

# **STUDY OF DEEP LEARNING BASED CLASSIFICATION OF DIGITAL IMAGES FOR REAL LIFE APPLICATIONS**

Thesis submitted by

**KYAMELIA ROY**

Doctor of Philosophy (Engineering)

Department of Electronics and Telecommunication Engineering  
Faculty Council of Engineering & Technology  
Jadavpur University  
Kolkata, India

**2023**

# **Study of Deep Learning based Classification of Digital Images for Real Life Applications**

Thesis submitted by

**KYAMELIA ROY**

(Index No. 137/19/E)

(Registration No. 1021907009)

**Doctor of Philosophy (Engineering)**

Under the guidance of

**Prof. Dr. Sheli Sinha Chaudhuri**

(Professor, Dept. of ETCE, Jadavpur University, WB, India)

Department of Electronics and Telecommunication Engineering  
Faculty Council of Engineering & Technology  
Jadavpur University  
Kolkata, India

**2023**

Title of the Thesis : **Study of Deep Learning based Classification of Digital Images for Real Life Applications**

Name, Designation and Institute of the Supervisor : **Prof. Dr. Sheli Sinha Chaudhuri**  
Professor, Department of ETCE  
Jadavpur University

List of Publications :

**Journals –**

1. Kyamelia Roy, Sheli Sinha Chaudhuri, Jaroslav Frnda, Srijita Bandyopadhyay, Soumen Banerjee and Jan Nedoma, “Diagnosis of Intestinal Diseases using Endoscopy images based on Encoder-Attention DeepNet,” Communicated to *Sensors*, August 2023. SCI/IF – 3.9. *Under Review*.
2. Kyamelia Roy, Sheli Sinha Chaudhuri, Jaroslav Frnda, Srijita Bandyopadhyay, Ishan Jyoti Ray, Soumen Banerjee and Jan Nedoma, “Detection of Tomato Leaf Diseases for Agro-Based Industries using novel PCA DeepNet,” *IEEE Access*, vol. 11, pp. 14983-15001, February 2023. SCI/IF – 3.476. DOI: 10.1109/ACCESS.2023.3244499
3. Kyamelia Roy, Sayan Pramanik, Sheli Sinha Chaudhuri and Soumen Banerjee, “Deep Neural Network based detection and segmentation of Ships using satellite imagery for Maritime Surveillance,” *Computer Systems Science and Engineering*, vol. 44, No. 1, pp. 647-662, January 2023. SCI/IF – 1.486. DOI:10.32604/csse.2023.024997
4. Kyamelia Roy, Sheli Sinha Chaudhuri and Sayan Pramanik, “Deep learning based real-time Industrial framework for rotten and fresh fruit detection using semantic segmentation,” *Microsystem Technologies (Springer)*, vol. 27, pp. 3365-3375, September 2021. Online: November 2020. SCI/IF – 1.737. DOI: 10.1007/s00542-020-05123-x

**Book Chapter –**

1. Transfer Learning Coupled Convolution Neural Networks in Detecting Retinal Diseases Using OCT Images - Kyamelia Roy, Sheli Sinha Chaudhuri, Probhakar Roy, Sankhadeep Chatterjee and Soumen Banerjee.  
Intelligent Computing: Image Processing Based Applications. Advances in Intelligent Systems and Computing (Springer), Vol. 1157, pp. 153-173, June 2020, Springer, Singapore. Mandal J., Banerjee S. (Eds). [https://doi.org/10.1007/978-981-15-4288-6\\_10](https://doi.org/10.1007/978-981-15-4288-6_10). ISBN: 978-981-15-4288-6.

**International Conferences –**

1. Kyamelia Roy, Sheli Sinha Chaudhuri, Soumi Bhattacharjee and Srijita Manna, “Classification of Citrus Fruits and Prediction of their largest producer based on Deep Learning Architectures,” in Proc. of 6<sup>th</sup> Int. Conf. on Opto-Electronics and Applied Optics (Optronix-2020), Kolkata, West Bengal, India, June 2020. In: Banerjee, S., Mandal, J. K. (eds) Advances in Smart Communication Technology and Information Processing. Lecture Notes in Networks and Systems, vol 165. Springer, Singapore. [https://doi.org/10.1007/978-981-15-9433-5\\_15](https://doi.org/10.1007/978-981-15-9433-5_15)
2. Kyamelia Roy, Sheli Sinha Chaudhuri and Probhakar Roy, “Capsule Neural Network Architecture Based Multi-Class Fruit Image Classification,” in Proc. of 6<sup>th</sup> Int. Conf. on Opto-Electronics and Applied Optics (Optronix-2020), Kolkata, West Bengal, India, June 2020. In: Banerjee, S., Mandal, J.K. (eds) Advances in Smart Communication Technology and Information Processing. Lecture Notes in Networks and Systems, vol 165. Springer, Singapore. [https://doi.org/10.1007/978-981-15-9433-5\\_17](https://doi.org/10.1007/978-981-15-9433-5_17)
3. Kyamelia Roy, Sheli Sinha Chaudhuri, Soumi Bhattacharjee, Srijita Manna and Tandrima Chakraborty, "Segmentation Techniques for Rotten Fruit detection," in Proc. of *IEEE Int. Conf. on Opto-Electronics and Applied Optics (Optronix-2019)*, Kolkata, India, March 2019. DOI: 10.1109/OPTRONIX.2019.8862367
4. Kyamelia Roy, Avirup Ghosh, Debamrit Saha, Jayita Chatterjee, Shayan Sarkar and Sheli Sinha Chaudhuri, “Masking based Segmentation of rotten fruits,” in Proc. of *IEEE Int. Conf. on Opto-Electronics and Applied Optics (Optronix-2019)*, Kolkata, India, March 2019. DOI: 10.1109/OPTRONIX.2019.8862396



PROFORMA – 1

**“Statement of Originality”**

I, **Kyamelia Roy**, registered on **Doctor of Philosophy in the department of Electronics and Telecommunication Engineering** do hereby declare that this thesis entitled **“Study of Deep Learning based Classification of Digital Images for Real Life Applications”** 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 7%.

Signature of Candidate: *Kyamelia Roy*

Date: *4/9/2023*

Certified by Supervisor: *S. S. Chaudhuri*

(Signature with date, seal) *4/9/23*

**Dr. Sheli Sinha Chaudhuri**  
*Professor*  
Dept. of Electronics & Tele-Comm. Engg.  
**JADAVPUR UNIVERSITY**  
**Kolkata-700 032**

PROFORMA – 2

**CERTIFICATE FROM THE SUPERVISOR**

This is to certify that the thesis entitled “**Study of Deep Learning based Classification of Digital Images for Real Life Applications**” submitted by Shri/Smt **Kyamelia Roy**, who got her name registered on **14<sup>th</sup> June 2019** for the award of Ph. D. (Engg.) degree of Jadavpur University is absolutely based upon her own work under the supervision of **Prof. Dr. Sheli Sinha Chaudhuri** and that neither her thesis nor any part of the thesis has been submitted for any degree/diploma or any other academic award anywhere before.

S.S.Chaudhuri 4/9/23

Signature of the Supervisor  
and date with Office Seal

**Dr. Sheli Sinha Chaudhuri**  
*Professor*  
Dept. of Electronics & Tele-Comm. Engg.  
JADAVPUR UNIVERSITY  
Kolkata-700 032

Dedicated to the Holy Lotus feet of Sri Sri Thakur  
Ramakrishna Dev, Sri Sri Ma Sarada Devi and Sri Sri  
Swamiji .....

## Acknowledgement

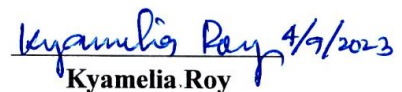
---

Writing of a thesis is the ultimate culmination of an arduous and lengthy journey spanning a long period. Before taking up the task of thesis compilation, it is logical and relevant to express my heartfelt acknowledgement to all those who have been actively or indirectly associated with me in my academic endeavor.

First and foremost, I express my sincere thanks and regards to my PhD supervisor, **Dr. Sheli Sinha Chaudhuri**, Professor and Ex-Head, Department of Electronics and Telecommunication Engineering at Jadavpur University, Kolkata for her constant encouragement, helpful guidance, valuable suggestions as well as providing necessary information and support throughout the period of my research. I am highly indebted to Prof. Chaudhuri for her advices, motivation and kind support in completion of my research work. Her insight and expertise have greatly assisted the research and her critical comments, insightful discussions and suggestions have helped in improving the quality of research. It is her pearl of wisdom that has given a complete shape to the thesis.

I also extend my thanks and appreciation to my students, colleagues, friends and co-researchers for their help and participation in providing support in my research and to all others who have willingly helped me out with their abilities.

Lastly, I would also like to thank all my **family members** for their love, encouragement and kind cooperation extended throughout the research tenure. Their sacrifices, unconditional love, support and continuous prayers have helped me to succeed in life and in successful completion of my research. I am deeply indebted to Sri Sri Thakur Ramakrishna Dev, Sri Sri Ma Sarada Devi and Sri Sri Swamiji for all their graces and blessings rendered from above. It would have not been possible for me to complete my research without their blessings.

  
Kyamelia Roy

## Abstract

---

Artificial intelligence (AI) refers to the simulation of human intelligence by software-coded heuristics. It is based on the principle that human intelligence can be defined in a way that a machine can easily mimic it and execute tasks, from the simplest ones to the more difficult or complex ones. Its ideal characteristic is its ability to rationalize and take actions that have the best chance of achieving a specific goal. A subset of AI is machine learning (ML), which refers to the concept that computer programs can automatically learn from and adapt to new data without being assisted by humans. Deep learning techniques enable this automatic learning through the absorption of huge amounts of unstructured data such as text, images, or video.

The applications for AI in today's society are endless. The technology is applied to many different sectors and industries such as healthcare, astronomy, gaming, finance, data security, social media, transport, automotive industry, robotics, entertainment, e-commerce, agriculture, education, etc. AI is making our daily life more comfortable and fast.

In this thesis, research on some of the applications of AI involving deep learning in healthcare sectors is highlighted. Research is being carried out in exploring the use of deep learning in ophthalmology in diagnosing retinal images for detection of some diseases through use of Optical coherence tomography (OCT) images. Here three diseases, viz. diabetic macular edema (DME), choroidal neovascularization (CNV), and drusen are considered. These diseases are classified using six different convolutional neural network (CNN) architectures and a comparison has been drawn in terms of accuracy, precision, F-measure and recall. The CNN architectures used are coupled with or without transfer learning. The designed models are found to identify the specific disease or no pathology when fed with multiple retinal images of various diseases. The training accuracies obtained for the different CNN architectures viz., four-convolutional layer deep CNNs, Google's Inception v3 and v4 with transfer learning and VGG (VGG-16 and VGG-19) with transfer learning are 87.15%, 91.40%, 93.32%, 85.31% and 83.63% respectively; while the corresponding validation accuracies are 73.68%, 88.40%, 86.95%, 85.30% and 79.50%.

Another application of deep learning in healthcare is depicted in the form of detection of gastrointestinal (GI) tract diseases using Wireless Capsule Endoscopy (WCE) images. Here, a well-defined methodology for the detection of eight classes of diseases viz., Vascular Ectasia, Tapeworm, Crohn's disease, Erosion, Esophagitis, Polyp, Ulcerative-Colitis as well as Normal case is proposed. Leveraging the power of Deep Neural Networks, the approach combines Generative Adversarial Networks (GANs) for image augmentation and Auto-Encoder for feature extraction. Furthermore, an attention-based CNN is employed for accurate classification of different disease classes. To enhance the detection performance, the classified outputs are further refined using a Faster Region-Based CNN architecture. The resulting hybridized framework, Encoder-Attention DeepNet, exhibits outstanding performance achieving a remarkable classification accuracy of 98.80% with an Intersection over Union (IoU) score of 0.9 as compared to existing architectures.

In recent years, computer vision is found to have wide applications in maritime surveillance with its sophisticated algorithms and advanced architecture. Automatic ship detection with computer vision techniques provide an efficient means to monitor as well as track ships in water bodies. In this thesis, a deep learning based model capable enough to classify between ships and no-ships as well as to localize ships in the original images using bounding box technique is proposed. Furthermore, classified ships are again segmented with deep learning based auto-encoder model. The proposed model, in terms of classification, provides successful results generating 99.5% and 99.2% validation and training accuracy respectively. The auto-encoder model also produces 85.1% and 84.2% validation and training accuracies. Moreover the IoU metric of the segmented images is found to be of 0.77 value. The experimental results reveal that the model is accurate and can be implemented for automatic ship detection in water bodies considering remote sensing satellite images as input to the computer vision system.

Farm production is an area which requires various resources, labor, money and time for best result. Now-a-day's farm production is becoming digital, and AI is emerging in this field as a very productive tool for farmers and horticulturist in increasing yield of crops and vegetables. Farm production is applying AI as agriculture robotics, solid and crop monitoring, predictive analysis, pest and plant disease control and for various other applications.

Computer vision finds wide range of applications in fruit processing industries, allowing the tasks to be done with automation. Classification of fruit's quality and thereby gradation of the same is very important for the industry manufacture unit for production of best quality finished food products and the finest quality of the raw fruits to be sellable in the market. In the thesis, detection of rotten or fresh apple has been accomplished based on the defects present on the peel of the fruit. The work proposes a semantic segmentation of the rotten portion present in the apple's RGB image based on deep learning architecture. UNet and a modified version of it, the Enhanced UNet (En-UNet) are implemented for segmentation yielding promising results. The proposed En-UNet model generated enhanced outputs than UNet with training and validation accuracies of 97.46% and 97.54% respectively while UNet as the base architecture attaining an accuracy of 95.36%. The best mean IoU score under a threshold of 0.95 attained by En-UNet is 0.866 while that of UNet is 0.66. The experimental results show that the proposed model is a better one to be used for segmentation, detection and categorization of the rotten or fresh apples in real time.

Also the advancement of Deep Learning and Computer Vision in the field of agriculture has been found to be an effective tool in detecting harmful plant diseases. Classification and detection of healthy and diseased crops play a very crucial role in determining the rate and quality of production. Thus the presented work in the thesis highlights a well-proposed novel method of detecting Tomato leaf diseases using Deep Neural Networks to strengthen agro-based industries. The present novel framework is utilized with a combination of classical Machine Learning model Principal Component Analysis (PCA) and a customized Deep Neural Network which has been named as PCA DeepNet. The hybridized framework also consists of GAN for obtaining a good mixture of datasets. The detection is carried out using F-RCNN. The overall work generated a classification accuracy of 99.60% with an average precision of 98.55%; giving a promising IoU score of 0.95 in detection. Thus the presented work outperforms any other reported state-of-the-art architectures.

## Contents

---

### **Chapter -1 Introduction to Deep Learning**

1.1 Deep Learning .....	21
1.2 Image classification with Deep Learning .....	22
1.3 Some typical Deep Learning architectures .....	26

### **Chapter -2 Implementation of Deep Learning architectures for classification of digital images**

2.1 Introduction to classification of digital images .....	31
2.2 Importance of OCT digital images and deep Learning in Ophthalmology .....	33
2.3 Theoretical Background .....	35
2.3.1 Convolutional Neural Network .....	35
2.3.2 Different Convolutional Neural Network Architectures .....	35
2.3.3 Training of the CNN Architectures .....	37
2.3.4 Dataset .....	37
2.3.5 Python Libraries used .....	37
2.4 Proposed methodology .....	38
2.4.1 Image Pre-processing .....	39
2.4.2 HDF5 files with one_hot encoded labels .....	40
2.4.3 H5PY to feed data set as vectors .....	41
2.4.4 Training module .....	41
2.4.5 Monitoring Training Accuracy and Loss .....	41
2.4.6 Trained module .....	41
2.5 Results and Discussions .....	42
2.5.1 Performance parameters .....	48
2.6 Conclusion .....	53

### **Chapter-3 Implementation of Deep Learning architectures for classification based on segmentation of digital images**

3.1 Introduction to segmentation of digital images .....	55
3.2 Importance of AI / ML in Horticulture .....	57
3.3 Some related works .....	59
3.4 Methodology .....	61



3.4.1	Data preprocessing .....	61
3.4.2	Dataset .....	63
3.5	Network architecture .....	64
3.5.1	UNet .....	64
3.5.2	En-UNet .....	64
3.5.3	Training Module .....	66
3.5.4	Trained Module .....	66
3.6	Results and discussions .....	67
3.7	Conclusion .....	70

**Chapter-4 Implementation of Deep Learning architectures for classification based on segmentation of digital images**

4.1	Introduction to classification and localization of digital images .....	73
4.2	Importance of AI / ML in Maritime surveillance .....	75
4.3	Methodology .....	80
4.3.1	Dataset .....	81
4.3.2	Data pre-processing .....	82
4.3.3	Network Architecture .....	83
4.4	Results and Discussions .....	87
4.5	Conclusion .....	91
4.6	Importance of AI / ML in Agriculture .....	93
4.7	Related Works .....	95
4.8	Methodology .....	98
4.8.1	Dataset .....	98
4.8.2	Data pre-processing .....	100
4.8.3	Annotation .....	101
4.8.4	Data Augmentation .....	102
4.8.5	Feature extraction .....	103
4.8.6	Image classification .....	104
4.8.7	Performance parameters .....	106
4.8.8	Detection .....	107
4.9	Results and Discussion .....	107
4.10	Conclusion .....	119
4.11	Importance of AI / ML in Bio-medical image analysis .....	119

4.12	Methodology .....	125
4.12.1	Data Acquisition .....	126
4.12.2	Dataset .....	126
4.12.3	Data pre-processing .....	127
4.12.4	Image Augmentation .....	130
4.12.5	Feature extraction .....	131
4.12.6	Classification .....	131
4.12.7	Evaluation Metrics .....	134
4.12.8	Detection .....	134
4.13	Results .....	136
4.14	Discussion .....	142
4.15	Conclusion .....	146
<b>Chapter-5 Conclusion and Future scope</b>		
5.1	Conclusion .....	148
5.2	Future scope .....	150
<b>Bibliography</b>		153

## List of Figures

---

2.1	Schematic diagram of CNN architecture .....	37
2.2	Workflow diagram of the proposed methodology .....	39
2.3	Input OCT images of the different eye-diseases along with that of a normal retina ..	39
2.4	Pre-processed images of different eye-diseases along with that of a normal retina ..	40
2.5	Training accuracy and loss for 4 convolution layer deep CNN .....	43
2.6	Training accuracy and loss for Inception v3 with normal training .....	43
2.7	Training accuracy and loss for Inception v3 with transfer learning .....	43
2.8	Training accuracy and loss for Inception v4 with transfer learning .....	44
2.9	Training accuracy and loss for VGG 16 with transfer learning .....	44
2.10	Training accuracy and loss for VGG 19 with transfer learning .....	44
2.11	Classified results with 6 architectures with and without transfer learning .....	46
2.12	Confusion Matrices for the 6 different architectures .....	50
2.13	Plot of model metrics for the different CNN architectures .....	52
2.14	Vanila Saliency maps for the Retinal diseases .....	52
2.15	Occlusion maps for the Retinal diseases .....	53
3.1	Workflow diagram of proposed methodology .....	62
3.2	Data preprocessing and generation of ground truth images (binary masks) .....	63
3.3	Conversion of RGB to Gray images (Apples) and the corresponding binary masks	63
3.4	Block diagrammatic representation of UNet .....	65
3.5	Block diagrammatic representation of En-UNet .....	65
3.6	The convolutional layers of En-UNet for segmentation .....	67
3.7	Accuracy and loss plot - (a) training and validation accuracy of UNet, (b) training and validation loss of UNet, (c) training and validation accuracy of En-UNet, (d) training and validation loss of En-UNet .....	69
3.8	RGB image mask and predicted output (a–b) UNet, (c–d) En-UNet .....	69
3.9	Segmented output images of UNet and En-UNet .....	70
4.1	The workflow diagram of the proposed methodology .....	81
4.2	Typical dataset sample images for ships and no-ships .....	82
4.3	Data preprocessing and generation of ground truth images .....	83
4.4	Created dataset: Conversion of RGB (original images) to corresponding binary masks and ground truth images .....	84
4.5	Block diagrammatic representation of 4-layers CNN .....	85

