604, 2006.

[169] Y. Oshrat, A. Bloch, A. Lerner, A. Cohen, M. Avigal, and G. Zeilig, "Speech prosody as a biosignal for physical pain detection," in *Conf Proc 8th Speech Prosody*, pp. 420–24, 2016.

[170] Z. Ren, N. Cummins, J. Han, S. Schnieder, J. Krajewski, and B. Schuller, "Evaluation of the pain level from speech: Introducing a novel pain database and benchmarks," in *Speech Communication; 13th ITG-Symposium*, pp. 1–5, VDE, 2018.

[171] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: audio, visual and spontaneous expressions," in *Proceedings of the 9th international conference on Multimodal interfaces*, pp. 126–133, 2007.

[172] H.-T. Hong, J.-L. Li, C.-M. Chang, and C.-C. Lee, "Improving automatic pain level recognition using pain site as an auxiliary task," in *2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pp. 284–289, IEEE, 2019.

[173] A. C. Williams and K. N. Stevens, "Vocal correlates of pain," *Pain*, 2002.

[174] B. Schuller, S. Steidl, and A. Batliner, "Acoustic pain recognition," in *INTERSPEECH*, 2013.

[175] M. Schmitt, F. Ringeval, and B. Schuller, "Computational paralinguistics challenge," in *INTERSPEECH*, 2016.

[176] J. Han, Z. Zhang, F. Ringeval, and B. Schuller, "Deep learning for pain vocalization detection," *IEEE Transactions on Affective Computing*, 2019.

[177] I. Lefter, L. Rothkrantz, and D. van Leeuwen, "Audio-based pain detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[178] J. Gratch, R. Artstein, G. Lucas, and G. Stratou, "Cross-cultural pain vocalizations," *Speech Communication*, 2019.

[179] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Federated learning for pain recognition," *Medical Image Analysis*, 2021.

[180] N. N. Vempala and F. A. Russo, "Discriminating pain from speech dynamics," *The Journal of the Acoustical Society of America*, 2017.

[181] J. A. Sturgeon, B. D. Darnall, M.-C. Kao, and S. C. Mackey, "Voice analysis for pain monitoring," *Pain Reports*, 2018.

[182] S. Roy and J. Gratch, "Acoustic markers of pain types," *Journal of Voice*, 2020.

[183] R. Cao, Y. Zhang, and B. Schuller, "Generative data augmentation for pain detection," *IEEE Transactions on Affective Computing*, 2021.

[184] A. Baird and B. Schuller, "Multilingual pain recognition," in *INTERSPEECH*, 2022.

[185] Z. Hammal and J. F. Cohn, "Automatic pain detection from speech," *IEEE Transactions on Affective Computing*, 2015.

[186] F. Ringeval, B. Schuller, M. Valstar, and J. Gratch, "Prediction of pain from voice," *IEEE Transactions on Affective Computing*, 2017.

[187] S. Amiriparian, M. Schmitt, and B. Schuller, "Computational paralinguistics challenge," in *INTERSPEECH*, 2020.

[188] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey of voice-based pain detection," *IEEE Transactions on Affective Computing*, 2017.

[189] K. Qian, Z. Zhang, F. Ringeval, and B. Schuller, "Semi-supervised pain recognition," *IEEE Transactions on Affective Computing*, 2021.

[190] P. Lopez-Otero, L. Docio-Fernandez, and C. Garcia-Mateo, "Analysis of medication effects on pain voice," *Speech Communication*, 2018.

[191] J. Zhao and B. Schuller, "Neural architecture search for pain detection," *IEEE Transactions on Affective Computing*, 2022.

[192] B. Schuller, S. Amiriparian, and M. Schmitt, "Interspeech paralinguistics challenge," in *INTERSPEECH*, 2021.

[193] L. Yao and B. Schuller, "Contrastive learning for pain recognition," *IEEE Transactions on Affective Computing*, 2022.

[194] S. Vhaduri and C. Poellabauer, "Automatic pain detection from formants," *IEEE Journal of Biomedical and Health Informatics*, 2019.

[195] J. Xu, B. S. Glicksberg, J. Bian, and F. Wang, "Federated learning for pain recognition," *Nature Digital Medicine*, 2023.

[196] G. Rizos and B. Schuller, "Ieee signal processing cup 2022: Pain recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[197] A. Saeed, D. Grangier, and N. Zeghidour, "Self-supervised learning for pain detection," *IEEE Transactions on Affective Computing*, 2021.

[198] R. Milner and B. Schuller, "Impact of microphone type on pain classification," *Speech Communication*, 2022.

[199] P. Tzirakis, J. Zhang, and B. Schuller, "End-to-end pain detection," *IEEE Transactions on Affective Computing*, 2018.

[200] M. Schmitt, S. Amiriparian, and B. Schuller, "Computational paralinguistics challenge 2023," in *INTERSPEECH*, 2023.

[201] Y. Liang and B. Schuller, "Transformer-based pain recognition from voice," *IEEE Transactions on Affective Computing*, 2023.

[202] W. Chen and B. Schuller, "Explainable ai for pain vocalization analysis," *Pattern Recognition*, 2023.

[203] H. Park and B. Schuller, "Demographic effects in pain voice recognition," *Speech Communication*, 2022.

[204] L. Wang and B. Schuller, "Spectral-temporal features for pain detection," *IEEE Transactions on Affective Computing*, 2023.

[205] S. Amiriparian, M. Schmitt, and B. Schuller, "Interspeech 2022 computational paralinguistics challenge," in *INTERSPEECH*, 2022.

[206] Y. Zhang and B. Schuller, "Few-shot learning for pain recognition," *IEEE Transactions on Affective Computing*, 2023.