4.6	Block diagrammatic representation of Auto-encoder .....	86
4.7	Accuracy and loss plot of CNN classifier model .....	88
4.8	Confusion Matrix of CNN classifier model .....	88
4.9	Detection of ships in a test input image .....	89
4.10	Accuracy and loss plot of Auto-encoder model .....	90
4.11	Typical Heat maps of the sample images .....	91
4.12	Block diagram of overall System .....	98
4.13	Dataset images – (a) Healthy (b) Late_blight (c) Early_blight (d) Seportia_leaf_spot (e) Yellow_leaf_curl_virus (f) Bacteria_spot (g) Target_spot (h) Mosaic_virus (i) Leaf_mold (j) Spider_Mites_two_spotted_spider .....	99
4.14	Graphical representation of data pre-processing .....	100
4.15	Schematic diagram of Data Pre-processing .....	101
4.16	Annotation of Tomato Leaf Images (a) Annotated image (b) XML document .....	102
4.17	Image Augmentation using CycleGAN .....	103
4.18	Projection of different classes of leaf after implementing PCA .....	104
4.19	Schematic representation of the classifier model .....	106
4.20	Block diagram of Faster Region-Based Convolutional Neural Networks .....	108
4.21	(a) Accuracy graph of the classifier (b) Loss graph of the classifier .....	109
4.22	Confusion matrix generated by PCA DeepNet .....	110
4.23	Performance evaluation of the different Machine Learning Classifiers on the Plant Village Dataset .....	116
4.24	Detected images of all the classes .....	117
4.25	Detected image of single class in Multiple position .....	117
4.26	Digestive system along with the capsule in the GI tract .....	121
4.27	Different components of Capsule .....	121
4.28	Proposed Workflow Diagram .....	125
4.29	Data Acquisition .....	127
4.30	Dataset images (a) Crohns (b) Erosion (c) Esophagitis (d) Normal (e) Polyp (f) Tapeworm (g) Ulcerative-colitis (h) Vascular Ectasia .....	127
4.31	Schematic diagram of Data Preprocessing .....	128
4.32	Graphical Representation of the Original Dataset .....	129
4.33	Annotated WCE images .....	129
4.34	Image augmentation using CycleGANs .....	131
4.35	Feature extraction using Auto encoder .....	132
4.36	ReLU Function .....	132

4.37	Diagrammatic representation of the Attention based CNN classifier .....	134
4.38	Confusion Matrix for Multi-Class Classification .....	135
4.39	Detection Architecture of F-RCNN .....	136
4.40	Accuracy graph of the Attention-based CNN .....	138
4.41	Loss graph of the Attention-based CNN .....	138
4.42	Confusion matrix of the Attention-based CNN .....	139
4.43	Precision-Recall curve for the Attention-based CNN .....	139
4.44	ROC Curve for the Attention-based CNN .....	139
4.45	Graphical representation of parameters using Attention-based CNN .....	140
4.46	Graphical representation of results for DL Classifiers .....	140
4.47	Graphical representation of results for ML classifiers .....	141
4.48	Detection results generated by F-RCNN .....	144

## List of Tables

---

2.1	Percentage of error in different architectures of CNN .....	38
2.2	Number of layers in different architectures of CNN .....	38
2.3	Accuracy percentages and losses for the 6 different architectures .....	42
2.4	Performance analysis in terms of validation accuracy .....	45
2.5	Performance Analysis in terms of precision .....	50
2.6	Performance Analysis in terms of recall .....	50
2.7	Performance Analysis in terms of F-measure .....	51
3.1	Type and number of layers in UNet and En-UNet .....	67
3.2	Semantic segmentation with UNet and En-UNet and performance parameter – Mean IoU for different output Images (A stands for UNet and B stands for En-UNet) .....	71
4.1	The architecture of 4-layer 2D CNN .....	85
4.2	The architecture of Auto-encoder .....	87
4.3	IoU metric of the segmented images .....	92
4.4	Performance analysis of the proposed model with that of other models for ship detection .....	93
4.5	Detail Information of Dataset employed .....	100
4.6	Detail information of Image annotation .....	101
4.7	Software and Hardware specifications .....	108
4.8	Detail structure of PCA DeepNet .....	111
4.9	Performance parameters obtained from the Confusion Matrix .....	112
4.10	Performance parameters obtained using different Optimizer on PCA DeepNet model .....	112
4.11	Details of Hyper-Parameters used in different architectural models .....	113
4.12	Comparison of existing pre-trained DL classifiers with the proposed work using Plant Village Dataset .....	115
4.13	Hyper-parameters of the Machine Learning Algorithms .....	115
4.14	Comparison of existing pre-trained DL classifiers with the proposed work using Plant Village Dataset .....	115
4.15	Comparison of existing pre-trained DL classifiers with the proposed work using Plant Village Dataset .....	116
4.16	Total images in the Dataset .....	128
4.17	Detail information of Data Annotation per class .....	130

4.18	Hardware and Software Specifications .....	136
4.19	Details of the Attention-based classifier used in WCE-Enttention-DeepNet .....	137
4.20	Results of Attention-based CNN .....	138
4.21	Comparison of different performance parameters generated using state-of-art DL Classifiers and proposed model .....	140
4.22	Hyper-parameters of the Machine Learning Algorithms .....	141
4.23	Performance parameters obtained from Machine Learning Algorithms .....	141
4.24	Comparison of the proposed work with other state-of-art works reported in literature .....	145

# Chapter-1

---

## INTRODUCTION TO DEEP LEARNING



## 1.1. DEEP LEARNING

Deep learning is a subset of machine learning that focuses on the development and application of artificial neural networks to solve complex problems. Computer vision with deep learning has revolutionized the field by enabling computers to understand and interpret visual data. It involves extraction, analysis, and understanding of features from images or videos. Traditionally, this was accomplished through handcrafted feature engineering, where human experts were involved in the task. However, these approaches had limitations in dealing with the inherent complexity and variability of visual data.

On the other hand, deep learning adopts data-driven strategy and automatically extracts representations from the data. It uses artificial neural networks that have many layers of interconnected nodes, or neurons, to replicate the functionalities of human brain. Owing to their depth, or the substantial number of layers, these networks are referred to as deep neural networks.

The most widely used deep learning architecture for computer vision tasks is convolutional neural networks (CNNs) [1-4]. CNNs are made to automatically learn hierarchical visual data representations. They are made up of several layers, including pooling, convolutional and fully connected layers.

The input image is subjected to filters by convolutional layers, which isolate regional patterns like edges, textures, and shapes. The feature maps' spatial dimensions are reduced by the pooling layers; thereby making it easier to extract robust and consistent features. On the basis of the learned features, the final classification or regression task is carried out by fully connected layers, also referred to as dense layers.

A significant labelled dataset is needed to train a CNN. By minimizing a loss function that measures the difference between predicted outputs and ground truth labels, the network learns to optimize its internal parameters (weights and biases) during training. Gradient descent

algorithms, such as stochastic gradient descent (SGD), are frequently used in conjunction with back-propagation to efficiently compute the gradients to achieve this optimization.

A deep learning model can be trained and then applied to a variety of computer vision tasks, such as image classification, object detection, semantic segmentation, image generation, and more. The strength of deep learning is its automatic learning of intricate patterns and features from unprocessed visual data; eliminating the need for explicit feature engineering.

Several computer vision benchmarks have shown that deep learning techniques perform remarkably well, and thus have been successfully used in a plethora of practical applications. They have been deployed for developments in fields like autonomous vehicles, medical image analysis, facial recognition, security systems and even imaginative ones like style transfer and image generation. Deep learning has significantly advanced computer vision by supplying strong tools for complex pattern recognition and automatic feature learning. It has completely changed the field by allowing machines to comprehend and analyze visual data with precision and effectiveness.

## **1.2. IMAGE CLASSIFICATION WITH DEEP LEARNING**

Image classification [5-6] is one of the fundamental tasks in computer vision and deep learning has proven to be highly effective in solving this issue. Deep learning models, particularly CNNs, have achieved state-of-the-art performance in image classification by automatically learning hierarchical representations from raw image data. An overview of the image classification process using deep learning is as follows:

- A. **Dataset Preparation:** The first step is to collect and prepare a labeled dataset for training the deep learning model. This dataset consists of a large number of images, each associated with a corresponding class label. The dataset is typically divided into three subsets: training set, validation set, and test set.
  
- B. **Model Architecture:** The deep learning model's architecture must be designed in the next step. CNNs are frequently used for image classification because of their capacity to learn spatial feature hierarchies. Multiple convolutional layers, pooling layers, and fully

connected layers make up the architecture in most cases. The performance of various CNN architectures, including VGGNet, ResNet and InceptionNet, in image classification tasks has been astounding.

- C. **Training the Model:** The labeled training set is used to train the deep learning model. By minimizing a loss function that gauges the difference between the predicted class probabilities and the ground truth labels, the model learns to optimize its internal parameters (weights and biases) throughout training. Back-propagation is frequently used in conjunction with gradient descent algorithms, such as SGD, to efficiently compute the gradients for this optimization. The model is trained iteratively over a number of epochs, with each epoch involving running the network with the entire training set.
- D. **Hyperparameter Tuning:** For deep learning models to perform at their best, a number of hyper-parameters must be tuned. The learning rate, batch size, network depth, filter sizes, etc. are some examples of these hyper-parameters. The validation set is used for hyper parameter tuning, where various combinations of the parameters are tested to determine the optimized performance.
- E. **Evaluation:** After training, the test set, which consists of unviewed images, is used to assess the model. For each test image, the trained model makes predictions, which are then compared to the ground truth labels to determine accuracy or other evaluation metrics like precision, recall and F1 score. An estimate of the model's performance on hypothetical data is provided by this evaluation.
- F. **Inference:** The model can be used to make predictions on test images after it has been trained and assessed. The trained model runs a network on an input image to process it, and the output layer gives the predicted class probabilities. The predicted label for the input image is the class with the greatest probability. It's important to note that methods like data augmentation, transfer learning, and fine-tuning are frequently used to increase performance and effectiveness for more difficult image classification tasks or when working with little labelled data. Creating a labelled dataset, creating and training a deep

learning model, analyzing its performance, and using the trained model to make predictions on new images are the general steps involved in image classification with deep learning.

G. **Output Layer:** The output layer is typically different from the hidden layers, depending on the specific task and the number of classes in the classification problem. For a multi-class classification task, the output layer usually employs the softmax activation function to convert the raw logits (scores) into a probability distribution over classes. The output of the classifier,  $\hat{Y}_i$ , for the input sample  $X_i$  is given by:

$$\hat{Y}_i = \text{Softmax}(W[L] * Z[L-1] + b[L])$$

$$\hat{Y}_i = \text{Softmax}(W[L] * Z[L-1] + b[L])$$

where, Softmax is the softmax activation function applied to the output of the final hidden layer ( $W[L] * Z[L-1] + b[L]$ ) to obtain class probabilities.

The output of a Deep neural network classifier, denoted by  $\hat{Y}_i$ , for each input sample  $X_i$ , is computed using the following mathematical equation:

**Input Layer:**  $Z[0] = X_i$

**Hidden Layers:** For each layer  $l = 1$  to  $L-1$ , the output of the  $l$ -th hidden layer is computed as:  $Z[l] = \text{Activation}(W[l] * Z[l-1] + b[l])$

where,  $Z[l]$  is the output of the  $l$ -th hidden layer, which is a vector of activation values (outputs of neurons) for that layer,  $W[l]$  is the weight matrix of the  $l$ -th hidden layer, which represents the strength of connections between neurons in layer  $l-1$  and layer  $l$ ,  $b[l]$  is the bias vector of the  $l$ -th hidden layer, which represents the intercept term of each neuron in the layer.

Activation is a function applied element-wise to the output of the linear transformation ( $W[l] * Z[l-1] + b[l]$ ). It introduces non-linearity to the model, allowing it to learn complex patterns in the data. Common activation functions include Rectified Linear Unit (ReLU), sigmoid, and tanh.

During training, the model's weights ( $W[l]$ ) and biases ( $b[l]$ ) are adjusted to minimize the classification error (e.g., cross-entropy loss) between the predicted probabilities ( $\hat{Y}_i$ ) and the true labels ( $Y_i$ ). This is typically done using optimization algorithms like (SGD) or variants such as Adam or RMSprop. The actual architecture of a deep learning classifier designed for any specific task can be much more complex, with various optimization techniques, regularization and advanced activation functions. However, the above mathematical representation captures the fundamental elements of a basic deep learning classifier.

In the present thesis, the research works are focused on three categories of classification:

- i) **Multi-Class Classification:** In multi-class image classification [7-10], images are classified into one of several predefined classes. Each image belongs to only one class. For example, classifying images into categories like "cat," "dog," "bird," and "car."
  
- ii) **Pixel-level Classification (Semantic segmentation):** When discussing semantic segmentation, the term "pixel-level classification" [11-14] refers to the process of giving each pixel in an image a class label with the intention of segmenting and identifying various objects or regions of interest within the image. Instead of a single label per image or per object, it aims to provide a dense pixel-wise classification. In order to divide an image into meaningful and coherent regions for semantic segmentation, each pixel is given a class label that denotes the category or class to which it belongs. The objective is to provide a detail understanding of the objects and regions within the image at the pixel level by giving each pixel a semantic label. A pixel-level segmentation map, also known as a segmentation mask or label map is the result of semantic segmentation, and it is where each pixel is given a class label. Depending on the application and domain, common classes include person, car, tree, road, building, sky, etc. The current research work uses the concept of this pixel-level classification to separate fresh fruits from rotten fruits.

- iii) **Classification followed by localization:** In computer vision tasks, particularly in object detection [15-19], classification is frequently followed by localization. The main concept is to first categorize an object's presence within an image, and then if the object is present, to precisely localize its location within the image.

Using a classification model, like a deep neural network, to identify the presence or absence of particular objects or classes within the image is the first step. The model outputs the odds that each class will be present after receiving the entire image as input. The following step is to precisely localize the bounding box around the object belonging to that class once the classification model predicts that a specific class is present in the image (i.e., a high confidence score for a specific class). The name of this procedure is object localization. The localization task aims to determine the coordinates of the bounding box that tightly encloses the object of interest (typically in terms of top-left and bottom-right corners).

Classification and localization may occasionally be combined into a single model. They are referred to as object detection models. They are made up of two main parts: a localization sub-network for predicting the bounding box's coordinates and a classification sub-network for predicting class probabilities. These models are jointly trained on labelled data, where each training sample contains ground-truth bounding box coordinates as well as class labels. Object detection algorithms can provide not only the class label of detected objects but also their precise spatial location by combining classification and localization. This enables a wide range of applications in industries like autonomous vehicles, surveillance, and robotics.

### **1.3. SOME TYPICAL DEEP LEARNING ARCHITECTURES**

Deep learning classification architectures refer to neural network models designed specifically for the task of image classification. These models aim to take an input image and generate at the output the probabilities or scores for each class label that the image belongs to. Over the years, various deep learning classification architectures have been proposed, each with its unique features and strengths. An insight to some of the popular deep learning architectures implemented for categorization of digital images are as follows:

- A. **Google's (Inception):** The Google research team created the Inception v3 deep learning architecture. It belongs to the group of convolutional neural networks called Inception [20] that are made to perform tasks like object and image recognition. The main goal of Inception v3 was to develop a deep learning model that is computationally efficient, capable of running on a variety of hardware platforms, and capable of achieving high accuracy on image classification tasks. The concept of "Inception" modules, also referred to as "GoogleNet" modules is used in Inception v3. These modules are made up of various parallel convolutional layers with various filter sizes (e.g.,  $1\times 1$ ,  $3\times 3$ , and  $5\times 5$ ) as well as pooling operations. The model can effectively capture features at various scales and resolutions by combining filters of various sizes, which enables it to learn rich representations of the input images. On the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 dataset, a sizable dataset for image classification, Inception v3 demonstrated state-of-the-art performance. In comparison to earlier iterations of the Inception architecture, it showed improved accuracy while maintaining computational efficiency. Inception v3 is an effective deep learning architecture that has influenced the development of computer vision and served as a model for later Inception iterations like Inception v4 and Inception-ResNet.
- B. **VGG:** The Visual Geometry Group (VGG) Net [21] was designed for the ILSVRC 2014 and demonstrated the power of using deeper convolutional neural networks for image classification tasks. The VGG network architecture became a milestone in deep learning and played a crucial role in the development of subsequent deeper networks. The VGG network architecture is known for its simplicity and uniformity. It consists of 16 or 19 layers of small  $3\times 3$  convolutional filters, followed by max-pooling layers. The use of multiple  $3\times 3$  filters instead of larger filters was a novel idea at that time and allowed the network to learn more complex features. The deeper variants of VGG with 19 layers are commonly referred to as VGG-19, while the variants with 16 layers are called VGG-16. VGGNet remains a crucial source of information in the field of computer vision and served as the foundation for numerous subsequent deep learning architectures.

C. **UNet**: A deep learning architecture created for semantic segmentation tasks, where the objective is to identify and categorize each pixel in an image. The UNet architecture [22] has been used for other segmentation tasks in computer vision, but it is particularly well known and widely used in the medical image segmentation field. UNet's U-shaped architecture, which is similar to an encoder-decoder design, is one of its key characteristics. It combines the context that the up-sampling path (decoder) provides with the localization ability of a fully convolutional network (encoder). With its encoder-decoder structure, skip connections, and overall U-shaped architecture, UNet is able to efficiently capture both local and global context while preserving fine-grained localization; particularly in situations where pixel-level accuracy is very essential.

D. **FRCNN**: Finding and locating objects in an image is the task of object detection. The original Region-based Convolutional Neural Network (R-CNN) model suggested using region proposals was created by outside algorithms (like selective search) to identify potential object regions in an image. Faster R-CNN (FRCNN) is an extension of that model. By incorporating the region proposal step directly into the network, Faster R-CNN significantly increases the speed and effectiveness of the R-CNN methodology. Basically, Region Proposal Network (RPN) or Faster R-CNN is a fully convolutional network that produces region suggestions right inside the deep learning architecture. A set of bounding box proposals (candidate regions) with corresponding objectness scores are produced by the RPN. The locations of potential objects in the image are then determined using these suggestions.

In FRCNN [23], the RPN and subsequent object detection network, share the convolutional layers which consist of classification and bounding box regression layers. By removing unnecessary computations, this sharing of convolutional layers increases efficiency and lowers computation. The convolutional layer extracted features are aligned to fixed-size feature maps using Region of Interest (ROI) pooling. This makes the model more adaptable for detecting objects of various scales by enabling it to handle regions of various sizes and shapes. The fixed-size feature maps from ROI pooling are used for bounding box regression and object classification. The regression head fine-tunes the bounding box coordinates for precise localization while the classification head forecasts



the likelihood that each region will belong to a particular object class. Faster R-CNN has significantly increased object detection accuracy and has become the de-facto method for many computer vision applications. It serves as the foundation for many cutting-edge object detection models and has contributed significantly to the development of computer vision.

## Chapter-2

---

# IMPLEMENTATION OF DEEP LEARNING ARCHITECTURES FOR CLASSIFICATION OF DIGITAL IMAGES

# Chapter-2 IMPLEMENTATION OF DEEP LEARNING ARCHITECTURES FOR CLASSIFICATION OF DIGITAL IMAGES

---

## 2.1. INTRODUCTION TO CLASSIFICATION OF DIGITAL IMAGES

Deep learning has revolutionized the field of computer vision by enabling highly accurate and efficient classification of digital images. Deep learning models present cutting edge technologies in variety of image classification tasks to the growing availability of large-scale labelled image datasets and improvements in hardware and algorithms.

The process of classifying images into predetermined classes or labels is referred to as image classification. CNNs, in particular, have demonstrated outstanding success in this field. The purpose of CNNs is to automatically learn how to represent images in a hierarchical manner by using multiple layers of interconnected neurons. These networks can effectively distinguish between different image classes because they can capture intricate patterns and features at various scales. The classification process using deep learning typically involves Data preparation, model architecture, training, evaluation, fine-tuning and optimization, deployment.

Numerous industries, including healthcare, autonomous driving, surveillance, e-commerce, and others, use deep learning-based image classification. It paves the way for sophisticated image recognition systems and intelligent decision-making based on visual inputs by enabling machines to automatically comprehend and interpret visual information.

By utilizing CNN architectures and sizable labelled datasets, deep learning has significantly advanced the field of image classification. This method has proved to be very successful in categorizing digital images automatically, and it has created new dimensions for computer vision applications.

The classification of digital images has been successfully applied to a number of deep learning architectures. Some of the most widely used architectures are as follows:

1. **Convolutional Neural Networks:** The most popular deep learning architecture for image classification uses convolutional neural networks. Convolutional, pooling, and fully

connected layers are just a few of the many layers of interconnected neurons that make them up. By identifying regional patterns and features, CNNs excel at automatically learning hierarchical representations of images.

2. **AlexNet:** One of the first deep learning models to receive widespread attention in the field of image classification was AlexNet. In 2012, it took first place in the ILSVRC. Compared to earlier techniques, AlexNet significantly reduced error rates by introducing the idea of using multiple GPU devices to speed up training.
3. **VGGNet:** The Visual Geometry Group at the University of Oxford created VGGNet, which is known for its uniform architecture and ease of use. It has many convolutional layers with small (3×3) filter sizes, pooling layers, and fully connected layers after that. VGGNet performed admirably in the ILSVRC 2014 competition.
4. **GoogleNet (Inception):** Also known as Inception, GoogleNet introduced the idea of "inception modules" that permit the network to process various receptive fields in parallel. The number of parameters was significantly decreased while still maintaining high accuracy with this architecture. The ILSVRC 2014 competition was won by GoogleNet.
5. **ResNet:** ResNet (short for Residual Network), which addresses the vanishing gradient issue in very deep networks, introduced the concept of residual connections. ResNet architectures, like ResNet-50, ResNet-101, and ResNet-152, have incredibly deep structures and have excelled at a number of image classification challenges.
6. **DenseNet:** This architecture uses a feed-forward connection between each layer and every other layer. It promotes feature reuse and facilitates gradient flow, improving parameter efficiency and accuracy. DenseNet has produced results that are competitive for image classification tasks.
7. **MobileNet:** MobileNet is a compact deep learning architecture made for embedded and mobile devices that have constrained computational power. It uses depth-wise separable convolutions to simplify the computation while preserving a respectable level of accuracy. Real-time systems frequently use MobileNet models.

These are just a few examples of image classification deep learning architectures. Various other architectures and variations have been created to address particular problems and to improve performance in various contexts. The task's complexity, the available computational resources,

and the desired trade-offs between accuracy and efficiency all play a role in the architecture choice. 4-layer CNN, Google's Inception v3 and v4, VGG-16 and VGG-19 are the classification architectures implemented in the present work for the classification task. The results generated with architectures employed with and without transfer learning are also compared.

## **2.2. IMPORTANCE OF OCT DIGITAL IMAGES AND DEEP LEARNING IN OPHTHALMOLOGY**

Eye diseases leading to blindness is a social menace which needs to be eradicated completely for the benefit of mankind and society at large. In developing countries, a large section of the society suffers from various eye diseases which go undiagnosed at times. The socio-economic condition is also a burden which often prevents the patients to treat such diseases at an earlier stage. Research is being carried out worldwide to combat the visual impairment and to provide proper scientific diagnosis and subsequent treatment of the diseases related to eyes. In this context, Optical Coherence Tomography (OCT), a non-invasive optical medical diagnostic imaging modality, has played a key role in being an integral imaging instrument in ophthalmology [24-25]. It generates 3D or cross-sectional images through measurement of echo time delay and magnitude of back scattered or back reflected light. The rapid development and tremendous impact of OCT imaging in clinical diagnosis is increasing day-by-day with its first study on human retina in [26-27]. It provides in-vivo cross sectional imaging of micro-structure in biological system [28-30] and facilitates imaging of retinal structure which cannot be obtained through other non-invasive diagnostic techniques. The ophthalmic treatment proves to be one of the most clinically developed applications of OCT imaging [31-32]. Its popularity across the globe is accounted to the availability of 4<sup>th</sup> generation instruments and half-dozen companies commercializing this technology worldwide for ophthalmic diagnosis. Its advantages for earlier diagnosis of pathologies are accounted for its textural and morphological variations in properties [33-34].

Artificial Neural Network (ANN), on the other hand, has plethora of applications in computer vision, speech processing, medical analysis, etc. The deep learning architectures or models are mostly supported by ANN. These architectures are implemented in various fields and have given promising results which are superior to human analysis in comparison. A deep learning method

was proposed to distinguish between normal OCT retinal images and Age-related Macular Degeneration (AMD or ARMD) affected images [35]. Likewise, an automated segmentation technique for detection of intra-retinal fluid using deep learning method in macular OCT scans was also proposed [36]. The application of deep learning for retinal disease diagnosis in field of ophthalmology has led to the development of a fully automated system for detection and quantification of macular fluid [37]. The results so obtained are highly perfect in terms of accuracy and precision. A popular approach in deep learning is transfer learning which has paved the path of reusing a pre-trained model on a new problem. Transfer learning is widely used in bio-medical image interpretation medical decision making for eye related diseases using retinal OCT images [38]. The retinal OCT images are classified for Diabetic Macular Edema (DME), dry AMD or no-pathology based on transfer learning with pre-trained CNN GoogleNet [39].

Three diseases of eye are considered viz., Diabetic Macular Edema (DME), Choroidal Neovascularization (CNV) and Drusen. The aforesaid diseases have been chosen owing to their threat in causing irreversible vision loss leading to blindness in developed and developing countries [40-42]. Early detection of such diseases would definitely lead to better control and overcoming the curse of blindness. Several researchers across the globe are working on OCT based image classification of DME [43-45], AMD [46], CNV [47] and Drusen [48-49] using deep learning and other CNN tools [50]. In [51], a computer-aided diagnosis model was proposed to distinguish DME, AMD and healthy macula based on linear configuration pattern (LCP) features of OCT images and Correlation-based Feature Subset (CFS) selection algorithm. An automated system using colour retinal fundus images based feature learning approach was developed [52], for earlier diagnosis and treatment of DME. Machine learning algorithm using receiver operator characteristic (ROC) analysis and Cohen's statistics was proposed [53] for automatically grading AMD severity stages from OCT scans. An automated algorithm was proposed for CNV area detection in participants with AMD [54] while automated segmentation of CNV in OCT images were carried out for treatment of CNV diseases [55]. The U-Net CNN architecture is applied for automated segmentation of Drusen from fundus image and further classification of early or advanced stage of AMD [56].

In this chapter, the advanced features of OCT images of retina for eye related diseases are explored for analysis, detection and subsequent classification using various architectures of Deep

Learning. Four pre-trained models viz., Google's Inception v3, Google's Inception v4, VGG-16 and VGG-19 followed by a shallow convnet are used for classification of the eye diseases.

## **2.3. THEORETICAL BACKGROUND**

### **2.3.1 Convolutional Neural Network**

A convolutional neural network (CNN or Convnet) is a type of deep feed-forward neural network finding a huge number of applications related to image analysis. Its receptive field property makes it more suitable for applications related to image processing in areas of bio-medical imaging, remote sensing and Geographic Information System (GIS), cognitive science, etc. The partial extensions of the receptive fields of different neurons aid to cover the entire image for its successful processing. The advantage of CNN is the time complexity which is less in comparison to other image classification methods. Among several properties, an important defining property of CNN makes it Shift Invariant Artificial Neural Network (SIANN) having shared weight architecture and translation invariance characteristics. In Deep Learning models, instead of only one complex transformation  $f(D)$  (where  $D$  represents raw data), multiple transformations are adopted in sequences ( $f(D) = k(g(h(D)))$ ) with decision boundaries present in the final layer of the CNN architecture. CNNs are universally applied in image processing, classification applications and computer vision wherein transformation is achieved through filters at each layer. Figure 2.1 depicts the schematic diagram of CNN architecture with several eye diseases as inputs and their respective classified outputs.

### **2.3.2 Different Convolutional Neural Network Architectures**

Amongst the different CNN Architectures, the ones used for study are 4-Convolution Layer Deep CNN, VGG (VGG-16 and VGG-19) and Google's Inception (Google's Inception v3 and Google's Inception v4). These are chosen owing to their increasing architectural sophistication, low error margin on the Imagenet [57] and better efficiency. Table-2.1 depicts the percentage of error in connection to the popularly available architectures of CNN. Two of the above mentioned architectures viz., 4-Convolution Layer Deep CNN and Google's Inception v3 are implemented with normal training and four architectures viz., Google's Inception v3 with Transfer Training,

Google's Inception v4 with Transfer Training, VGG-16 and VGG-19 with Transfer Training are all implemented with transfer learning. The accuracy in case of transfer learning is observed to be more than that for normal training.

Transfer learning method utilizes a pre-trained neural network and to perform the operation, the final classification layer of the network is removed and the next-to-last layer of the CNN is extracted. The Inception or VGG models pre-trained with Imagenet [57] dataset to classify 1000 classes is used as base models for the architectures using transfer learning. A shallow convolution neural network is made as the top layer to the base model without freezing the base model. The shallow convolution neural network comprises of the following layers:

- Batch normalization layer to increase speed and accuracy.
- ReLU activation layer.
- Dropout layer to prevent overfitting.
- Dense layer with 4 units/nodes (corresponding to 4 output classes) with softmax activation
- Adam optimizer is used with learning rate of 0.001

The image data is then fed from the previously built HDF5 dataset files to the base models without freezing the base models so that while training the transfer learning models, the pre-trained base models can have the ability to reconfigure some of its weights according to new type of data fed to it for better classification purposes. This feature makes the model less computationally expensive than Inception v3 transfer learning model used in [38]. The pre-trained convolution base model just acts as a feature extractor in the training process without adjusting the pre-trained weights any further for different types of dataset that has quite different types of features than the ImageNet dataset, fed to it for classification purposes. Hence with only 10 epochs of run for the used model and the usage of only 16256 images, an accuracy of about 88% on the validation dataset is achieved. In Table-2.2, the summarization of the different layers in each of the architectures is depicted.



### 2.3.3 Training of the CNN Architectures

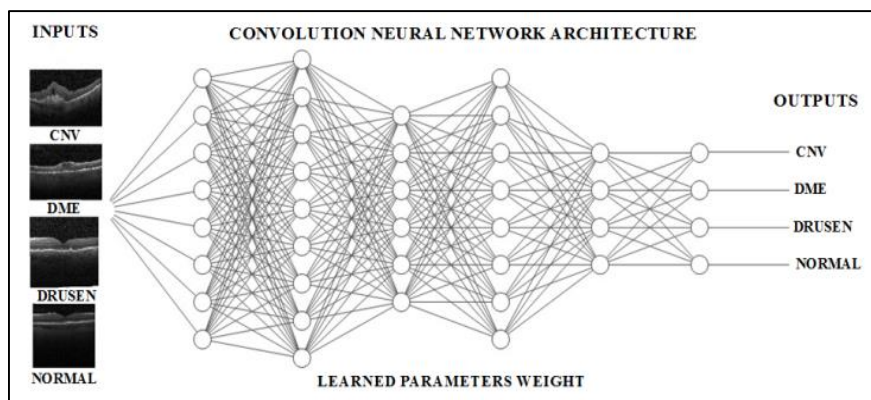
CNN has two cycles - forward and backward cycle. Forward cycle involves classification of the images and computing loss (i.e. the error between the predicted label and actual label) while backward cycle involves correction of the learnable parameters and weights based on the computed loss through a back-propagation algorithm. Backward cycle is only employed during the training phase. In the training phase, the convolutional neural network architecture is fed by batches of images with their actual labels and trained over 10 epochs.

### 2.3.4 Dataset

The OCT Image dataset [38] is used here for classification. The dataset contains about 207130 OCT images in total out of which 108312 OCT images are of 4686 patients (37206 with choroidal neovascularization, 11349 with diabetic macular edema, 8617 with drusen and 51140 normal) with 3 different eye diseases. 16256 OCT images (4,064 with choroidal neovascularization, 4,064 with diabetic macular edema, 4,064 with drusen, and 4,064 normal) are taken in 127 batches to train our convolutional neural network architectures.

### 2.3.5 Python Libraries used

Tensorflow, Tensorboard, Keras, Keras-vis, TFlearn, Numpy, H5py, OpenCV, Matplotlib, Requests, Scikit-learn, Pillow.



**Figure 2.1 Schematic diagram of CNN architecture**

**Table-2.1 Percentage of error in different architectures of CNN**

<b>Networks</b>	<b>Error</b>
AlexNet	16.0%
VGG-16	7.4%
VGG-19	7.3%
GoogLeNet	6.7%
Google's Inception v3	5.6%
Google's Inception v4	5.0%

**Table-2.2 Number of layers in different architectures of CNN**

	<b>Fully connected</b>	<b>2D Conv.</b>	<b>2D Max Pooling</b>	<b>2D Avg. Pooling</b>	<b>Batch Norm.</b>	<b>Drop out</b>	<b>Merge</b>
<b>4 Convolution Layer Deep CNN</b>	4	4	4	0	0	1	0
<b>Inception v3 with Normal Training</b>	2	94	4	9	94	1	10
<b>Inception v3 with Transfer Training</b>	3	95	4	9	95	1	10
<b>Inception v4 with Transfer Training</b>	3	150	4	14	150	1	25
<b>VGG-16 with Transfer Training</b>	3	14	5	0	1	1	0
<b>VGG-19 with Transfer Training</b>	3	17	5	0	1	1	0

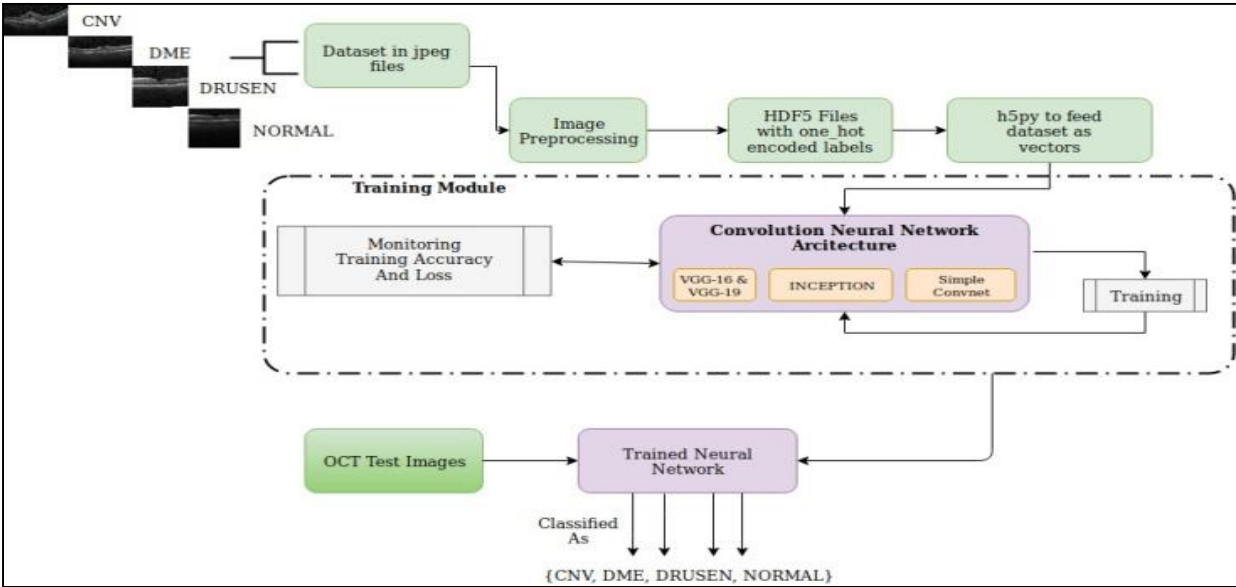
## 2.4 PROPOSED METHODOLOGY

The methodology of the present work is represented in a workflow diagram shown in Figure 2.2. The workflow diagram comprises of different steps involved for the present work. At the onset, the raw OCT images of the different eye diseases, shown in Figure 2.3, are taken as inputs and are pre-processed by means of normalization resulting in images as shown in Figure 2.4. The training and validation datasets comprising of 16256 OCT images for training and 1000 OCT images for validation are created in the HDF5 file. The datasets are then converted into feature vectors and fed to the neural network architectures for continuous training as well as monitoring the accuracy. The training module is followed by testing where validation accuracy is measured

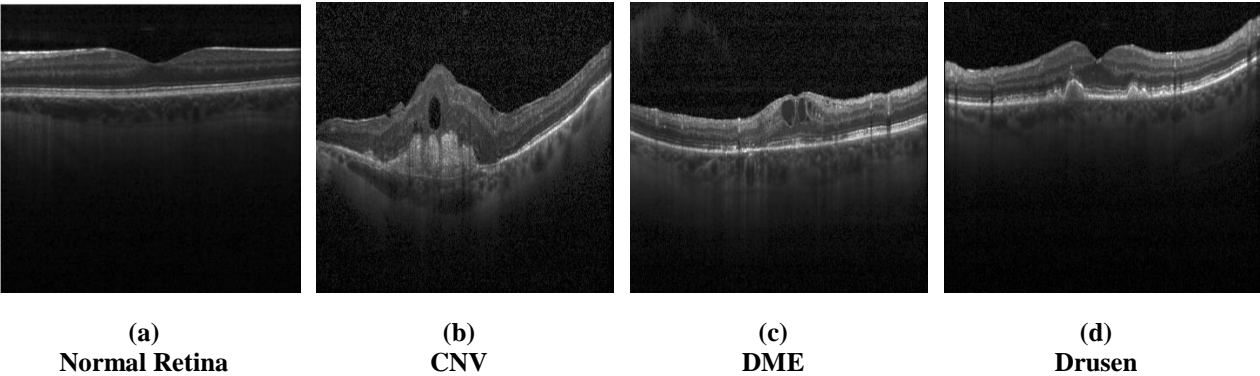
against the unknown image data and classifications of the diseased and normal retina are achieved as the output. A brief description of the different modules of the proposed methodology is discussed below:

**2.4.1 Image Preprocessing**

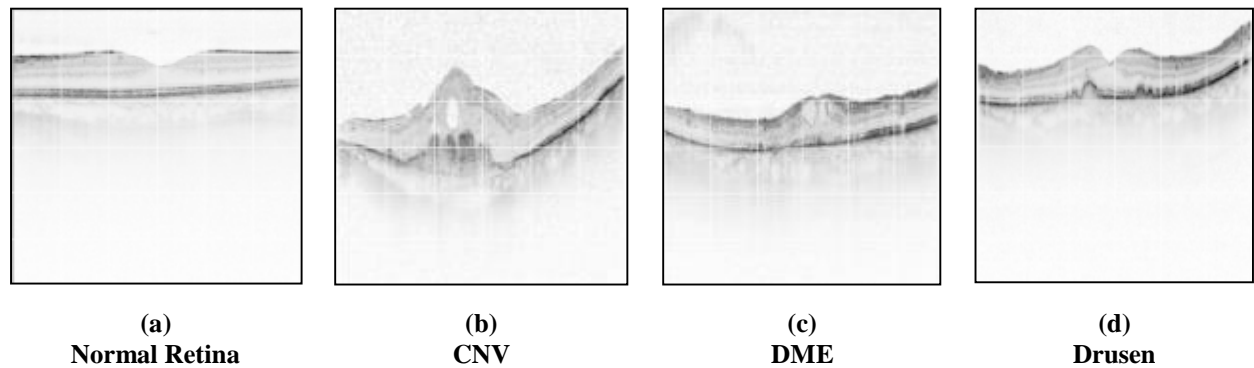
The OCT images in jpeg file format are all resized into images of size 299 pixel  $\times$  299 pixel and normalized i.e., the pixel data is mathematically divided by 255 before feeding them into a HDF5 file to be further fed into the neural network. The pre-processed images are shown in Figure 2.4.