[207] M. Johnson and B. Schuller, "Clinical condition-specific pain vocalizations," *Journal of Medical Systems*, 2023.

[208] J. Smith and B. Schuller, "Edge computing for real-time pain monitoring," *IEEE Internet of Things Journal*, 2023.

[209] R. Anderson and B. Schuller, "Postoperative pain monitoring via voice analysis," *Anesthesia & Analgesia*, 2023.

[210] S. Lee and B. Schuller, "Cross-modal pain recognition," *IEEE Transactions on Affective Computing*, 2023.

[211] B. Schuller and J. Gratch, "Special issue on computational pain assessment," *IEEE Transactions on Affective Computing*, 2023.

[212] D. Brown and B. Schuller, "Denoising algorithms for pain voice analysis," *Speech Communication*, 2023.

[213] S. Taylor and B. Schuller, "Psychological basis of pain vocalizations," *Journal of Nonverbal Behavior*, 2023.

[214] E. Wilson and B. Schuller, "Federated transfer learning for pain recognition," *IEEE Transactions on Affective Computing*, 2023.

[215] M. Garcia and B. Schuller, "Cross-cultural pain expression analysis," *Journal of Voice*, 2023.

[216] K. Roberts and B. Schuller, "Temporal alignment of pain vocalizations," *IEEE Transactions on Affective Computing*, 2023.

[217] L. White and B. Schuller, "Editorial: Pain and voice special issue," *Journal of Voice*, 2023.

[218] M. Harris and B. Schuller, "Multitask learning for pain analysis," *IEEE Transactions on Affective Computing*, 2023.

[219] O. Martin and B. Schuller, "Voice assistants for pain self-reporting," *Journal of Medical Internet Research*, 2023.

[220] R. Thompson and B. Schuller, "Graph networks for pain voice analysis," *IEEE Transactions on Affective Computing*, 2023.

[221] M. Schmitt and B. Schuller, "Paralinguistics in medicine challenge," in *INTERSPEECH*, 2023.

[222] B. Adams and B. Schuller, "Active learning for pain annotation," *IEEE Transactions on Affective Computing*, 2023.

[223] J. Green and B. Schuller, "Physiological correlates of pain voice," *Psychophysiology*, 2023.

[224] R. Clark and B. Schuller, "Privacy-aware pain monitoring," *IEEE Transactions on Affective Computing*, 2023.

[225] D. Miller and B. Schuller, "Vocal biomarkers in healthcare," *IEEE Journal of Biomedical and Health Informatics*, 2023.

[226] S. Borna, C. R. Haider, K. C. Maita, R. A. Torres, F. R. Avila, J. P. Garcia, G. D. De Sario Velasquez, C. J. McLeod, C. J. Bruce, R. E. Carter, *et al.*, "A review of voice-based pain detection in adults using artificial intelligence," *Bioengineering*, vol. 10, no. 4, p. 500, 2023.

[227] T.-Q. Dao, E. Schneiders, J. Williams, J. R. Bautista, T. Seabrooke, G. Vigneswaran, R. Kolpekwar, R. Vashistha, and A. Farahi, "Tame pain: Trustworthy assessment of pain from speech and audio for the empowerment of patients," 2025.

[228] A. Lu, M. Kajol, W. Lu, and D. Sullivan, "Poster: Painnova: Privacy-aware voice-based pain-level detection," in *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pp. 4761–4763, 2025.

[229] A. Ghosh, S. Umer, B. C. Dhara, and G. M. N. Ali, "A multimodal pain sentiment analysis system using ensembled deep learning approaches for iot-enabled healthcare framework," *Sensors*, vol. 25, no. 4, p. 1223, 2025.

[230] S. Nawab, T. Quatieri, and J. Lim, "Signal reconstruction from short-time fourier transform magnitude," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 4, pp. 986–998, 1983.

[231] N. Holz, P. Larrouy-Maestri, and D. Poeppel, "The variably intense vocalizations of affect and emotion (vivae) corpus prompts new perspective on non-speech perception.," *Emotion*, vol. 22, no. 1, p. 213, 2022.

[232] B. Martinez and M. Valstar, "Vivae: A variational autoencoder for voice actor encoding," *arXiv preprint arXiv:2006.13886*, 2020.

[233] M. Valstar, B. Schuller, K. Smith, F. Eyben, B. Jiang, S. Bilakhia, S. Schnieder, R. Cowie, and M. Pantic, "Avec 2016: Depression, mood, and emotion recognition workshop and challenge," in *Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge*, pp. 3–10, 2016.

[234] M. Pantic and I. Patras, "Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 2, pp. 433–449, 2006.

[235] S. R. Livingstone and F. A. Russo, "The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english," *PLOS ONE*, vol. 13, no. 5, p. e0196391, 2018.

[236] B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi, *et al.*, "The interspeech 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism," *Proceedings of Interspeech*, pp. 148–152, 2013.

[237] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network," *2016 IEEE ICASSP*, pp. 5200–5204, 2016.

[238] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2010.

[239] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. Matthews, "Painful data: The unbc-mcmaster shoulder pain expression archive database," in *Face and Gesture 2011*, pp. 57–64, IEEE, 2011.

[240] S. Walter, S. Gruss, H. Ehleiter, J. Tan, H. C. Traue, P. Werner, A. Al-Hamadi, S. Crawcour, A. O. Andrade, and G. M. da Silva, "The biovid heat pain database data for the advancement and systematic validation of an automated pain recognition system," in *2013 IEEE international conference on cybernetics (CYBCO)*, pp. 128–131, IEEE, 2013.