**Figure 2.2** Workflow diagram of the proposed methodology



**Figure 2.3** Input OCT images of the different eye-diseases along with that of a normal retina



**Figure 2.4** Pre-processed images of different eye-diseases along with that of a normal retina

### 2.4.2 HDF5 files with one\_hot encoded labels

An HDF5 file is a container for two kinds of objects: *datasets*, which are array-like collections of data, and *groups*, which are folder-like containers that hold datasets and other groups to address current and anticipated requirements of modern systems and applications. The HDF5 files help to mathematically operate on huge datasets of terabytes of data and load them into the RAM possible. The pre-processed OCT images created are converted into two HDF5 files as training and validation datasets. The 16256 image data forms our feature i.e., ‘X’ and the type of images i.e., CNV, DME, DRUSEN or NORMAL with one\_hot encoding forms our labels i.e., ‘y’ stored in one HDF5 file for the training dataset. Similarly, the 1000 Image data forms our feature i.e., ‘X’ and the type of images i.e., CNV, DME, DRUSEN, or NORMAL with one\_hot encoding forms our labels i.e., ‘y’ stored in one HDF5 file for the validation dataset. A one\_hot encoding is a representation of categorical variables as binary vectors. It is basically labelling of the image data which defines the class of the data for perfect classification at the output stage. This first requires that the categorical values be mapped to integer values and then each integer value be represented as binary vector, of the size of the different types of categorical values (for our case it is four representing three different classes of eye disease and one for the normal eye OCT). The binary vector representation corresponding to a type image representing diseased or normal eye is marked as binary ‘1’ with all other values marked as binary ‘0.’

### **2.4.3 H5PY to feed data set as vectors**

The HDF5 files created in the previous step needs the h5py module for reading purposes in a python script. The h5py module is used to read the HDF5 dataset files and store the image data as feature vector 'X' and one\_hot encoded labels as 'y.' This features and labels are further fed into the convolution neural network architectures in batches of 128 images for 10 epochs.

### **2.4.4 Training module**

Training accuracy gives the percentage of images being used in that training batch labelled with the correct class. Validation accuracy gives the true measure of performance of the architecture on a particular data set which is not used in the training data set. The total numbers of images used for training are 16256 in 127 batches. Each batch contains 128 OCT images. The training time is approximately 1hour for the simple convnet with 4-convolution layer deep CNN with normal training, 4hours for Google's inception v3 CNN with normal training and 4hours each for Google's Inception v3, Google's inception v4, VGG-16 and VGG-19 CNN with transfer learning training. With process in progress, accuracy involved in the training is studied.

### **2.4.5 Monitoring Training Accuracy and Loss**

The training accuracy and loss of the convolutional neural network architectures can be monitored using the event log files generated by the TensorFlow Framework while training the architecture and reading them with TensorBoard where all accuracy and loss metrics gets plotted instantaneously. The instantaneous training accuracy and loss value can also be observed from the Keras or TFlearn metrics while training.

### **2.4.6 Trained module**

A trained neural network is obtained after the training module is continuously trained with the said architectures and performance is continuously monitored from keras and TFlearn metrics. At the trained module, final test accuracy is performed with random set of test images. This testing gives the validation accuracy of the network and by means of which the estimation of the

performance on the classification task is also evaluated. The trained module gives the classified output as CNV, DME, DRUSEN and normal retina which is the prime goal of the research work.

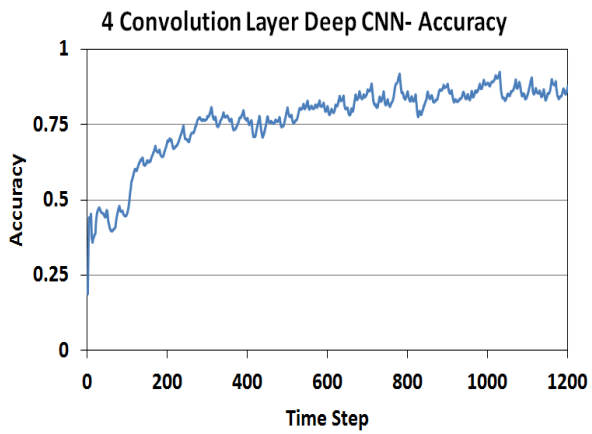
## 2.5. RESULTS AND DISCUSSIONS

The different retinal diseases or malfunctions are evaluated through a comprehensive simulation study and the proposed AI model is validated against different parameters for the different CNN architectures. All neural networks are trained on Google colabs GPU backend with Tesla k80 GPU.

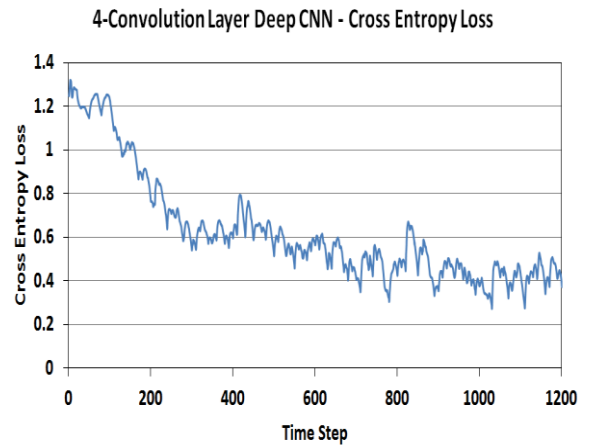
The different CNN architectures under the current study are first compared in terms of training accuracy and loss, obtained through simulation using training data sets as inputs, which are enlisted in Table-2.3. The respective graphs depicting the training accuracy and loss are shown through Figure 2.5 to Figure 2.10. The architectures are then validated using the validation data sets and the accuracies so obtained are enlisted in Table-2.4. Thus the already trained architectures are now capable enough to classify any given OCT retinal images into the four categories of diseases under consideration. Now OCT images are fed to the different architectures and the corresponding classified outputs are viewed through Python output window as depicted through Figure 2.9 to Figure 2.14.

**Table-2.3 Accuracy percentages and losses for the 6 different architectures**

Models	Training Metrics	
	Accuracy	Loss
4-Convolution Layer Deep CNN	87.15 %	0.42
Inception v3 with Normal Training	44.53 %	1.20
Inception v3 with Transfer Training	91.40 %	0.31
Inception v4 with Transfer Training,	93.32 %	0.25
VGG-16 with Transfer Training	85.31 %	0.44
VGG-19 with Transfer Training	83.63 %	0.52

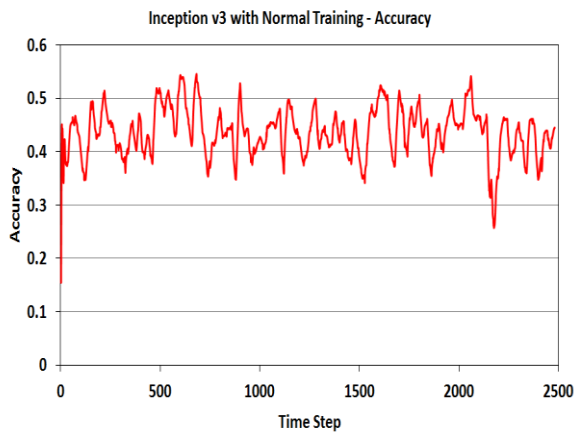


(a)

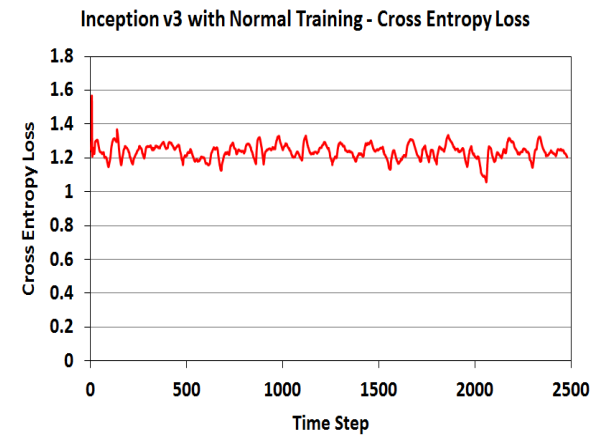


(b)

**Figure 2.5 Training accuracy and loss for 4 convolution layer deep CNN**

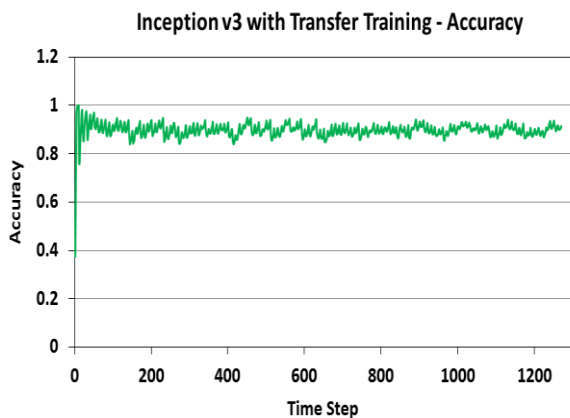


(a)

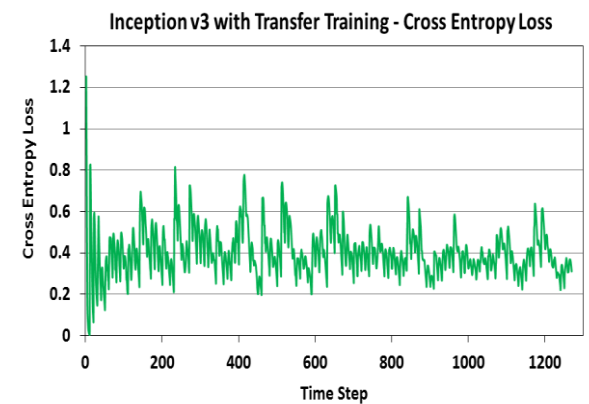


(b)

**Figure 2.6 Training accuracy and loss for Inception v3 with normal training**

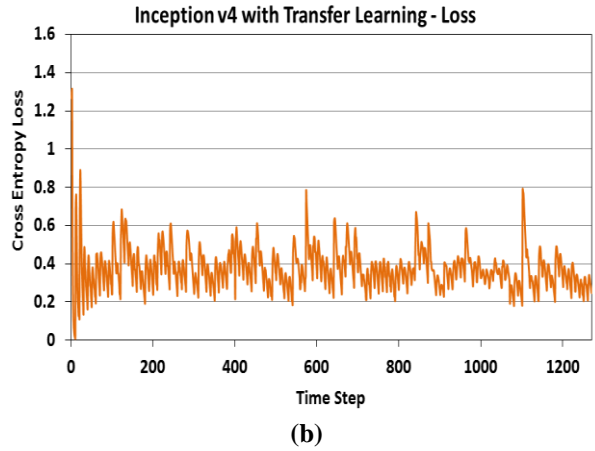
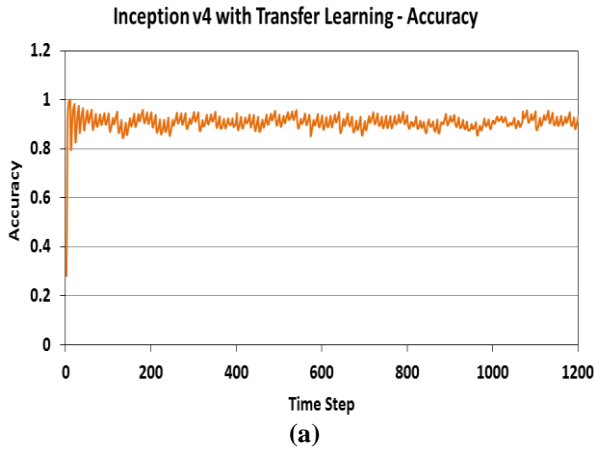


(a)

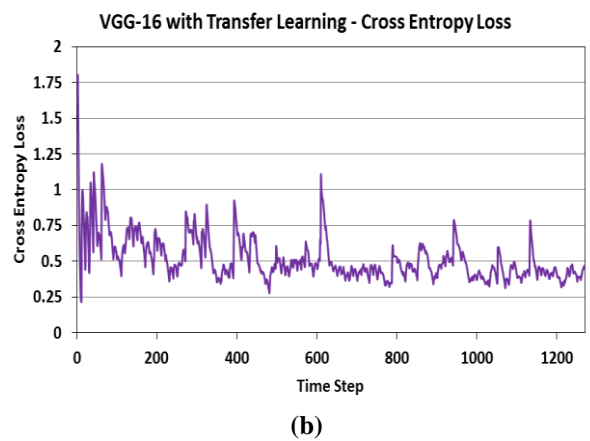
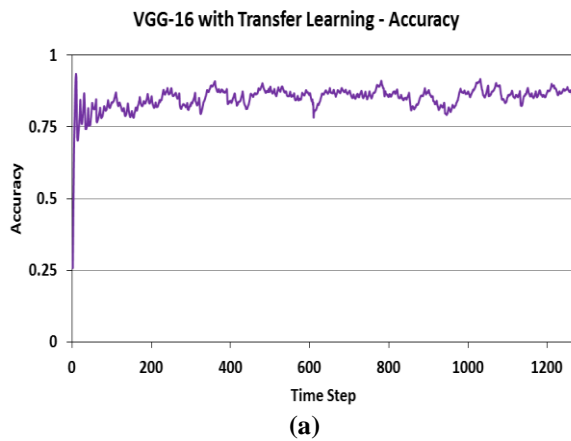


(b)

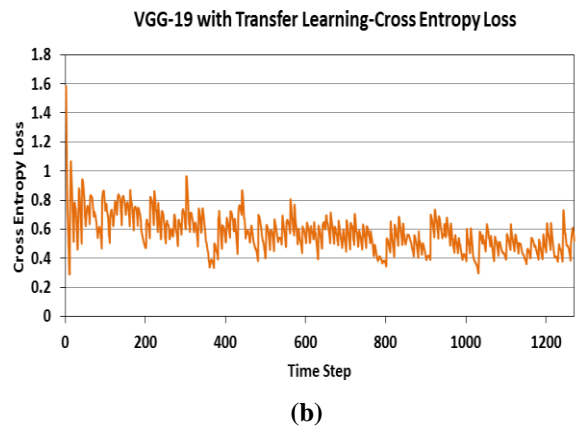
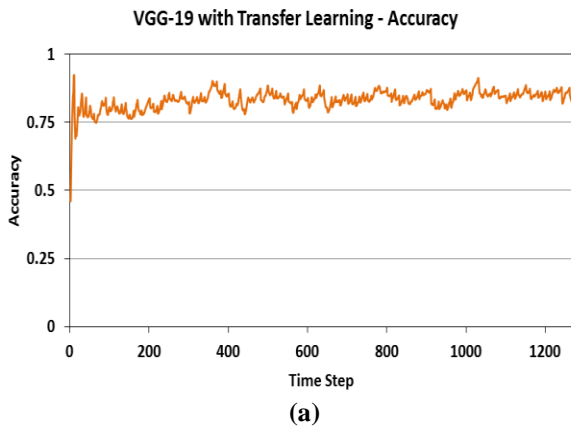
**Figure 2.7 Training accuracy and loss for Inception v3 with transfer learning.**



**Figure 2.8 Training accuracy and loss for Inception v4 with transfer learning**



**Figure 2.9 Training accuracy and loss for VGG 16 with transfer learning**



**Figure 2.10 Training accuracy and loss for VGG 19 with transfer learning**



**Table-2.4 Performance analysis in terms of validation accuracy**

<b>Models</b>	<b>Accuracy</b>
4-Convolution Layer Deep CNN	73.68 %
Inception v3 with Normal Training	25.00 %
Inception v3 with Transfer Training	88.39 %
Inception v4 with Transfer Training	86.95 %
VGG-16 with Transfer Training	85.30 %
VGG-19 with Transfer Training	79.50 %

```
4 CONVOLUTION LAYER DEEP CONVOLUTIONAL NEURAL NETWORK WITH NORMAL TRAINING
Enter The Image Path : ./NORMAL-2633503-1.jpeg

The Image Is Classified to be a NORMAL Retina!
```

**(a). 4-Convolution Layer Deep CNN**

```
GOOGLE'S INCEPTION V3 CONVOLUTIONAL NEURAL NETWORK WITH NORMAL TRAINING
Enter The Image Path : ./DRUSEN-8292670-1.jpeg

The Image Is Classified to be a CNV Retina!
```

**(b). Inception v3 with Normal Training**

```
GOOGLE'S INCEPTION V3 CONVOLUTIONAL NEURAL NETWORK WITH TRANSFER LEARNING TRAINING

Enter The Image Path : ./DME-9925591-2.jpeg

The Image Is Classified to be a DME Retina!
```

**(c). Inception v3 with Transfer Training**

```
GOOGLE'S INCEPTION V4 CONVOLUTIONAL NEURAL NETWORK WITH TRANSFER LEARNING TRAINING

Enter The Image Path : ./CNV-8598714-1.jpeg

The Image Is Classified to be a CNV Retina!
```

**(d). Inception v4 with Transfer Training**

```
VGG-16 CONVOLUTIONAL NEURAL NETWORK WITH TRANSFER LEARNING TRAINING

Enter The Image Path : ./CNV-28682-9.jpeg

The Image Is Classified to be a CNV Retina!
```

**(e). VGG-16 with Transfer Training**

```
VGG-19 CONVOLUTIONAL NEURAL NETWORK WITH TRANSFER LEARNING TRAINING

Enter The Image Path : ./CNV-8598714-1.jpeg

The Image Is Classified to be a CNV Retina!
```

**(f). VGG-19 with Transfer Training**

**Figure 2.11 Classified results with 6 architectures with and without transfer learning**

For the first architecture, 4-Convolution Layer Deep CNN with normal training, the training accuracy obtained is 87.15% with 0.42 cross entropy loss (shown in Figure 2.5) as compared to the validation accuracy of 73.68%. From Figure 2.11(a) it is revealed that with this architecture proper classification is obtained for the diseased image ‘Drusen’ thereby authenticating the validation accuracy.

For the second architecture which is Google’s Inception v3 CNN, the training accuracy obtained is 44.53% (shown in Figure 2.6a) against a validation accuracy of 25% with normal training. The cross entropy loss in this case is 1.2 (shown in Figure 2.6b) which is very high in comparison to losses associated with other architectures. Thus owing to higher loss and lower values of training and validation accuracy, this architecture is not a promising one and leads to misclassification of images which is quite evident in case of a typical ‘Drusen’ image being classified as ‘CNV’ as revealed in the Python output window shown in Figure 2.11(b). However the same architecture trained with transfer learning yields better results in terms of both training and validation accuracies of 91.4% and 88.4% respectively with only 0.31 cross entropy loss (shown in Figure 2.7). Hence the classified output shown in Figure 2.11(c) perfectly classifies the OCT images as that of DME which authenticates the results.