[241] A. B. Ashraf, S. Lucey, J. F. Cohn, T. Chen, Z. Ambadar, K. M. Prkachin, and P. E. Solomon, "The painful face–pain expression recognition using active appearance models," *Image and vision computing*, vol. 27, no. 12, pp. 1788–1796, 2009.

[242] P. Lucey, J. Cohn, J. Howlett, S. Lucey, and S. Sridharan, "Recognizing emotion with head pose variation: Identifying pain segments in video," *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, vol. 41, no. 3, pp. 664–674, 2011.

[243] G. Littlewort-Ford, M. S. Bartlett, and J. R. Movellan, "Are your eyes smiling? detecting genuine smiles with support vector machines and gabor wavelets," in *In Proceedings of the 8th Joint Symposium on Neural Computation*, Citeseer, 2001.

[244] S. Umer, B. C. Dhara, and B. Chanda, "Face recognition using fusion of feature learning techniques," *Measurement*, vol. 146, pp. 43–54, 2019.

[245] C. Bisogni, A. Castiglione, S. Hossain, F. Narducci, and S. Umer, "Impact of deep learning approaches on facial expression recognition in healthcare industries," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5619–5627, 2022.

[246] P. Cunningham and S. J. Delany, "k-nearest neighbour classifiers-a tutorial," *ACM computing surveys (CSUR)*, vol. 54, no. 6, pp. 1–25, 2021.

[247] I. Steinwart and A. Christmann, *Support vector machines*. Springer Science & Business Media, 2008.

[248] M. P. LaValley, "Logistic regression," *Circulation*, vol. 117, no. 18, pp. 2395–2399, 2008.

[249] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708, 2014.

[250] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2892–2900, 2015.

[251] A. Gupta, S. Lee, and W. Zhang, "Edge-ai for real-time pain assessment: A lightweight deep learning approach," in *Proceedings of the ACM Conference on Health, Inference, and Learning*, pp. 112–125, ACM, 2023.

[252] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[253] Z.-x. Liu, D.-g. Zhang, G.-z. Luo, M. Lian, and B. Liu, "A new method of emotional analysis based on cnn–bilstm hybrid neural network," *Cluster Computing*, vol. 23, no. 4, pp. 2901–2913, 2020.

[254] Y. Zhang, Q. Yang, and X. Huang, "Multimodal fusion for pain recognition," *IEEE Transactions on Affective Computing*, 2021.

[255] J.-B. Grill, F. Strub, F. Altché, *et al.*, "Bootstrap your own latent," in *NeurIPS*, 2020.

[256] P. Lucey, J. F. Cohn, and K. M. Prkachin, "Painful data: The unbc-mcmaster shoulder pain archive," in *FG*, 2011.

[257] P. Werner, A. Al-Hamadi, and R. Niese, "Biovid heat pain dataset," *Scientific Data*, 2019.

[258] M. Kächele, P. Thiam, and M. Amirian, "X-ite pain dataset," *IEEE Transactions on Affective Computing*, 2020.

[259] Z. Liu, Y. Lin, Y. Cao, *et al.*, "Swin transformer: Hierarchical vision transformer," in *ICCV*, 2021.

[260] S. Yan, Y. Xiong, and D. Lin, "Dynamic graph networks for pain au modeling," in *CVPR*, 2021.

[261] T. Chen, S. Kornblith, and M. Norouzi, "A simple framework for contrastive learning," *ICML*, 2020.

[262] A. Radford, J. W. Kim, and C. Hallacy, "Learning transferable visual models from natural language," *ICML*, 2021.

[263] B. Zhou, A. Khosla, and A. Lapedriza, "Weakly supervised pain localization," in *ICCV*, 2021.

[264] B. Settles, "Active learning literature survey," *Machine Learning*, 2009.

[265] M. Valstar, J. Cohn, and M. Pantic, "Emopain challenge 2023," in *FG*, 2023.

[266] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning," *ICML*, 2017.

[267] R. Selvaraju, M. Cogswell, and A. Das, "Grad-cam: Visual explanations from deep networks," in *ICCV*, 2017.

[268] Q. Yang, Y. Liu, and T. Chen, "Federated learning," *Machine Learning*, 2019.

[269] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NeurIPS*, 2020.

[270] B. Zoph, V. Vasudevan, and J. Shlens, "Learning transferable architectures for scalable image recognition," in *CVPR*, 2018.

[271] X. Liu, H. Wang, and H. Tang, "Cross-dataset pain analysis," *IEEE Transactions on Affective Computing*, 2021.

[272] D. Kollias and S. Zafeiriou, "Aff-wild2 pain dataset," in *CVPR*, 2023.

[273] A. Ruiz, F. Martínez, and I. Valera, "Multitask pain recognition," *IEEE Transactions on Affective Computing*, 2021.

[274] L. Wang, Y. Xiong, and Z. Wang, "Spatiotemporal attention for pain videos," in *NeurIPS*, 2021.

[275] O. Vinyals, C. Blundell, and T. Lillicrap, "Matching networks for one-shot learning," *NeurIPS*, 2016.

[276] G. Hinton, O. Vinyals, and J. Dean, "Distilling knowledge in a neural network," *NeurIPS*, 2015.

[277] S. M. Mavadati and M. H. Mahoor, "Disfa+: Extended disfa for pain analysis," in *FG*, 2022.

[278] K. Sohn, D. Berthelot, and N. Carlini, "Fixmatch: Simplifying semi-supervised learning," *NeurIPS*, 2020.

[279] J. Lee, D. Kim, and S. Park, "Hybrid cnn-rnn for pain recognition," *IEEE Transactions on Affective Computing*, 2021.

[280] Y. Ganin, E. Ustinova, and H. Ajakan, "Domain-adversarial training of neural networks," *JMLR*, 2016.

[281] N. Jakob, Y. Zhang, and D. Chen, "Quantization for edge devices," *IEEE IoT*, 2022.

[282] J. Pearl, *Causality*. Cambridge, 2009.

[283] P. Barros, D. Jirak, and S. Wermter, "Omg-pain challenge 2023," in *ACII*, 2023.

[284] H. Akbari, L. Yuan, and R. Qian, "Multimodal transformers for pain recognition," in *ICASSP*, 2021.

[285] A. Dosovitskiy, L. Beyer, and A. Kolesnikov, "An image is worth 16x16 words: Transformers for image recognition," *ICLR*, 2020.

[286] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *NeurIPS*, 2018.

[287] Y. Wang, A. Gupta, and S. Gupta, "Dynamic graph networks for pain progression," in *CVPR*, 2022.

[288] D. Wang, E. Shelhamer, and S. Liu, "Tent: Fully test-time adaptation," *ICLR*, 2020.

[289] M. Arjovsky, L. Bottou, and I. Gulrajani, "Invariant risk minimization," *ICLR*, 2019.

[290] R. T. Q. Chen, Y. Rubanova, and J. Bettencourt, "Neural ordinary differential equations," *NeurIPS*, 2018.

[291] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *NeurIPS*, 2017.

[292] C. Saharia, W. Chan, and S. Saxena, "Photorealistic text-to-image diffusion models," in *CVPR*, 2022.

[293] X. Zhang, X. Zhou, and M. Lin, "Transformer-cnn hybrids for pain recognition," *IEEE Transactions on Affective Computing*, 2022.

[294] Z. Yue, Q. Sun, and X. Wang, "Disentangled causal learning for pain recognition," in *CVPR*, 2022.

[295] A. Santoro, S. Bartunov, and M. Botvinick, "Meta-learning with memory-augmented neural networks," *ICLR*, 2017.

[296] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 2879–2886, IEEE, 2012.

[297] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, no. 1, p. 7068349, 2018.

[298] S. D. Sharghi and F. A. Kamangar, "Geometric feature-based matching in stereo images," in *1999 Information, Decision and Control. Data and Information Fusion Symposium, Signal Processing and Communications Symposium and Decision and Control Symposium. Proceedings (Cat. No. 99EX251)*, pp. 65–70, IEEE, 1999.

[299] H. Baali, A. Khorshidtalab, M. Mesbah, and M. J. Salami, "A transform-based feature extraction approach for motor imagery tasks classification," *IEEE journal of translational engineering in health and medicine*, vol. 3, pp. 1–8, 2015.

[300] S. Umer, B. C. Dhara, and B. Chanda, "An iris recognition system based on analysis of textural edgeness descriptors," *IETE Technical Review*, vol. 35, no. 2, pp. 145–156, 2018.

[301] S. Manocha and M. A. Girolami, "An empirical analysis of the probabilistic k-nearest neighbour classifier," *Pattern Recognition Letters*, vol. 28, no. 13, pp. 1818–1824, 2007.

[302] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[303] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[304] D. McNeely-White, J. R. Beveridge, and B. A. Draper, "Inception and resnet features are (almost) equivalent," *Cognitive Systems Research*, vol. 59, pp. 312–318, 2020.

[305] X. Xiang, F. Wang, Y. Tan, and A. L. Yuille, "Imbalanced regression for intensity series of pain expression from videos by regularizing spatio-temporal face nets," *Pattern Recognition Letters*, vol. 163, pp. 152–158, 2022.

[306] D. Bourou, A. Pampouchidou, M. Tsiknakis, K. Marias, and P. Simos, "Video-based pain level assessment: Feature selection and inter-subject variability modeling," in *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pp. 1–6, IEEE, 2018.

[307] P. Ekman and W. V. Friesen, *Facial action coding system: A technique for the measurement of facial movement*. Consulting Psychologists Press, 1978.

[308] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 39–58, 2009.

[309] S. E. Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal, "Emonets: Multimodal deep learning approaches for emotion recognition in video," in *Proceedings of the ACM International Conference on Multimodal Interaction*, pp. 461–466, 2015.

[310] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "Affectnet: A database for facial expression, valence, and arousal computing in the wild," *IEEE Transactions on Affective Computing*, vol. 10, no. 1, pp. 18–31, 2016.

[311] O. Arriaga, M. Valdenegro-Toro, and P. Plöger, "Real-time convolutional neural networks for emotion and gender classification," *Neurocomputing*, vol. 388, pp. 212–221, 2019.

[312] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *IEEE Computer Vision and Pattern Recognition Workshops*, pp. 94–101, 2010.

[313] A. Dhall, O. Ramana Murthy, R. Goecke, J. Joshi, and T. Gedeon, "Collecting large, richly annotated facial-expression databases from movies," *IEEE Multimedia*, vol. 19, no. 3, pp. 34–41, 2015.

[314] S. Zhao, H. Cai, H. Zhang, L. Chen, and G. Liu, "Video-based emotion recognition using deep learning approaches," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, no. 3, pp. 1258–1269, 2021.

[315] G. Bargshady, X. Zhou, R. C. Deo, J. Soar, F. Whittaker, and H. Wang, "Enhanced deep learning algorithm development to detect pain intensity from facial expression images," *Expert Systems with Applications*, vol. 149, p. 113305, 2020.

[316] R. M. Al-Eidan, H. Al-Khalifa, and A. Al-Salman, "Deep-learning-based models for pain recognition: A systematic review," *Applied Sciences*, vol. 10, no. 17, p. 5984, 2020.

[317] P. Gouverneur, F. Li, W. M. Adamczyk, T. M. Szikszay, K. Luedtke, and M. Grzegorzek, "Comparison of feature extraction methods for physiological signals for heat-based pain recognition," *Sensors*, vol. 21, no. 14, p. 4838, 2021.