Similar results are also obtained with Google’s Inception v4 architecture where the training accuracy, loss and validation accuracy are 93.32%, 0.25 and 86.95% respectively. The plot of training accuracy and loss are shown in Figure 2.8 while the classified output image is shown in Figure 2.11(d). Thus both Google’s Inception v3 and v4 with transfer learning gives better performance with superior classification abilities.

The next series of architectures with transfer training are VGG-16 and VGG-19 yielding training accuracies of 85.31% and 83.63% respectively with corresponding validation accuracies of 85.3% and 79.5% (shown in Figure 2.9 and Figure 2.10 respectively). These CNN architectures are already been pre-trained with Image Net dataset and thus when trained with transfer learning produces higher values of accuracy. They also require less training time due to the pre-trained Image Net weights and in such architectures the learning performance is also very fast. In both cases the amount of cross entropy losses are less and both gives perfect classified outputs when fed with OCT test images as shown in Figure 2.11(e) and Figure 2.11(f) respectively. Thus

collectively studying CNN architectures it is concluded that among the six architectures, Google’s Inception v3 with transfer learning is recorded with the highest accuracy than Google’s Inception v4 with transfer learning followed by VGG-16, VGG-19, 4-Convolution Layer Deep neural network and Inception v3 with normal training. Moreover architectures with transfer learning gives better performance than its normal training counterpart and hence Google’s Inception v3 and v4 architectures with transfer learning are obvious choices for classification amongst the proposed ones.

**2.5.1 Performance Parameters**

To validate the performance of a system few measures has been taken into consideration viz. sensitivity, specificity and accuracy. These statistical metrics are calculated in terms of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) through use of equations (1) to (4).

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots (1)$$

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots (2)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots (3)$$

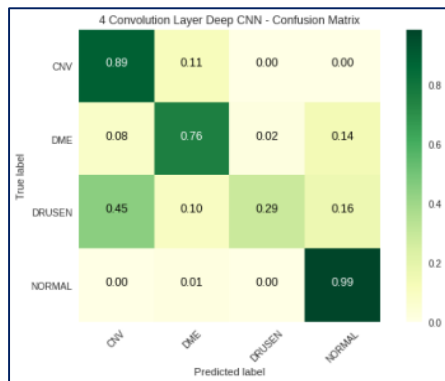
$$FMeasure = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \dots\dots\dots (4)$$

Based on true positive outcome, the system will correctly predict the presence of disease and for true negative; it correctly predicts the absence of the disease. Similarly false positive outcome predicts the presence of the disease whereas in reality there is no disease and false negative gives the absence of the disease in presence of it. The way of summarization of prediction of the results classifying as TP, TN, FP and FN is a confusion matrix. The confusion matrices depicting the performance of the different CNN architectures on a set of test data for which the true value is known are shown in Figure 2.12. The matrices provide visualization of the degree of correlation between true labels with the predicted labels for the different architectures. Form the matrices it is

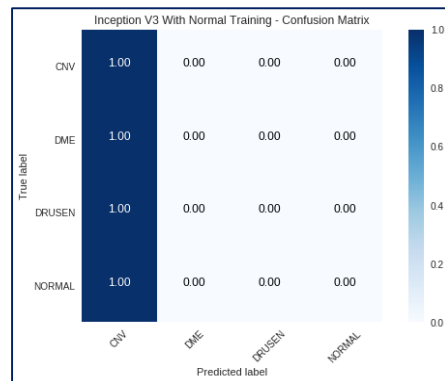
quite evident that for all the architectures except Google’s inception v3 with normal training, the results obtained are satisfactory thereby reflecting perfect classification of the eye diseases.

The performance characteristics of the different CNN architectures are further studied in terms of precision, recall and F-measure which are calculated from their respective test phase confusion matrices. Table-2.5 reports the precision achieved by different classifiers. For each classifier the values of Micro, Macro and Weighted Precision are found to be close to each other.

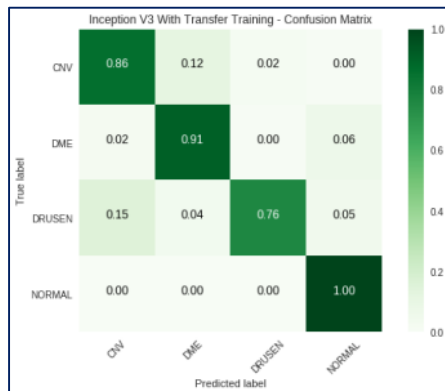
Table-2.6 shows the performance analysis in terms of Recall. The table establishes the ingenuity of “Inception V3 with Transfer Training” as it achieved highest Recall. Finally, Table-2.7 reports the comparative analysis in terms of F-Measure. In terms of F-Measure, the performance of “Inception V4 (With Transfer Training)” and “VGG-16” (with Transfer Training)” is similar. However, the performance of “Inception V3 (With Transfer Training)” is again better than other architectures under the current study.



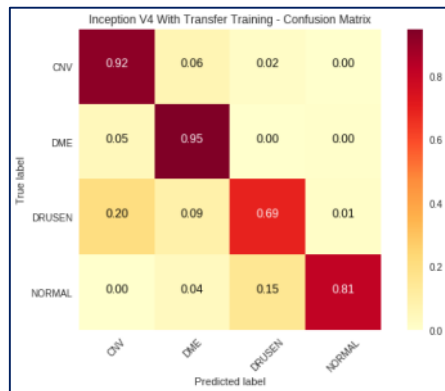
(a). 4-Convolution Layer Deep CNN



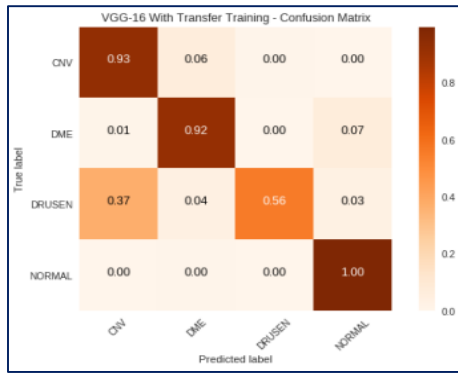
(b). Inception v3 with Normal Training



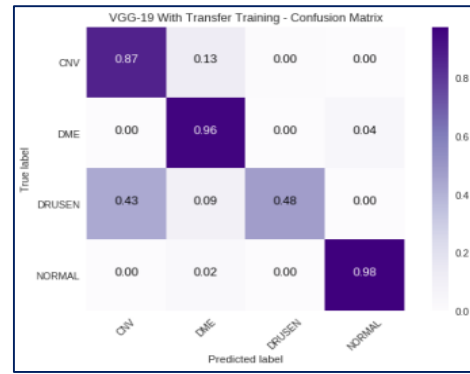
(c). Inception v3 with Transfer Training



(d). Inception v4 with Transfer Training



(e). VGG-16 with Transfer Training



(f). VGG-19 with Transfer Training

Figure 2.12 Confusion Matrices for the 6 different architectures.

Table-2.5 Performance Analysis in terms of precision

Models	Precision		
	Micro	Macro	Weighted
4 Convolution Layer Deep CNN	77.74%	73.10%	77.74%
Inception v3 With Normal Training	6.25%	25.00%	6.25%
Inception v3 With Transfer Training	88.95%	88.40%	88.95%
Inception v4 With Transfer Training	85.05%	84.30%	85.05%
VGG-16 With Transfer Training	87.77%	85.30%	87.77%
VGG-19 With Transfer Training	85.43%	82.00%	85.43%

Table-2.6 Performance Analysis in terms of recall

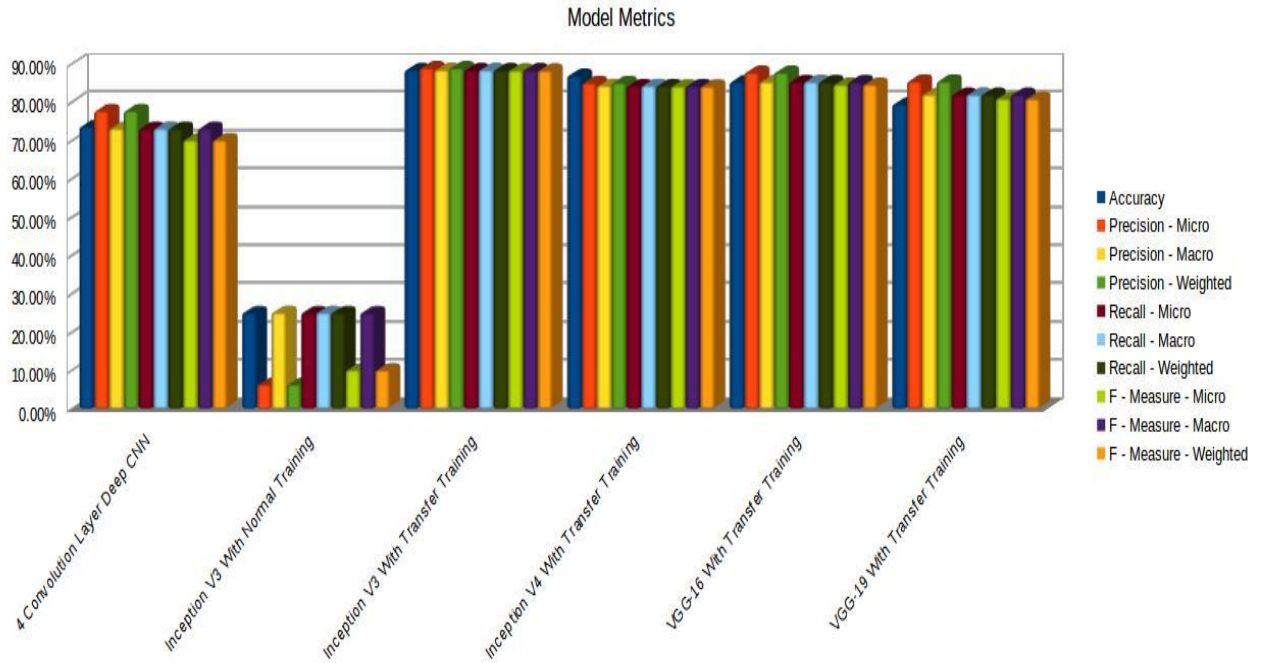
Models	Recall		
	Micro	Macro	Weighted
4 Convolution Layer Deep CNN	73.10%	73.10%	73.10%
Inception v3 With Normal Training	25.00%	25.00%	25.00%
Inception v3 With Transfer Training	88.40%	88.40%	88.40%
Inception v4 With Transfer Training	84.30%	84.30%	84.30%
VGG-16 With Transfer Training	85.30%	85.30%	85.30%
VGG-19 With Transfer Training	82.00%	82.00%	82.00%

**Table-2.7 Performance Analysis in terms of F-measure**

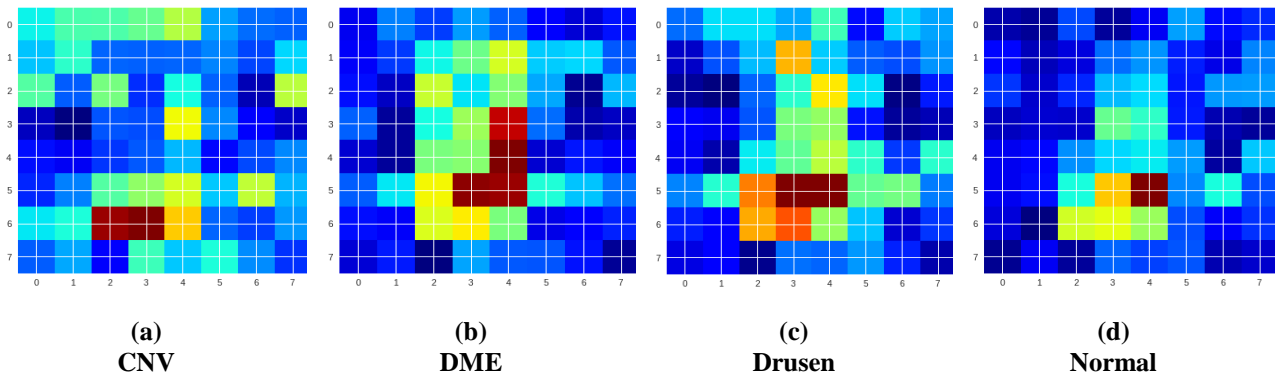
Models	F – Measure		
	Micro	Macro	Weighted
4 Convolution Layer Deep CNN	70.10%	73.10%	70.10%
Inception v3 With Normal Training	10.00%	25.00%	10.00%
Inception v3 With Transfer Training	88.25%	88.40%	88.25%
Inception v4 With Transfer Training	84.14%	84.30%	84.14%
VGG-16 With Transfer Training	84.60%	85.30%	84.60%
VGG-19 With Transfer Training	80.94%	82.00%	80.94%

A plot of model metrics for the different CNN architectures is depicted in the form of a bar graph shown in Figure 2.13.

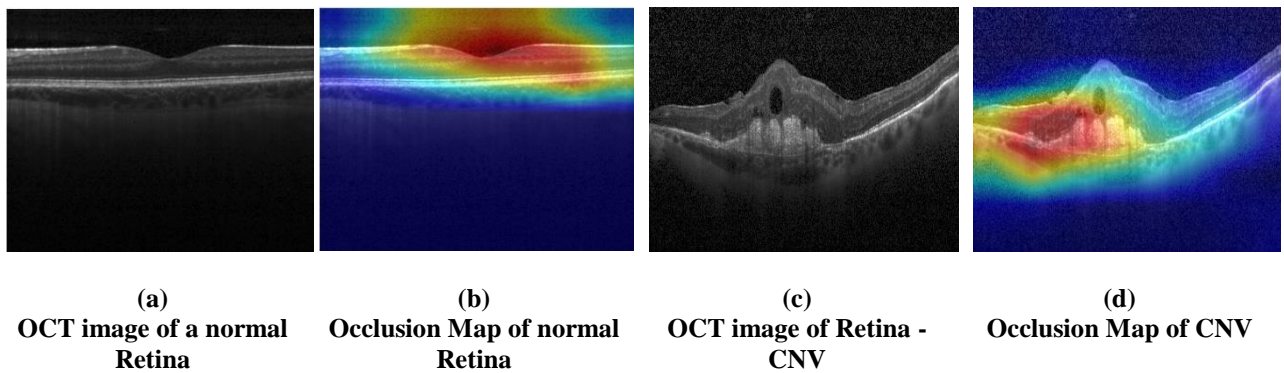
As a test to facilitate the classification process, the vanilla saliency map for the four retinal diseases under consideration are obtained as shown in Figure 2.14. Thus any slight variation in input OCT images would lead to small variation in the output and these gradients would highlight salient image regions that most contribute towards the output. To make the decision for the final stage classification, occlusion test is a very efficient technique. Gradient weighted class activation mapping (Grad-cam) technique is used in the present work to produce these occlusion maps. The occlusion maps for the retinal diseased images shown in Figure 2.15, eminently shows the regions of interest of the target OCT image. These regions are the ones where the deformations have occurred for the diseased retina and also those in case of a normal retina are clearly visible from the occlusion maps for the four categories of OCT images. Occlusion maps are computed over the last convolution layer whereas the saliency maps are computed over the dense output layers. Occlusion maps contains more details than saliency maps since they use pooling features that contains more spatial details which gets lost in the dense layers.



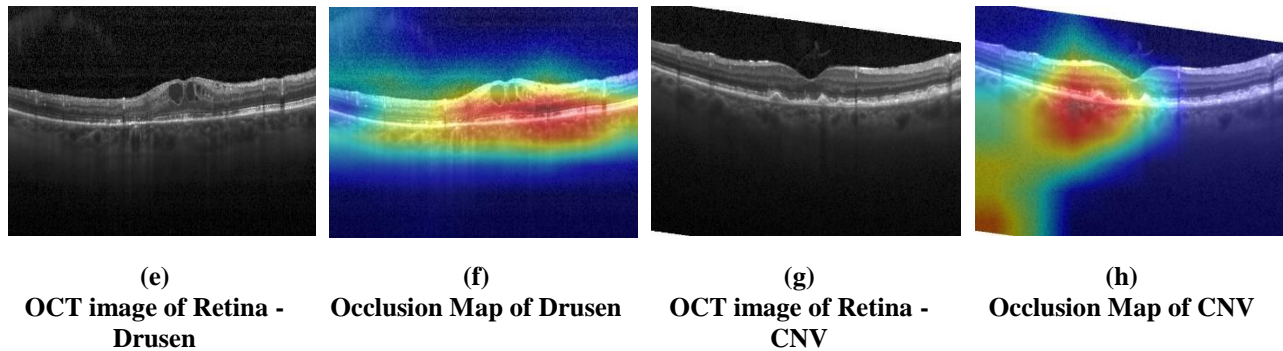
**Figure 2.13** Plot of model metrics for the different CNN architectures



**Figure 2.14** Vanilla Saliency maps for the Retinal diseases







**Figure 2.15 Occlusion maps for the Retinal diseases**

## 2.6 CONCLUSION

OCT based retinal images have been analyzed for the specific disease detection. The OCT retinal images being the input data are implemented with six different CNN models and classified results are obtained at the output. The output is correctly classified for the different 4-classes of the retinal images as shown in Figure 2.11. The proposed model with and without pre-trained data set has given good accuracy percentages except with the model inception v3 without transfer learning. Hence, from the present research work it can be concluded that models with transfer learning yields more accuracy than without it. At the output layers of the CNN architecture how the output is classified based on the region of interest are also shown by the saliency maps and occlusion maps for the four categories. These occlusion maps show the accurate regions of the affected retina and hence can also be shared with the clinical professionals. Thus the method used in the present research work is able to classify among the diseases as well as detect it with a lesser number of epochs and with higher accuracy. This ability of the proposed research work can be used to assist the clinicians in the future where the number of patient is huge to detect the diseases. The research work would definitely open up new window of research amongst the research community world-wide to carry out further researches in fields of bio-medical imaging.

## Chapter-3

---

# IMPLEMENTATION OF DEEP LEARNING ARCHITECTURES FOR CLASSIFICATION BASED ON SEGMENTATION OF DIGITAL IMAGES

# Chapter-3 IMPLEMENTATION OF DEEP LEARNING ARCHITECTURES FOR CLASSIFICATION BASED ON SEGMENTATION OF DIGITAL IMAGES

---

## **3.1. INTRODUCTION TO SEGMENTATION OF DIGITAL IMAGES**

Pixel-level classification, also referred to as semantic segmentation, assigns a class label to each individual pixel in an image. It aims to understand the fine-grained details within an image by labelling each pixel based on its semantic meaning. This is in contrast to object detection or image classification, which operates at a higher level by identifying objects or labelling the entire image.

Pixel-level classification aims to divide an image into regions where each pixel is assigned to a particular class or category. For instance, various regions could represent the sky, road, buildings, trees, etc. in an outdoor scene. A thorough understanding of the scene's composition can be gained and useful data can be extracted for a variety of applications by labelling each pixel. Also a foreground object can be classified from the background of an image by implementing pixel level classification.

Due to the complexity and variety of real-world images, pixel-level classification is a difficult task. To accurately assign class labels to pixels, models must learn both high-level contextual information and low-level features, such as edges and textures. While more recent techniques make use of deep learning, particularly CNNs, to automatically learn hierarchical representations from data, traditional approaches heavily rely on handcrafted features and segmentation algorithms.