[318] P. Thiam, P. Bellmann, H. A. Kestler, and F. Schwenker, "Exploring deep physiological models for nociceptive pain recognition," *Sensors*, vol. 19, no. 20, p. 4503, 2019.

[319] E. Othman, P. Werner, F. Saxen, A. Al-Hamadi, and S. Walter, "Cross-database evaluation of pain recognition from facial video," in *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pp. 181–186, IEEE, 2019.

[320] K. Pikulkaew, E. Boonchieng, W. Boonchieng, and V. Chouvatut, "Pain detection using deep learning with evaluation system," in *Proceedings of Fifth International Congress on Information and Communication Technology*, pp. 426–435, Springer, 2021.

[321] L. Ismail and M. D. Waseem, "Towards a deep learning pain-level detection deployment at uae for patient-centric-pain management and diagnosis support: Framework and performance evaluation," *Procedia Computer Science*, vol. 220, pp. 339–347, 2023.

[322] E. Othman, P. Werner, F. Saxen, A. Al-Hamadi, S. Gruss, and S. Walter, "Classification networks for continuous automatic pain intensity monitoring in video using facial expression on the x-ite pain database," *Journal of Visual Communication and Image Representation*, vol. 91, p. 103743, 2023.

[323] M. A. Ullah, S. Marium, M. K. Siddiquee, *et al.*, "Multimodal sentiment analysis from social media," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 12, pp. 1–8, 2017.

[324] M. Soleymani, D. Garcia, B. Jou, *et al.*, "A survey of multimodal sentiment analysis," *Image and Vision Computing*, vol. 65, pp. 3–14, 2017.

[325] K. S. Rao and S. G. Koolagudi, "Multimodal emotion recognition using lstm and audio-visual features," *IEEE Transactions on Affective Computing*, vol. 13, no. 2, pp. 678–691, 2021.

[326] S. A. Abdu, A. A. El-Latif, *et al.*, "Deep learning approaches for multimodal video sentiment analysis," *Neural Computing and Applications*, vol. 33, no. 15, pp. 1–18, 2021.

[327] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?," in *ICCV*, 2021.

[328] K. M. Prkachin and P. E. Solomon, "Automated measurement of pain from facial expressions," *IEEE Transactions on Affective Computing*, 2012.

[329] M. Valstar, T. Almaev, and J. M. Girard, "Facial expression recognition under partial occlusion," in *FG*, 2016.

[330] A. Diba, V. Sharma, and L. Van Gool, "Temporal linear encoding networks for weakly supervised action localization," in *ICCV*, 2017.

[331] A. Arnab, M. Dehghani, and G. Heigold, "Vivit: A video vision transformer," *ICLR*, 2021.

[332] G. Bargshady, C. Joseph, N. Hirachan, R. Goecke, and R. F. Rojas, "Acute pain recognition from facial expression videos using vision transformers," in *2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1–4, IEEE, 2024.

[333] E. Holden *et al.*, "From facial expressions to algorithms: A narrative review of animal pain assessment using artificial intelligence," *Frontiers in Veterinary Science*, vol. 11, p. 1436795, 2024.

[334] E. Holden *et al.*, "Extensions of computer-vision-based facial pain detection for postoperative and clinical monitoring." Discussed in narrative review, 2024.

[335] C. W. Tan, T. Du, J. C. Teo, D. X. H. Chan, W. M. Kong, and B. L. Sng, "Automated pain detection using facial expression in adult patients with a customized spatial temporal attention long short-term memory (sta-lstm) network," *Scientific Reports*, vol. 15, no. 1, p. 13429, 2025.

[336] J.-T. Zhang, X.-Y. Hu, W. Duan, M.-H. Ji, and J.-J. Yang, "Application of deep learning-based facial pain recognition model for postoperative pain assessment," *Journal of Clinical Anesthesia*, vol. 105, p. 111898, 2025.

[337] A. Saxena, "Convolutional neural networks: an illustration in tensorflow," *XRDS: Crossroads, The ACM Magazine for Students*, vol. 22, no. 4, pp. 56–58, 2016.

[338] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," pp. 248–255, 2009.

[339] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[340] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[341] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.

[342] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 10, pp. 4291–4308, 2020.

[343] K. Cho, A. Courville, and Y. Bengio, "Describing multimedia content using attention-based encoder-decoder networks," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1875–1886, 2015.

[344] F. Wang and D. M. Tax, "Survey on the attention based rnn model and its applications in computer vision," *arXiv preprint arXiv:1601.06823*, 2016.

[345] R. Zhi, C. Zhou, J. Yu, T. Li, and G. Zamzmi, "Multimodal-based stream integrated neural networks for pain assessment," *IEICE TRANSACTIONS on Information and Systems*, vol. 104, no. 12, pp. 2184–2194, 2021.

[346] M. A. Haque, R. B. Bautista, F. Noroozi, K. Kulkarni, C. B. Laursen, R. Irani, M. Bellantonio, S. Escalera, G. Anbarjafari, K. Nasrollahi, *et al.*, "Deep multimodal pain recognition: a database and comparison of spatio-temporal visual modalities," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 250–257, IEEE, 2018.

[347] U. Gaur and et al., "Multimodal pain recognition," *IEEE Transactions on Affective Computing*, 2021.

[348] P. Werner and et al., "Pain assessment," *Pain*, 2014.

[349] C. G. Snoek, M. Worring, and A. W. Smeulders, "Early versus late fusion in semantic video analysis," in *Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 399–402, 2005.

[350] D. Ramachandram and G. W. Taylor, "Deep multimodal learning: A survey on recent advances and trends," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 96–108, 2017.

[351] Z. Zhang, J. Han, and C. Deng, "Multimodal information fusion for affective computing: A review," *Information Fusion*, vol. 57, pp. 115–129, 2020.

[352] T. Baltrusaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2019.

[353] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multimedia systems*, vol. 16, no. 6, pp. 345–379, 2010.

[354] L. Kessous, G. Castellano, and G. Caridakis, "Multimodal emotion recognition in speech-based interaction using facial expression, body gesture and acoustic analysis," *Journal on Multimodal User Interfaces*, vol. 3, no. 1-2, pp. 33–48, 2010.

[355] M. Arif-Rahu and M. J. Grap, "Facial expression and pain in the critically ill non-communicative patient: state of science review," *Intensive and critical care nursing*, vol. 26, no. 6, pp. 343–352, 2010.

[356] P. Werner, D. Lopez-Martinez, S. Walter, A. Al-Hamadi, S. Gruss, and R. Picard, "Automatic recognition methods supporting pain assessment: A survey," *IEEE Transactions on Affective Computing*, 2019.

[357] K. A. Puntillo, A. B. Morris, C. L. Thompson, J. Stanik-Hutt, C. A. White, and L. R. Wild, "Pain behaviors observed during six common procedures: results from thunder project ii," *Critical care medicine*, vol. 32, no. 2, pp. 421–427, 2004.

[358] A. d. Williams, "Facial expression of pain: an evolutionary account.," *Behav Brain Sci*, vol. 25, no. 4, pp. 439–455, 2002.

[359] A. Twycross, T. Voepel-Lewis, C. Vincent, L. S. Franck, and C. L. von Baeyer, "A debate on the proposition that self-report is the gold standard in assessment

of pediatric pain intensity," *The Clinical journal of pain*, vol. 31, no. 8, pp. 707–712, 2015.

[360] B. McGuire, P. Daly, and F. Smyth, "Chronic pain in people with an intellectual disability: under-recognised and under-treated?," *Journal of Intellectual Disability Research*, vol. 54, no. 3, pp. 240–245, 2010.

[361] P. L. Manfredi, B. Breuer, D. E. Meier, and L. Libow, "Pain assessment in elderly patients with severe dementia," *Journal of Pain and Symptom Management*, vol. 25, no. 1, pp. 48–52, 2003.

[362] A. C. d. C. Williams and K. D. Craig, "Pain expression," *Nature Reviews Neurology*, 2016.

[363] X. Liu and et al., "Multimodal healthcare applications," *Journal of Biomedical Informatics*, 2020.

[364] S. Poria and et al., "Multimodal fusion techniques," *Information Fusion*, 2017.

[365] S. Chen and et al., "Bias reduction in pain assessment," *JMIR Medical Informatics*, 2019.

[366] S. Gruss, S. Walter, and H. Ehleiter, "Automatic pain assessment with multimodal data," in *International Conference on Affective Computing and Intelligent Interaction*, pp. 562–568, 2019.

[367] P. Thiam, H. A. Kestler, and F. Schwenker, "Multimodal deep learning for pain intensity classification," *Pattern Recognition Letters*, vol. 128, pp. 1–7, 2019.

[368] H. Lu, Q. Zhang, and L. Chen, "Wearable-based multimodal pain monitoring for chronic conditions," *npj Digital Medicine*, vol. 5, no. 1, pp. 1–10, 2022.

[369] P. Schmidt and S. Tschiatschek, "Addressing bias in multimodal pain datasets," *ACM Transactions on Computing for Healthcare*, vol. 1, no. 4, pp. 1–22, 2020.

[370] R. Li, Y. Liu, and T. Zhang, "Privacy-preserving multimodal healthcare analytics," *Nature Digital Health*, vol. 1, pp. 1–12, 2023.

[371] J. Wang and M. Zheng, "Edge computing for real-time pain recognition," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 1–12, 2023.

[372] P. Rajpurkar and E. Oren, "Ai in clinical decision support for pain management," *The Lancet Digital Health*, vol. 4, no. 2, pp. e76–e77, 2022.

[373] A. Johnson and T. Pollard, "Next-generation ehrs with multimodal data integration," *Journal of Medical Systems*, vol. 35, no. 5, pp. 1585–1594, 2011.

[374] A. C. d. C. Williams, "Facial expression of pain: An evolutionary account," *Behavioral and Brain Sciences*, vol. 25, no. 4, pp. 439–455, 2007.

[375] D. Liu, W. Chen, Z. Pei, and J. Wang, "Multimodal data fusion for pain intensity assessment," *IEEE Transactions on Affective Computing*, vol. 13, no. 1, pp. 379–392, 2020.

[376] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea, "Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2236–2246, 2019.

[377] B. Schuller, M. Valstar, J. Gratch, R. Cowie, and M. Pantic, "Multimodal machine learning for pain intensity assessment," *IEEE Transactions on Affective Computing*, vol. 10, no. 4, pp. 499–510, 2018.

[378] S. Lautenbacher and M. Kunz, "Pain assessment in humans," *Neuroscience & Biobehavioral Reviews*, vol. 131, pp. 462–473, 2021.

[379] A. Zadeh, P. Liang, S. Poria, E. Cambria, and L.-P. Morency, "Memory fusion network for multi-view sequential learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[380] P. P. Liang, A. Zadeh, Y. C. Liu, and L.-P. Morency, "Multimodal transformer for unaligned multimodal language sequences," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 6558–6569, 2019.

[381] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2019.

[382] S. Malmasi and M. Zampieri, "Detecting hate speech in social media," in *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pp. 467–472, 2017.

[383] R. A. Khan, A. Meyer, H. Konik, and S. Bouakaz, "Pain detection through shape and appearance features," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, 2013.