For pixel-level classification, deep learning models typically use an encoder-decoder architecture. By gradually down-sampling the input image, the encoder network, which is typically built on a pre-trained CNN, captures the low-level features. After up-sampling the low-resolution features and combining them with skip connections from the encoder to keep the high-

level contextual information, the decoder network recovers the spatial resolution. This procedure enables the model to produce detailed predictions for each input pixel.

It takes annotated training data, where each pixel is labelled with its corresponding class, to train a pixel-level classification model. As it frequently requires manual annotation by human subject-matter experts, this can be a time-consuming and expensive process. The model can, however, be used to automatically classify each pixel in unseen images after being trained, opening up a variety of applications like scene understanding, image segmentation, autonomous driving, medical image analysis, and more.

A computer vision task called "pixel-level classification" entails assigning a class to each pixel in an image. Deep learning techniques have made significant advancements in this difficult problem, enabling more precise and effective semantic segmentation of images.

For pixel-level classification tasks, semantic segmentation with U-Net is an effective and successful method. The encoder-decoder architecture known as U-Net, which was created specifically for semantic segmentation, is renowned for its capacity to effectively handle both local and global context and capture detailed information. The name "U-Net" comes from the network architecture's resemblance to a "U" shape.

The expanding path (decoder) and the contracting path (encoder) are the two main components of the U-Net architecture. While the expanding path creates a dense pixel-wise segmentation map, the contracting path extracts features from the input image and captures context.

The low-resolution feature maps are upsampled by the expanding path, which also restores the spatial data that was lost during the downsampling procedure. The upsampling operation is followed by a concatenation with the corresponding feature maps from the contracting path in each step of the expanding path. As a result, the decoder can produce precise and thorough segmentations by utilising both low-level and high-level features.

U-Net makes use of skip connections to increase localization accuracy and provide more precise boundaries. The network can access data at various scales with concatenation which directly link the feature maps from the contracting path to the corresponding layers in the expanding path. U-

Net effectively manages both local and global context by combining features from various levels, improving segmentation performance.

Typically, the cross-entropy loss or another pixel-wise classification loss is used to optimize U-Net during training. A dense prediction map is created by the network using an image as input, and it is then compared to the ground truth annotation at each pixel. The network parameters are updated through the use of back-propagation, which enables the model to develop precise pixel-level segmentations.

Several semantic segmentation tasks, including medical image segmentation, satellite image analysis, and scene understanding in autonomous vehicles, have been successfully completed using U-Net. It is a well-liked option for pixel-level classification due to its effectiveness and efficiency, and it has served as the foundation for much cutting-edge architecture in semantic segmentation.

### **3.2. IMPORTANCE OF AI / ML IN HORTICULTURE**

Artificial intelligence (AI) with the aid of computer vision is boosting various sectors for quality production with high efficiency. In fields of agriculture and food industry, AI extends its help to the farmers and the manufacturers to improve their effectiveness and to overcome the traditional challenges under environmental hostile impacts. The adoption of AI in the agro-based industries has strengthened the technology to a greater extent. With the implementation of automation techniques, the food processing units have shown promising outcomes owing to excellent production and smart packaging.

In the twenty-first century, both fruit and food processing industries are undergoing soaring competitive positions. Global trade and market flow of fruits and vegetables determines geographical closeness between exporter and importers. In case of exporting or importing, there is a long and time-consuming process of transportation which causes hindrance in checking the quality of rotten or nearly rotten fruits amongst a bulk quantity of fruits. Thus production of fruits is likely to shrink to a greater extent as compared to previous years' world fruit production and trade. Besides all other obstacles, uncertain weather conditions, climate change and

temperature rise are other major causes of concern behind the declination of trade. Moreover, apart from export and import of fresh fruits, the food processing sectors is also severely hampered owing to scrutinization of rotten fruit and degradation in its quality. Thus a professional and efficient method is required for sorting and gradation of good quality fruits. Herein application of computer vision based systems facilitated with the image processing techniques yields highly accurate and precise results. The methodological proficiency enables computer vision based systems for checking defective fruits and thereby improving the potential for healthy fruit production. Computer vision makes possible, practices for monitoring, harvesting and managing effectively various types of fruits in the orchards and crops in the fields [58]. Quality inspection and the categorization of fruits in the traditional methods require skilled persons. Human involvement in performing the task of detecting rotten or defective fruits varies from person to person based on just seeing and sensing from outside. Furthermore, the requirement for enhanced quality food products is increasing under new governmental regulations and consumer market demands. An automated system based on capturing the image of the fruit thereby processing on the detailed features of the image results in high precision and fast response. Computer vision systems supported with deep learning architectures for detection of rotten or fresh fruit are becoming popular among the researchers [59]. These types of automated systems generate accurate outcomes.

Thus, the idea of a computer vision based system for detection of rotten or fresh apple is presented in the present chapter. The rotten apple is identified using segmentation technique wherein the rotten portion of the fruit is segmented out. The segmentation process has been done based on deep learning architecture, taking RGB image of apple (rotten/ fresh) as input to the system. Color being the primary identifier by appearance determines the quality of the fruit. As every apple is different and are not identical, the conventional image processing techniques are not going to help out with better results. In the last 5–6 years with increased computation power and availability of the dataset, the deep convolution networks performed excessively well in visual recognition and identification. UNet has been widely used for diagnosing medical images and the network outperformed the traditional segmentation techniques [22]. UNet or any of its modified form has not been implemented so far in horticulture for the purpose of segmentation in the fruit's image. Hence, segmentation task is done here using UNet and a

modified version of it, the Enhanced UNet (En-UNet). The En-UNet is found to produce better results than the original UNet in terms of accuracy, precision and other model performance parameters and hence can be implemented with a developed automated system in the domain of horticulture.

### **3.3. SOME RELATED WORKS**

Segmentation techniques for fruit detection, identification and classification have been studied extensively in the past decades. A deep learning segmentation has been implemented in analyzing the blueberry fruit traits like cluster compactness, maturity of the fruit and number of berries for ease of the blueberry breeders [60]. A masked R-CNN model has been trained and tested for the detection purpose of the blueberries on maturity. The model resulted with average precision for validation and test data is 78.3% and 71.6% considering 0.5 over union threshold. The accuracies are 90.6% and 90.4% respectively. Green Shoot thinning [61] in wine grapes is done for quality wine with an automated process that would yield higher efficiency and good performance. Deep learning architectures like Segnet [62] and Fully Convolutional Network (FCN) are implemented for the successful semantic segmentation. The FCN model viz. FCN VGG-16 achieved better F1 score as compared to Segnet-VGG16, Segnet-VGG19 and FCN-Alexnet. In [63], the authors proposed a network model for performing real-time detection with segmentation of apples and branches. The developed network mainly utilized the atrous spatial pyramid pooling and gate feature pyramid arrangement for feature extraction. ResNet-101 with 0.832 F1 score showed very high performance in detection of the apples with accuracy of 87.6% on segmentation task. A voxel-based convolutional network (VCNN) [64] is implemented for classification and segmentation of structural components in maize. The comparison results with some traditional clustering techniques and deep learning methods (Pointnet and Pointnet++) showed very good results. The Lidar's implementation for the separation of structural components in maize by means of classification and segmentation paved a new path to be applied for other fields as well. Detection of foreign objects (eg. dried leaves, paper, packing materials, plastics or metal parts) in walnuts has been done using fully convolutional deep network [65]. The proposed model acknowledged with the segmentation task generating good outputs. The authors [66] in

their paper has proposed a hybrid model for classification of mangoes using segmentation technique. The approach is based on BPNN and discriminant analyzer. An enhanced method using Fuzzy C-means clustering has been implemented for the segmentation and back propagation based discriminant classifier is used in the classification practice. For the process of feature selection Maximally Correlated Principal Component Analysis (MCPCA) is utilized. Authors in the paper [67] have undergone the segmentation task for cucumber detection. Masked R-CNN with Feature Pyramid Network (FPN) has been executed. For identification purpose, deep learning architectures have excelled and generated perfect outputs. Identification of flowers (apple, peach and pear) by the proposed method [68] generated semantic segmentation. An end-to-end residual CNN has been employed for the work. Segnet architecture effective for semantic segmentation is a fully convolutional network with the deployment of encoder and decoder. The network has been compared with FCN [69], Deeplab-large FOV [70], DeconvNet [71] and reported to have better segmentation results.

The traditional machine learning approaches have put forward many detection, segmentation and classification algorithms. However the classical methods involve a lot of hand crafted procedures for feature selection and extraction. The authors in their paper [72] suggested a method of colorful fruit image segmentation taking texture features into account. The process of segmentation followed by classification using multi-class support vector machine (MSVM) can be used for multi-class classification [73]. Algorithms such as Naive Bayes, ANN and decision Tree have also been applied for sorting of fruits like orange [74]. Clustering technique viz. K-Means clustering have been applied for detecting the rotten part of defective apples [75]. Different classical segmentation techniques viz. color, edge and marker segmentations generates useful results in detection of rotten portion in vegetables like tomatoes [76]. Mask generation [77] for segmentation have been an efficient way for producing desired segmented outputs. Deep learning approach introduces mask R-CNN [78] which involves two processes for segmentation–detection of component and generation of mask.

In the present research, segmenting the rotten portion of apple has been done with UNet as the backbone architecture along with some modification in the network for enhanced output. Unet finds wide range of applications in medical imaging for detection and identification functions. Segmentation of pancreas [79] for the diagnosis of cancer is one such instance of Unet's



application in medical domain. UNet with different modifications in its architecture have been adopted and came up with new names with the variations. The authors in [80] proposed UNet++ (a nested UNet), fundamentally designed keeping encoder and decoder blocks with dense skip pathways in nested fashion. RIC-UNet [81], AD-UNet [82] and FD-UNet [83] are the other modified versions of the UNet architecture designed for segmentation of nuclei in histological images, in fundus images or artifacts removal in 2DPET images. The mentioned UNet or some of the deep learning architectures have been applied universally for medical imaging analysis. However the architectures performing segmentation [84] for the identification [85] of fruits, vegetables, detection of defective vegetable or fruit and recognition of anomalies [86] in the field of horticulture, can function efficiently with profoundness. Though many segmentation algorithms have already been used in recent past but for segmentation of rotten or fresh apple, UNet or any of its modified versions is implemented for the first time for such application.

### **3.4. METHODOLOGY**

The overall methodology of the segmentation procedure is shown in Fig. 3.1. The proposed method is initialized with the required dataset [87] to the training model. The data set comprises of fresh and rotten apple RGB images. The dataset is preprocessed using image processing and are fed to the training module with a continuous monitoring on the training accuracy and loss. In the workflow diagram followed by the training module, the testing module validates the accuracy of an unknown image (fresh/rotten) and generates the required output based on the type of the input image. As the task is activated for segmenting out the rotten portion of an apple's image, hence if any rotten apple's image is fed as input, the segmented portion of the rotten part is achieved at the output of the testing module.

#### **3.4.1 Data Preprocessing**

The input RGB images are preprocessed and are converted into gray images. Further, the gray images are masked by the means of thresholding and inverse binarization as shown in Fig. 3.2. After this step the binary masks of the input RGB images are obtained. In the masked output binary image only the rotten portion is present (white color) for a rotten apple's RGB image

and the mask remains completely black for a fresh apple. The approach of generating masks has been done to produce a general dataset which reflects the ground truth of the input image distinguishing the rotten and fresh apple as shown in Fig. 3.3. The set of masks produced both for the fresh and rotten apples are fed to the training module of the automated system for segmenting out the rotten part in any input RGB image of an apple.

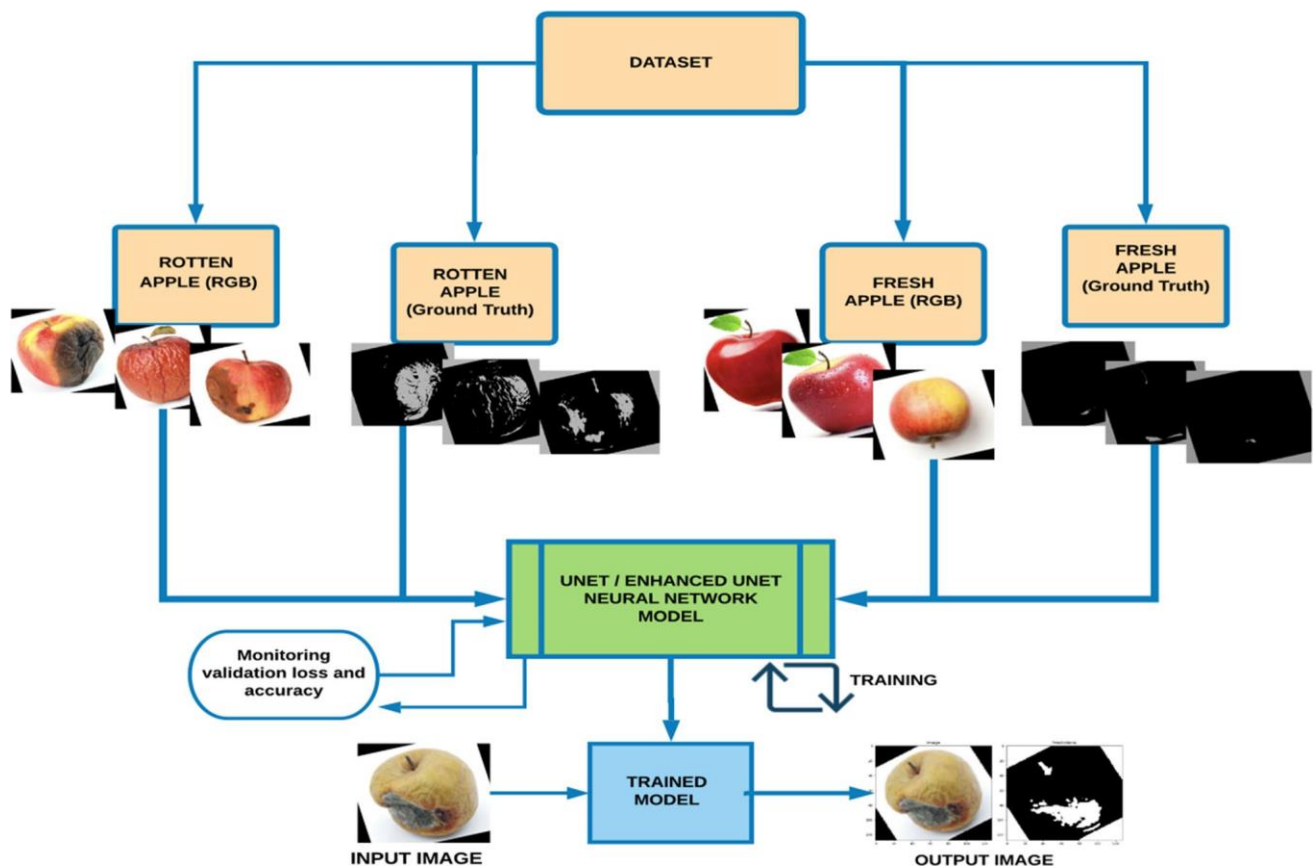
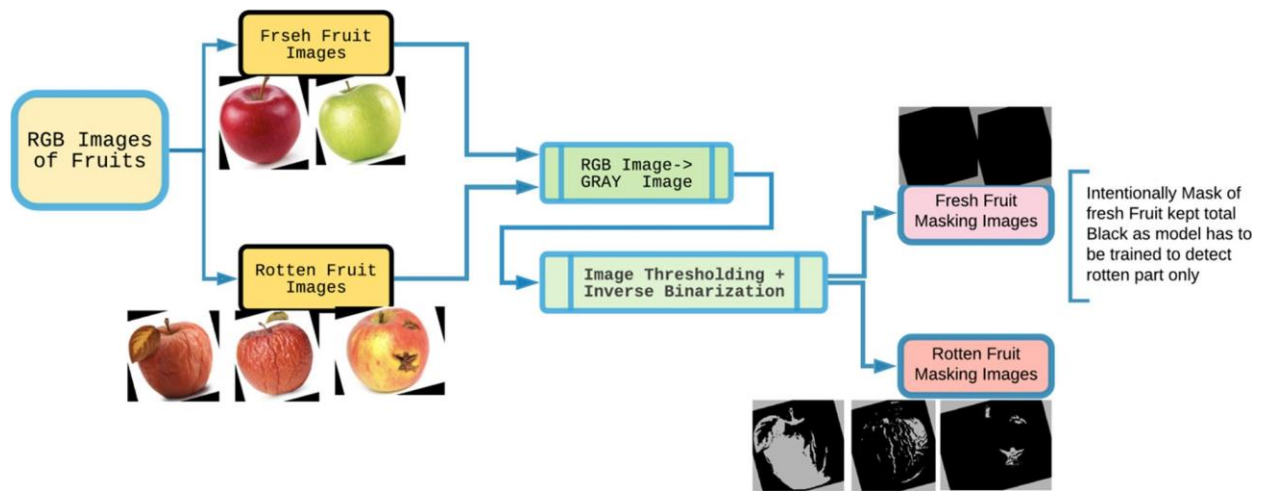


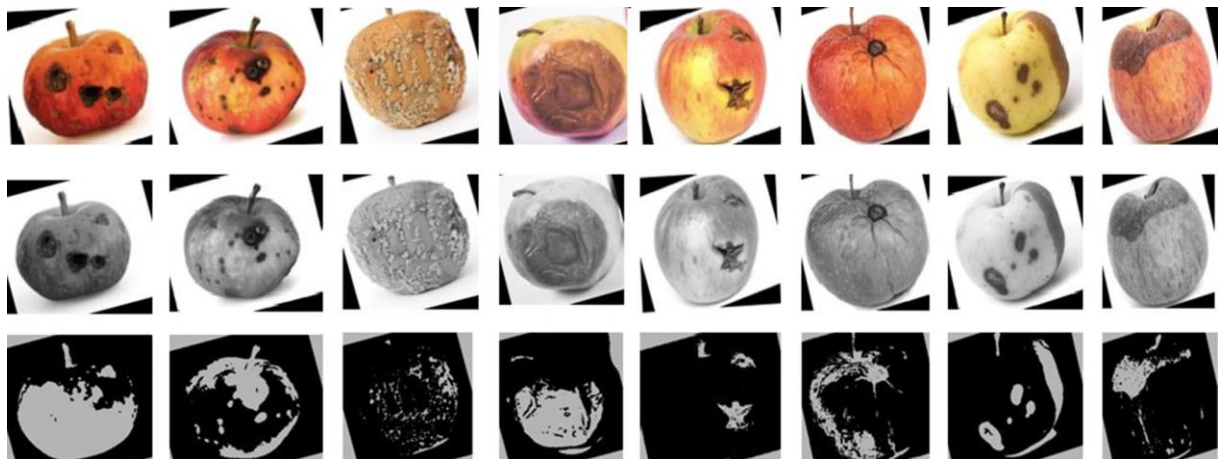
Figure 3.1 Workflow diagram of proposed methodology



**Figure 3.2 Data preprocessing and generation of ground truth images (binary masks)**

### 3.4.2 Dataset

The fruit image data set [87] is used for the segmentation purpose. The dataset contains many categories of fruits but in this case only apple has been taken for the mentioned task. The apple's data is RGB image data, containing 1693 fresh apple's image and 2342 rotten apple's image. Hence a total of 4035 number of images has been taken to train the UNet / En-UNet (Enhanced-UNet) model.