[384] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a cpu and gpu math expression compiler," in *Proceedings of the Python for scientific computing conference (SciPy)*, vol. 4, pp. 1–7, Austin, TX, 2010.

[385] M. Liu, S. Li, S. Shan, and X. Chen, "Au-aware deep networks for facial expression recognition," in *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pp. 1–6, IEEE, 2013.

[386] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE transactions on neural networks*, vol. 8, no. 1, pp. 98–113, 1997.

[387] Y. Chen and Z. Zhang, "Research on text sentiment analysis based on cnns and svm," in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 2731–2734, IEEE, 2018.

[388] R. Kumar, A. K. Ojha, M. Zampieri, and S. Malmasi, "Proceedings of the first workshop on trolling, aggression and cyberbullying (trac-2018)," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, 2018.

[389] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *Proceedings of the NAACL student research workshop*, pp. 88–93, 2016.

[390] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[391] F. Del Vigna12, A. Cimino23, F. Dell'Orletta, M. Petrocchi, and M. Tesconi, "Hate me, hate me not: Hate speech detection on facebook," in *Proceedings of the First Italian Conference on Cybersecurity (ITASEC17)*, pp. 86–95, 2017.

[392] S. Malmasi and M. Zampieri, "Challenges in discriminating profanity from hate speech," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 30, no. 2, pp. 187–202, 2018.

[393] S. Poria, E. Cambria, and A. Gelbukh, "Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 2539–2544, 2015.

[394] C. Langlet and C. Clavel, "Adapting sentiment analysis to face-to-face human-agent interactions: from the detection to the evaluation issues," in *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 14–20, IEEE, 2015.

[395] B. Schuller, "Recognizing affect from linguistic information in 3d continuous space," *IEEE Transactions on Affective computing*, vol. 2, no. 4, pp. 192–205, 2011.

[396] A. Calder, G. Rhodes, M. Johnson, and J. Haxby, *Oxford handbook of face perception*. Oxford University Press, 2011.

[397] J. W. Tanaka and I. Gordon, "Features, configuration, and holistic face processing," *The Oxford handbook of face perception*, pp. 177–194, 2011.

[398] R. Adolphs, "Perception and emotion: How we recognize facial expressions," *Current directions in psychological science*, vol. 15, no. 5, pp. 222–226, 2006.

[399] G. W. Cottrell, M. N. Dailey, C. Padgett, and R. Adolphs, "Is all face processing holistic? the view from ucsd," *Computational, geometric, and process perspectives on facial cognition*, pp. 347–396, 2001.

[400] D. Neth and A. M. Martinez, "A computational shape-based model of anger and sadness justifies a configural representation of faces," *Vision research*, vol. 50, no. 17, pp. 1693–1711, 2010.

[401] M. White, "Parts and wholes in expression recognition," *Cognition & Emotion*, vol. 14, no. 1, pp. 39–60, 2000.

[402] L. Zhang and G. W. Cottrell, "When holistic processing is not enough: Local features save the day," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 26, 2004.

[403] G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer methods for mathematical computations*, vol. 259. Prentice-Hall Englewood Cliffs, NJ, 1977.

[404] M. E. Stokes, C. S. Davis, and G. G. Koch, "Categorical data analysis using the sas system. sas institute," *Inc., Cary, NC*, pp. 34–35, 1995.

[405] V. N. Vapnik, "The nature of statistical learning," *Theory*, 1995.

[406] T. Evgeniou and M. Pontil, "Workshop on support vector machines: theory and applications," 2001.

[407] X. Xia, C. Xu, and B. Nan, "Inception-v3 for flower classification," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, pp. 783–787, IEEE, 2017.

[408] P. Goldsborough, "A tour of tensorflow," *arXiv preprint arXiv:1610.01178*, 2016.

[409] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei, "The unreasonable effectiveness of noisy data for fine-grained recognition," in *European Conference on Computer Vision*, pp. 301–320, Springer, 2016.

[410] D. Kirange, R. R. Deshmukh, and M. Kirange, "Aspect based sentiment analysis semeval-2014 task 4," *Asian J. Comput. Sci.{&} Inf. Technol*, vol. 4, no. 8, pp. 72–75, 2014.

[411] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowledge-Based Systems*, vol. 108, pp. 42–49, 2016.

[412] J. Yuan, S. Mcdonough, Q. You, and J. Luo, "Sentribute: image sentiment analysis from a mid-level perspective," in *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, pp. 1–8, 2013.

[413] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2472–2481, 2018.

[414] Z. Ding, M. Zhu, V. W. Tam, G. Yi, and C. N. Tran, "A system dynamics-based environmental benefit assessment model of construction waste reduction management at the design and construction stages," *Journal of cleaner production*, vol. 176, pp. 676–692, 2018.

[415] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-cnn: Octree-based convolutional neural networks for 3d shape analysis," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–11, 2017.

[416] K. Okada, J. Steffens, T. Maurer, H. Hong, E. Elagin, H. Neven, and C. von der Malsburg, "The bochum/usc face recognition system and how it fared in the feret phase iii test," in *Face Recognition*, pp. 186–205, Springer, 1998.

[417] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[418] K. Mistry, L. Zhang, S. C. Neoh, C. P. Lim, and B. Fielding, "A micro-ga embedded pso feature selection approach to intelligent facial emotion recognition," *IEEE transactions on cybernetics*, vol. 47, no. 6, pp. 1496–1509, 2016.

[419] A. Ghosh, S. Umer, M. K. Khan, R. K. Rout, and B. C. Dhara, "Smart sentiment analysis system for pain detection using cutting edge techniques in a smart healthcare framework," *Cluster Computing*, vol. 26, no. 1, pp. 119–135, 2023.

[420] S. Hossain, S. Umer, V. Asari, and R. K. Rout, "A unified framework of deep learning-based facial expression recognition system for diversified applications," *Applied Sciences*, vol. 11, no. 19, p. 9174, 2021.

[421] S. Umer, R. K. Rout, C. Pero, and M. Nappi, "Facial expression recognition with trade-offs between data augmentation and deep learning features," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–15, 2021.

[422] "2d face dataset with pain expression:." `http://pics.psych.stir.ac.uk/2D_face_sets.htm`.

[423] C. Saravanan, "Color image to grayscale image conversion," in *2010 Second International Conference on Computer Engineering and Applications*, vol. 2, pp. 196–199, IEEE, 2010.

[424] T. Hadjistavropoulos, K. Herr, D. C. Turk, P. G. Fine, R. H. Dworkin, R. Helme, K. Jackson, P. A. Parmelee, T. E. Rudy, B. L. Beattie, *et al.*, "An interdisciplinary expert consensus statement on assessment of pain in older persons," *The Clinical journal of pain*, vol. 23, pp. S1–S43, 2007.

[425] M. F. Valstar and M. Pantic, "Fully automatic recognition of the temporal phases of facial actions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 1, pp. 28–43, 2011.

[426] C. Shan, "Learning local binary patterns for gender classification on real-world face images," *Pattern recognition letters*, vol. 33, no. 4, pp. 431–437, 2012.

[427] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[428] A. Dhall, R. Goecke, and J. Joshi, "Sfew-pain: In-the-wild pain expressions," in *ACII*, 2022.

[429] L. Breiman, "Random forests," in *Machine Learning*, pp. 5–32, Springer, 2001.

[430] J. D. Gibbons, *Nonparametric statistics: An introduction*, vol. 9. Sage, 1993.

[431] R. Ande, B. Adebisi, M. Hammoudeh, and J. Saleem, "Internet of things: Evolution and technologies from a security perspective," *Sustainable Cities and Society*, vol. 54, p. 101728, 2020.

[432] P. Lucey, J. F. Cohn, I. Matthews, S. Lucey, S. Sridharan, J. Howlett, and K. M. Prkachin, "Automatically detecting pain in video through facial action units," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 3, pp. 664–674, 2010.

[433] M. S. Hossain, "Patient state recognition system for healthcare using speech and facial expressions," *Journal of medical systems*, vol. 40, no. 12, pp. 1–8, 2016.

[434] P. Thiam, V. Kessler, S. Walter, G. Palm, and F. Schwenker, "Audio-visual recognition of pain intensity," in *IAPR Workshop on Multimodal Pattern Recognition of Social Signals in Human-Computer Interaction*, pp. 110–126, Springer, 2017.

[435] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, pp. 18–25, 2015.

[436] A. Ghosh, S. Umer, M. K. Khan, R. K. Rout, and B. C. Dhara, "Smart sentiment analysis system for pain detection using cutting edge techniques in a smart healthcare framework," *Cluster Computing*, pp. 1–17, 2022.

[437] S. Umer, B. C. Dhara, and B. Chanda, "Biometric recognition system for challenging faces," in *2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pp. 1–4, IEEE, 2015.

[438] A. Just, "Two-handed gestures for human-computer interaction," tech. rep., IDIAP, 2006.

[439] H. Hasan and S. Abdul-Kareem, "Fingerprint image enhancement and recognition algorithms: a survey," *Neural Computing and Applications*, vol. 23, no. 6, pp. 1605–1610, 2013.

[440] H. Hasan and S. Abdul-Kareem, "Retracted article: Static hand gesture recognition using neural networks," *Artificial Intelligence Review*, pp. 147–181.

[441] H. Nugroho, D. Harmanto, and H. R. H. Al-Absi, "On the development of smart home care: Application of deep learning for pain detection," in *2018*

IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES), pp. 612–616, IEEE, 2018.

[442] G. Menchetti, Z. Chen, D. J. Wilkie, R. Ansari, Y. Yardimci, and A. E. Çetin, "Pain detection from facial videos using two-stage deep learning," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1–5, IEEE, 2019.

[443] R. A. Virrey, C. D. S. Liyanage, M. I. b. P. H. Petra, and P. E. Abas, "Visual data of facial expressions for automatic pain detection," *Journal of Visual Communication and Image Representation*, vol. 61, pp. 209–217, 2019.

[444] J. J. McAuley and J. Leskovec, "From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 897–908, 2013.

[445] K. Olszewski, Z. Li, C. Yang, Y. Zhou, R. Yu, Z. Huang, S. Xiang, S. Saito, P. Kohli, and H. Li, "Realistic dynamic facial textures from a single image using gans," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5429–5438, 2017.

[446] S. Gruss, R. Treister, P. Werner, H. C. Traue, S. Crawcour, A. Andrade, and S. Walter, "Pain intensity recognition rates via biopotential feature patterns with support vector machines," *PLoS One*, vol. 14, no. 10, p. e0222642, 2019.

[447] L.-P. Morency, R. Mihalcea, and P. Doshi, "Multimodal sentiment analysis," in *Proceedings of the ACL-HLT*, pp. 17–24, 2011.

[448] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *NeurIPS*, vol. 32, 2020.

[449] S. Rosenthal, N. Farra, and P. Nakov, "Semeval-2017 task 4: Sentiment analysis in twitter," in *Proceedings of SemEval*, pp. 502–518, 2017.

[450] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[451] V. Venerito, G. Cazzato, *et al.*, "Large language model–driven sentiment analysis for facilitating fibromyalgia diagnosis," *RMD Open*, vol. 10, no. 2, p. e002xxx, 2024.