**Figure 3.3 Conversion of RGB to Gray images (Apples) and the corresponding binary masks**

## **3.5 NETWORK ARCHITECTURE**

The proposed work is compiled using UNet architecture and a modified version of the same. The modified architecture is found to generate enhanced outputs as compared to UNet and hence is named as Enhance UNet. The block diagram representation of both the networks is shown in Fig. 3.4 and Fig. 3.5 respectively. The UNet architecture is a category of fully convolutional network performing down- sampling and up-sampling. The down- and up-sampling processes are connected with a concatenation operator and hence maintain symmetry of the architecture. The UNet architecture predicts a good segmentation map combining the localization and contextual information from the sampling process. In the present chapter, segmentation has been depicted using UNet and En-UNet.

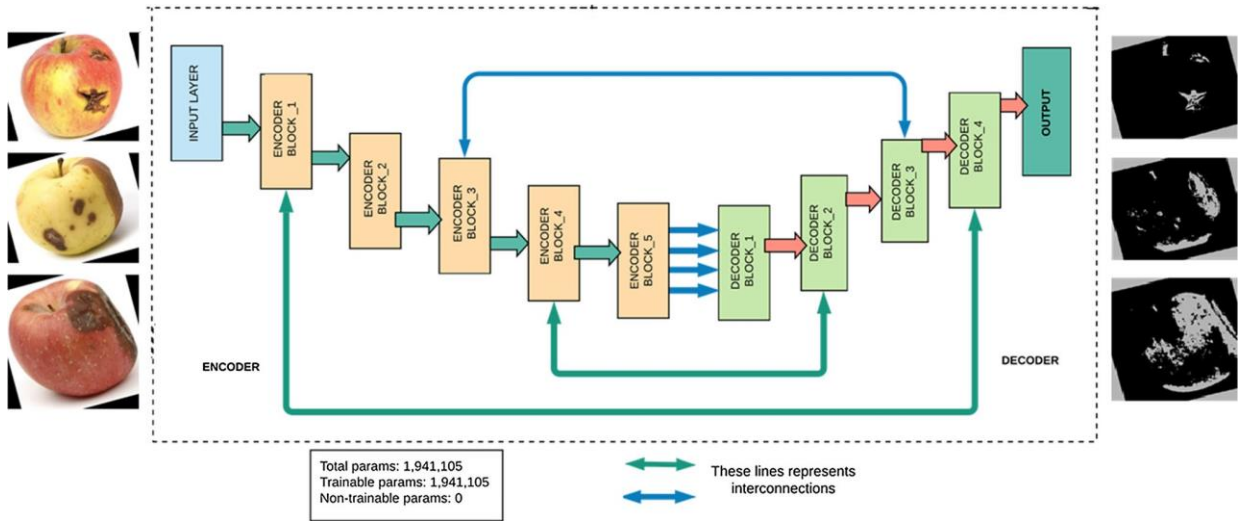
### **3.5.1 UNet**

In UNet the top-most layer being the input layer an image of size  $128 \times 128 \times 3$  is fed. For each of the encoder blocks, 2 convolution layers and 1 maxpooling layer is present and a sequential interconnection between the encoder and decoder block is retained. The decoder blocks consist of the convolution transpose layer for the expansion purpose with a concatenation layer. The maximum depth of convolution used is 128. Adam optimizer is present in the UNet architecture. Adam optimizer accelerates the search in the direction of finding out the minima, however sometimes it is missed due to the step size of the gradient. Binary cross entropy is used to calculate the loss function.

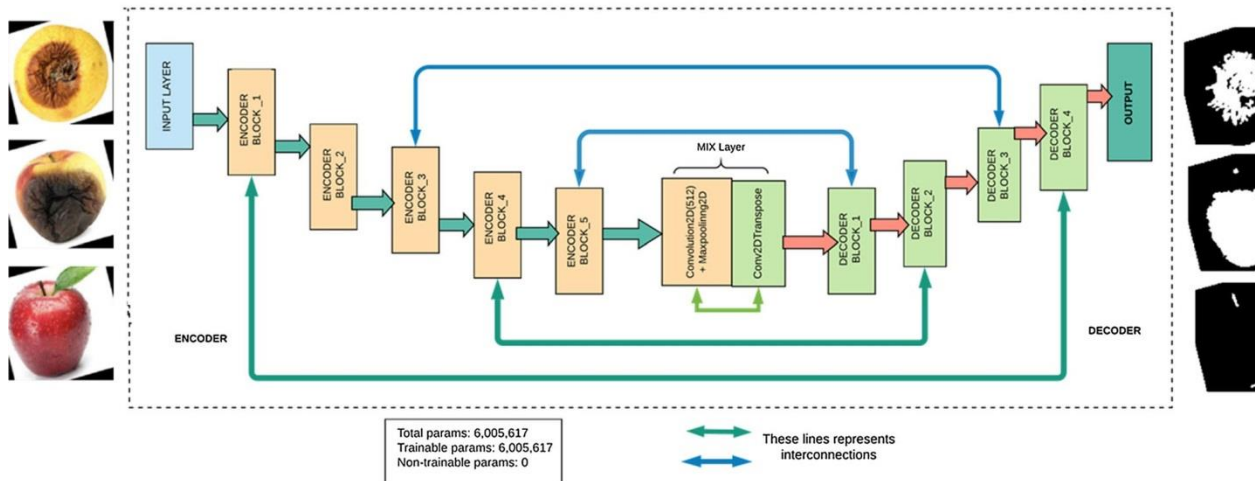
### **3.5.2 En-UNet**

In the modified architecture of the UNet named as En-UNet, each encoder block consists of 2 convolutional 2D layer, 1 dropout layer with one maxpooling layer and the decoder block encompasses 2 convolution 2D layer, 1 dropout layer and Convolution 2D transpose layer with the concatenation. Along with these layers, between the encoder and the decoder, an extra layer of 2D convolution, maxpooling, 2D convolutional transpose layer is also used in the modified architecture. Due to the addition of these extra layers in the architecture the maximum convolutional depth is increased to 512. Hence the layers present in the enhanced UNet can be

represented as: (i). Encoder: Conv 2D + Dropout + Maxpooling (ii). Decoder: Conv 2D Transpose + Concatenate + Conv2D + Dropout.



**Figure 3.4** Block diagrammatic representation of UNet



**Figure 3.5** Block diagrammatic representation of En-UNet

The encoder block 1 as shown in Fig. 3.5 consists of 2 convolution layers, each with 16 filters of kernel size  $3 \times 3$  followed by the activation function ‘elu’ (exponential linear unit). The image is then passed through a pooling layer of size  $2 \times 2$ . The procedure is repeated in encoder block 2, 3 and 4 with 32 filters, 64 filters and 128 filters each. After these encoder blocks an additional layer has been incorporated which enhanced the quality of the output more

than the general UNet. Incorporation of this layer increased the depth and thereby extracts more features in the feature map producing desired outputs. This layer is a conv 2D with 512 filters of  $3 \times 3$  kernel size and conv 2D transpose with 256 filters connected directly to the decoder. Each decoder block includes the number of filters in the decreasing order as 128, 64, 32 to 16 for the up-sampling process. Each decoder block is concatenated with the corresponding encoder block maintaining symmetry. The output has one filter and 'sigmoid' activation function. In the output, segmentation of the rotten portion present in the input RGB apple's image is obtained in white color and the output is entirely black if the input image is that of a fresh apple. The segmented output is different for fresh and rotten apple which distinguishes the task and generates the actual prediction of being fresh / rotten from the RGB input image. The proposed model is compiled using RMS-Prop optimizer and binary cross entropy as the loss function. The total number of layers present is reported in Table-3.1. Figure 3.6 portrays the type of convolutional layers with the number of filters used in segmentation task.

### **3.5.3 Training Module**

Training accuracy gives the actual prediction of the input image being rotten or fresh. In the training module two types of architectures are trained viz. UNet and En-UNet; UNet being the backbone of the modified architecture (En-UNet). In the training module 3102 images of size  $128 \times 128 \times 3$  are trained in 97 batches. Each batch includes 32 images. The training time consumed by UNet is observed to be 37 min while that of En-UNet is 46 min. The instant training accuracy and training loss per epoch have been monitored from the event log files of the Tensorflow model.

### **3.5.4 Trained Module**

The trained model is the outcome of the training module with the mentioned architectures. In the trained model any random input test image (rotten/fresh apple) can be fed and the result generated is the actual deciding predictions. The trained model also gives the model accuracy estimation performance of the segmentation task.

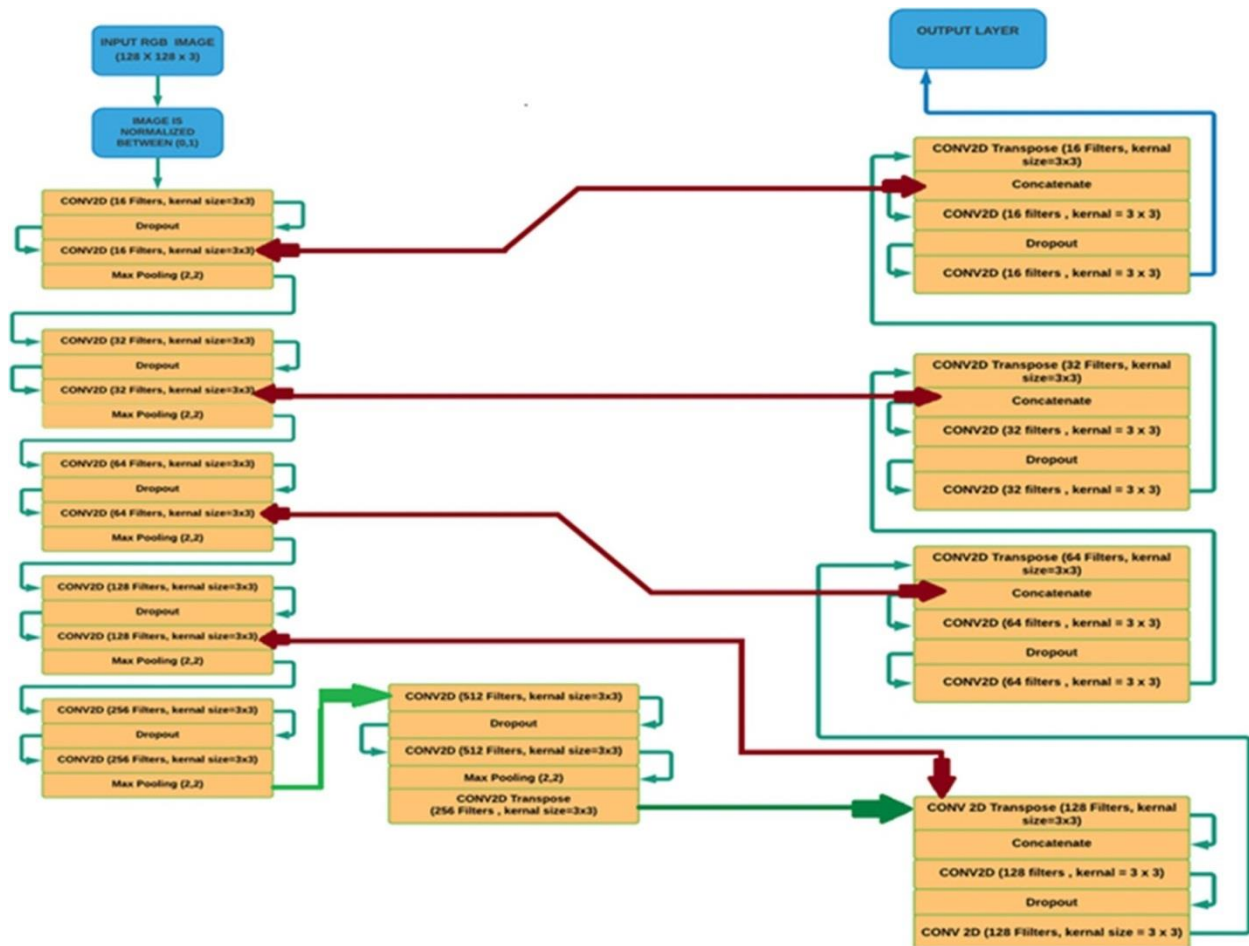


Figure 3.6 The convolutional layers of En-UNet for segmentation

Table-3.1 Type and number of layers in UNet and En-UNet

Layers architecture	2D Conv.	Drop out	2D Max Pooling	2D Conv. Transpose	Concatenation
UNet	19	9	4	4	4
Proposed En-UNet	21	11	6	5	4

### 3.6 RESULTS AND DISCUSSIONS

A comprehensive study based on simulation has been accomplished and the proposed deep learning segmentation models are validated against various performance parameters. Training of the models is done in Google Colabs with GPU Tesla K80 (2496 CUDA cores).

The deep learning segmentation architectures (UNet and En-UNet) that are executed in the present work are validated against the RGB image data, where En-UNet have shown better results than UNet. The objective behind this experiment was to explore the performance of both the models on the segmentation task. The UNet achieved training and validation accuracies of 93.19% and 95.36% respectively on segmentation of apples and encountered training and validation loss of 0.07 and 0.1086 respectively. The training and validation accuracies obtained by En-UNet are 97.46% and 97.54% respectively against training loss of 0.0657 and validation loss of 0.0618. The graphical representation of the accuracies and loss are depicted in Fig. 3.7. The predicted output should have a match of ground truth masks generated from the input RGB apple's images. The generated masks and corresponding predicted output for both the architectures are shown in Fig. 3.8. From the figure it can be observed that the predicted output finds best match with the ground truth images for En-UNet over UNet.

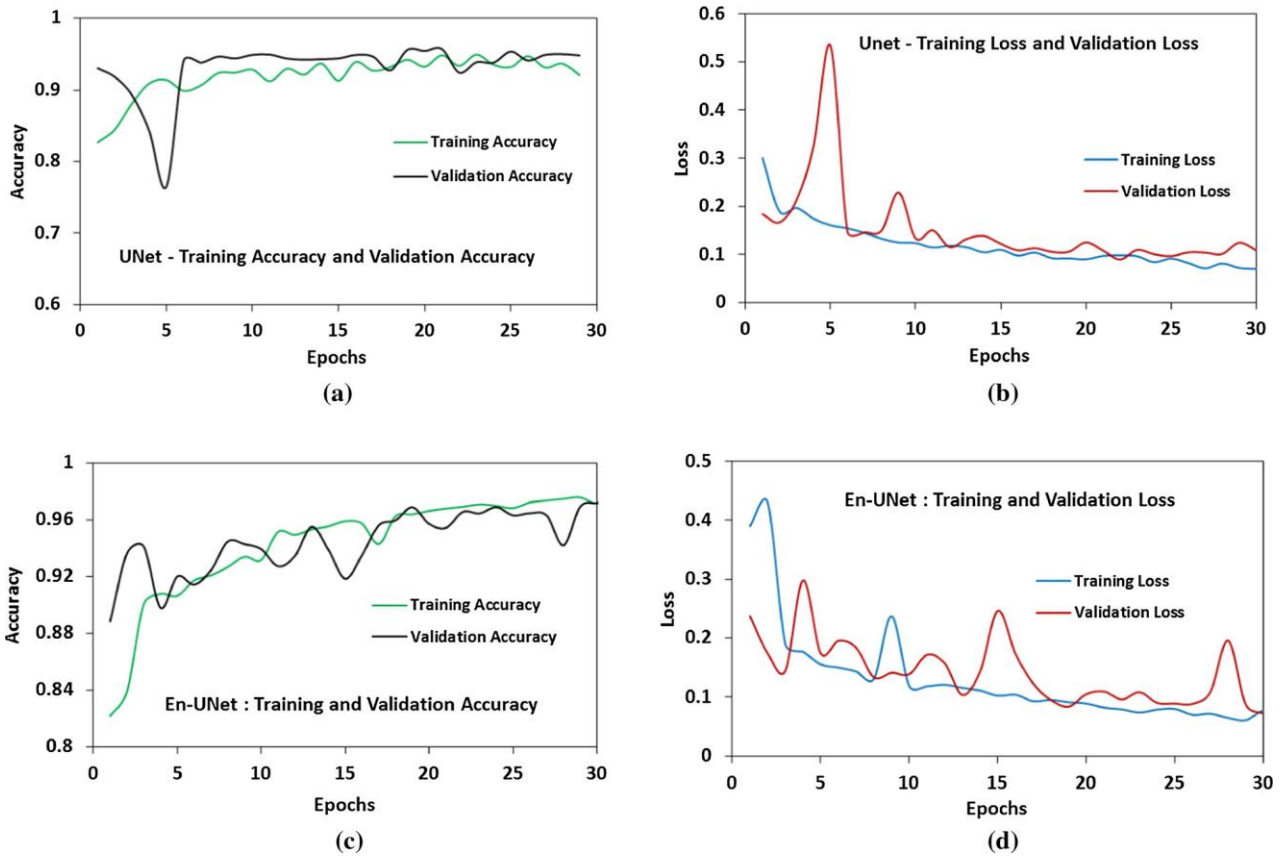
To evaluate the semantic segmentation model, IoU (Intersection Over Union) metric is very effective and commonly used. This IoU is also known as Jaccard Index. The IoU score of each class is calculated and then the mean IoU is computed to present a global IoU score on the semantic segmentation. The average IoU under a threshold of 0.95 with the input and predictions for the models used in the work is reported in Table-3.2 and the differences in the values of the mean IoU can be witnessed. The best IoU score achieved is 0.866 with En-UNet and 0.66 with that of UNet.

The segmentation results obtained from UNet and En- UNet are represented in Fig. 3.9 for a comparable visualization. Better results are obtained from the En-UNet model which can also be seen from the figure. The segmented outputs for both rotten and fresh apple are displayed. For rotten apple images the segmentation got enhanced in case of En-UNet segmentation and every region under rotten portion is present in the output image.

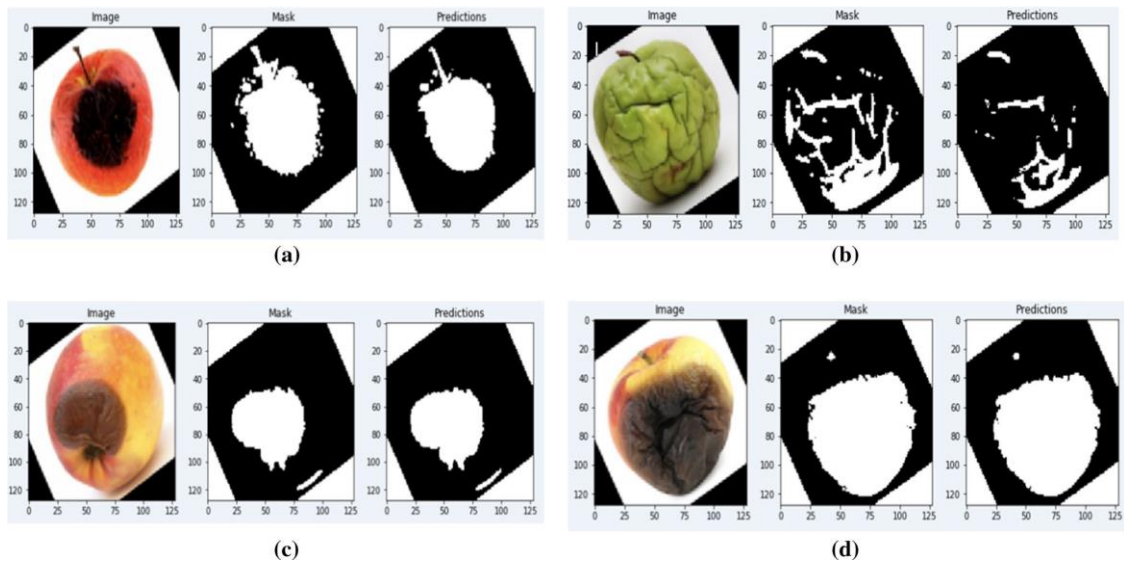
In UNet, segmentation brings out lesser portion of rotten areas as compared to En-UNet. Moreover in case of fresh apples the result shows that UNet has segmented out some portion in the fresh apple owing to over segmentation whereas En-UNet has done justice to the task of segmentation and produced nearly the same output as the ground truth (black) for fresh apples. The total trainable parameters for UNet are 1,941,105 whereas those for En- UNet are 6,005,617. The optimizer RMS Prop used for En- UNet also facilitated the architecture for



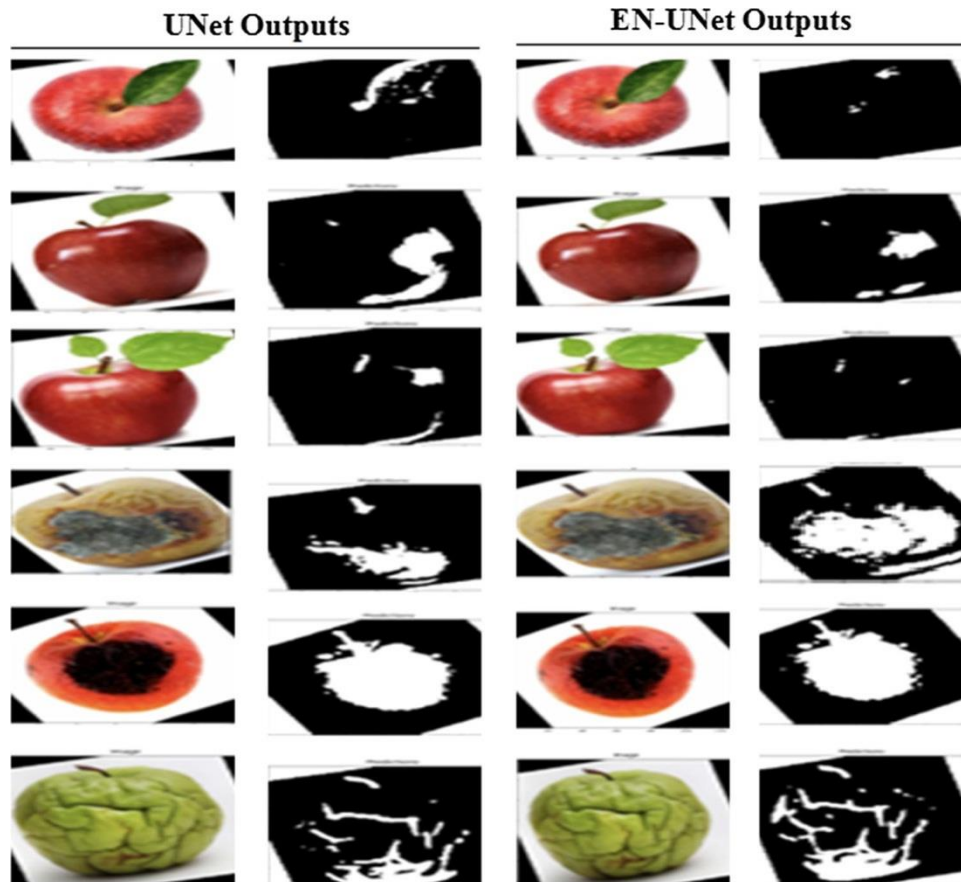
perfect finding of the minima.



**Figure 3.7 Accuracy and loss plot - (a) training and validation accuracy of UNet, (b) training and validation loss of UNet, (c) training and validation accuracy of En-UNet, (d) training and validation loss of En-UNet**



**Figure 3.8 RGB image mask and predicted output (a–b) UNet, (c–d) En-UNet**





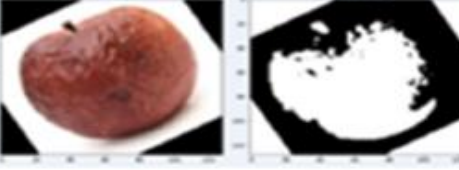
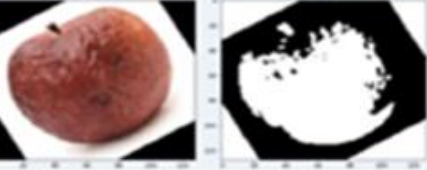
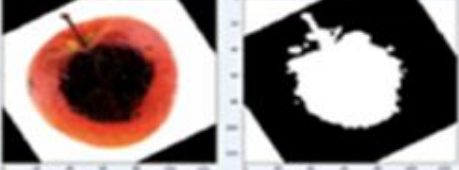

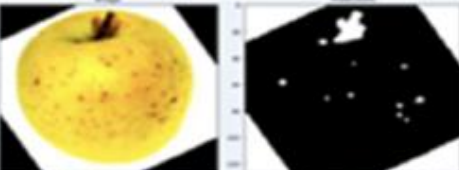
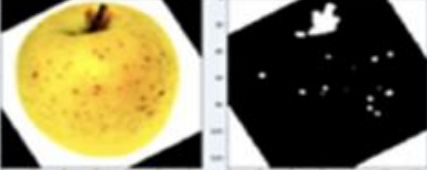
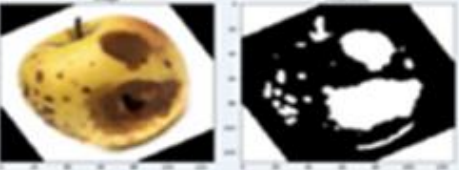
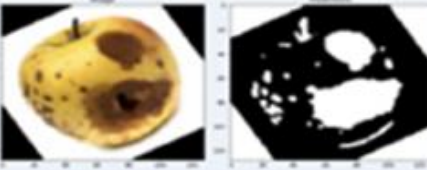
**Figure 3.9 Segmented output images of UNet and En-UNet**

### 3.7 CONCLUSION

Real time semantic segmentation of rotten apples is carried out leading to categorization of fresh apples from the rotten ones. The RGB images of the rotten and fresh apples are fed to two architectures viz. UNet and En-UNet. Before feeding the image data to the networks, binary masks as ground truths are generated and then the predicted outputs are validated against the ground truths. Comparisons of both the models are carried out in terms of accuracy and IoU score. From the results, UNet achieved validation accuracy of 95.36% whereas En-UNet achieved 97.54% validation accuracy. The best mean IoU score under a threshold of 0.95 attained by En-UNet is 0.866 and that of UNet is 0.66. The ability of the proposed work can be very much useful in the horticulture domain as well as fruit industries with fast automation system developed for good quality fruit detection and production of quality food products. The

present research work will definitely help the researchers with new ideas for building sophisticated automation system for smart agriculture and horticulture based industries.

**Table-3.2 Semantic segmentation with UNet and En-UNet and performance parameter – Mean IoU for different output Images (A stands for UNet and B stands for En-UNet)**

Input Image with predicted Output (UNet)	Input Image with predicted Output (En-UNet)	IoU <sub>A</sub>	IoU <sub>B</sub>
		0.66	0.86
		0.33	0.35
		0.33	0.45
		0.21	0.27
		0.45	0.50

## Chapter-4

---

# IMPLEMENTATION OF DEEP LEARNING ARCHITECTURES FOR CLASSIFICATION AND LOCALIZATION OF DIGITAL IMAGES

## Chapter-4 IMPLEMENTATION OF DEEP LEARNING ARCHITECTURES FOR CLASSIFICATION AND LOCALIZATION OF DIGITAL IMAGES

---

### **4.1. INTRODUCTION TO CLASSIFICATION AND LOCALIZATION OF DIGITAL IMAGES**

In computer vision, the two-step classification and localization approach is used to recognize and locate objects in an image. It is frequently used in tasks involving object detection, where the objective is to identify the precise location of objects in an image by drawing bounding boxes around them, in addition to classifying the objects that are present in the image.

The entire process of classification followed by localization is also known as detection. Finding and identifying numerous objects of interest within an image or a video is the process of object detection. Deep learning algorithms have been found to be robust and efficient in the detection process used in computer vision. It has revolutionized computer vision tasks, including object detection, owing to its ability to automatically learn hierarchical features from raw data. The key concepts and techniques involved in the detection process include classification and localization with the resultant output as detection. The deep learning algorithms with different sophisticated models have proved to be the best in detection procedure with efficacy. The algorithm is first trained to divide the entire image into various classes or categories. The classifier's job, for instance, would be to identify which class each image in a dataset of images of cars, bicycles, and pedestrians belongs to. Convolutional neural networks and other deep learning architectures, which are proficient at learning to extract features and make class predictions, are frequently used to accomplish this. Localizing the objects in the image is done in the second step after classification is complete. Finding the bounding boxes that enclose the objects of interest is the process of localization. It can be represented as a collection of four values for each object, typically written as (x, y, width, height), where (x, y) stands for the top-left corner coordinates of the bounding box and width and height stand for the box's dimensions.

An object detection system is created by combining the classification and localization steps. The system can recognize multiple objects in an image, classify them according to their respective functions, and provide bounding boxes around each object to show where it is in relation to other objects. A labeled dataset is necessary to train a classification followed by localization model. Images, class labels, and bounding box annotations for each object should all be included in this dataset. The model gains the ability to simultaneously categorize the objects and forecast their bounding boxes based on the provided annotations during training.

Although object detection is a difficult task, it has many uses in a variety of industries, including robotics, autonomous vehicles, and surveillance. To increase the precision and effectiveness of object detection systems, researchers have created a number of methodologies and architectures, including Single Shot Multibox Detector (SSD), You Only Look Once (YOLO), and Region-based CNN. R-CNN and Faster R-CNN are the state-of-the-art algorithms for the detection purpose. The development of R-CNN significantly improved the object detection process. It separated the region proposal and classification phases of the detection process into two steps. To begin, potential object regions were generated using selective search or other methods for region proposal. After that, CNNs were used to categorize and improve the bounding boxes of these suggested regions. R-CNN produced good accuracy but was slow because each region had to be processed separately. A region proposal network was introduced by Faster R-CNN and integrated into the object detection architecture. By directly generating region proposals from CNN features, the RPN makes the entire process trainable. Accuracy and speed were both increased by faster R-CNN as compared to R-CNN.

In the present work, the localization of the objects is accomplished with the classifier as the backend architecture. Classification being the primary task is very important in localization. The classifiers are fine-tuned and customized with alteration of the various hyper-parameters for the best result generation. Three case studies have been carried out in the present research and the architectures are modeled in a new and novel methodology. For better and fast performance of the models, new models are designed in a hybridized manner with inclusion of traditional machine learning techniques. The hybrid models with customized classifiers for detection has proved to out-perform the state-of-the art models depicted in the result sections of each case

study. For each of the case studies, a comparative analysis has also been reported in the comparison tables.

## **4.2. IMPORTANCE OF AI / ML IN MARITIME SURVEILLANCE**

The booming shipping industry with incredible rise in its number of ships and cargoes has led to prolific growth in maritime transportation accounting for 90% of international trade. It not only supports domestic and global manufacturing companies through transportation of their commodities and products; it also helps in delivering goods directly to consumers. Shipping is the lifeline of worldwide economy for export / import of various raw items and consumer products and thus special care needs to be adopted for global maritime safety. Even though maritime transportation has been a dominant support of global trade, its advancement also leads to traffic violations in the waterways and thus in spite of the growing traffic, the access to major trade commodities remains solely the main driving point in setting the maritime networks. The maritime trade has truly been struck in recent years owing to detection of illegal activities involving maritime terrorism, smuggling and sea-jacking [88]. These menaces put threat to global, regional and national economies; which if not curbed would lead to catastrophic disaster. Hence pledge for safe navigation of sea vessels and safety of sea activities demands surveillance of ocean ships in coastal countries [89-90]. It is herein that the automatic ship detection becomes utmost necessity for maritime security management and surveillance [91]. The primary functions of it are traffic flow monitoring, prevention of marine pollution and detection of illegal fishing and cargo transportation. Intelligent detection of ships by means of automation and computer vision makes the task easier and flexible. Sometimes, infiltration of the intruders through waterways causes big threat to the nation's security. Here, movement of water vessels in irregular pathways is an identification of abnormality in the oceanic perimeter. Thus, computer aided detection system will be efficient in identification of such anomalies. In case of military, automatic ship detection helps in enhancement of maritime security through Intelligence, Surveillance and Reconnaissance (ISR) efforts [92]. A wide range of functionalities in the shipping industry encompasses the defense of territory and naval battles, dynamic harbor surveillance, monitoring of traffic and sea pollution, management of fisheries, etc. [93]. ISR involves development of highly sophisticated sensor systems required for collection of

increased volumes of heterogeneous data. Some typical sensors on-board an ISR enabled maritime patrol aircraft includes Electro-Optical/Infrared (EO/IR) camera, radar, Electronic Support Measures (ESM), etc. They facilitate capturing of environmental, individual and conventional signatures and subsequently generate large volumes of collected data.

The critical job involving maritime security and civil management is typically accomplished with the help of Automated Identification System (AIS), which employs radio frequencies in the VHF band to wirelessly broadcast the location of the ships to the nearby receivers on other ships and land based systems [94]. The AIS is effective only when connected with transponder device on the ships; however the functionality ruptures if the transponder is disconnected or not installed on the ships. Satellite imagery under this condition is very helpful in detection and identification of the ships in the water bodies with the aid of machine learning algorithms.

The promising technology for collection of ship / sea vessels related necessary data through high resolution images obtained using satellites and aerial remote sensing devices [95] are recent topics of worldwide research. Moreover during cyclones, large number of ships in the ocean sinks due to the fact that they remain cut-off from the land based systems. Hence, automatic ship identification with satellite images can assist in finding and rescuing the ships in the heavy cyclones; thereby saving many lives. The detection of ships from the high resolution remote sensing images sometimes become tedious owing to the disturbances like clouds, islands, mist, haze, coastlines, tides, etc. However, optical remote sensing images for ship detection are very popular these days owing to a plethora of applications in defense and civil domains. This satellite imagery can provide real-time position information for navigation management control and maritime search and rescue operation ensuring success and work safety at sea and on inland rivers. Additionally, it contributes to the administration and development of important coastal zones and harbors; endorsing ecological protection and sea health.

Ship detection through remote sensing has gained lots of importance owing to the availability of high-resolution Synthetic Aperture Radar (SAR) images suitable for object detection and environment monitoring. These images are capable of providing high-resolution images of the oceans both during day as well as during night. It has proved to be an effective technology for



detection of ships and a hot research area throughout the globe [96-97]. These images are weather independent and are most suitable for monitoring maritime activities like ship detection and oil spills. The quad-polarization SAR (QP SAR) mode, amongst the different available polarimetric SAR modes, is found to capture the richest information of the observed area [98]. However, the linear dual-polarization SAR mode having lower system complexity supports a wider swath width [98] while the compact polarimetric SAR (CP SAR) mode provides a compromise between swath width and scattering information [99]. In [100], the authors have developed a novel algorithm for ships detection using low resolution SAR imagery for ships greater than 35 m length and using high resolution SAR imagery for ships greater than 32 m length. In [101], a segmentation method from CP SAR images is proposed for detection of ships in which pixel-wise detection is based on a fully convolutional network, U-Net. In [102], an enhanced GPU based deep learning method for ship detection using SAR images is reported. The authors proposed a modified deep learning framework called You Only Look Once version 2 (YOLOv2) to model the architecture and train it. Moreover a new architecture with less number of layers called YOLOv2-reduced is also developed. The results exhibited better accuracy in ship detection with appreciable reduction in computational time. In [103], an absolute new detection method is reported to differentiate sea vessels from complex backgrounds from SAR image using proposed local contrast variance weighted information entropy (LCVWIE).

The unique and state-of-the-art performance for detection of ships using optical remote sensing is reported in literature [104-105]. In [104], the authors proposed a fast and robust ship detection algorithm based on deep CNNs. Here, initially deep CNN is used for feature extraction and subsequently a RPN is applied for discrimination of ship targets. In [105], a Rotation Dense Feature Pyramid Networks (R-DFPN) framework was proposed which was found to detect ships effectively in different scenes including ocean and port. The surveillance video system is also reported to be used for ship detection [106-107]. In [106], the authors applied dynamic fusion technique on background subtraction (BS) and saliency detection (SD) technique results for final boat detection from maritime surveillance videos while in [107], the authors used a vision based autonomous landing algorithm for the same. For smart monitoring concerning effective utilization of port resources, detection and recognition of ships are very

vital aspects. However it imposes serious challenges owing to several issues like complex ship profiles, ship background, object occlusion, variations of weather, light conditions, etc. In [108], the authors propose an on-site processing approach called Embedded Ship Detection and Recognition using Deep Learning (ESDR-DL), in which the video stream is processed using embedded devices for accurate ship detection and recognition. In [109], an object detection system based on Histogram of Oriented Gradients (HOG) is proposed for finding ships in maritime videos. The work is further extended in [110], in which the author applied HOG-SVM (Support Vector Machine) detector to a ship detection system for quantitative evaluation. The results were very promising for ships with resolutions larger than  $128 \times 64$ . In [111], a novel multi-level ship detection algorithm is proposed for different offshore ships detection under all possible imaging variations using Multi-Scale Analysis and Fourier HOG Descriptor. In [112], a FCN with task partitioning for inshore ship detection in optical remote sensing images is reported. In [113], the authors proposed novel hierarchical complete and operational ship detection from space-borne optical images (SDSOI) approach based on shape and texture features, which is considered a sequential coarse-to-fine elimination process of false alarms.

The rapid and continuous increase in the hardware computing power has resulted in the faster development of deep learning based algorithms for object detection. Deep learning algorithms have been extensively adopted for remote sensing image analysis. In [114], the authors proposed a novel detection algorithm called region-based deep forest (RDF) comprising of a deep forest ensemble with a simple region proposal network in order to discriminate ship targets. In [115], a CNN based novel method is proposed which starts from a global search for the relatively distinct ship head with an efficient classification network for inshore ship detection. In [116], an algorithm has been developed by the authors for ship detection by combining CNN with constant false alarm rate (CFAR) in which the proposed CNN is based upon the CFAR global detection algorithm and image recognition with CNN model. In [117], an effective novel rotation-invariant CNN (RICNN) model is proposed for detection of objects through introduction of a new rotation-invariant layer. In [118], determination of presence of ships or not from aerial image of visible spectrum is presented in which neural codes extracted from CNN is combined with k-Nearest Neighbor method to improve performance. In [119], a novel deep feature-based method is developed in order to detect the presence of ships in very

high-resolution optical remote sensing images. The authors have used a regional proposal network in order to generate ship candidates from feature maps produced by deep CNN. In [120], a CNN based effective ship detection framework in remote-sensing images is proposed wherein the framework is so designed so as to predict bounding box of ship with orientation angle information in complex remote-sensing scenes. In [121], the authors proposed a saliency-aware CNN for ship detection using real time captured visual images. The proposed CNN framework consisted of exhaustive ship discriminative features like deep feature, saliency map and coastline prior. The authors in [122], with the help of deep networks, put forward a fast R-CNN method for ship detection from high-resolution remote sensing imagery. This method proved to be an effective one for offshore and inland river ship detection from high-resolution remote sensing imagery. In [123], a ship detection and segmentation method was proposed based on an improved Mask R-CNN model capable of accurate detection and segmentation of ships up to the pixel level. The proposed model could successfully improve the feature maps in describing the features of the target. In [124], the authors developed a CNN based novel method for SAR image change detection. Here the prime idea focused on producing the classification results directly from the original two SAR images through CNN without the use of any pre-processing operations. The developed algorithm was found to be robust in detecting ship targets under complex conditions such as wave clutter background, target in close proximity, ship close to the shore and multi-scale varieties. Even when applied to the change detection of heterogeneous images, the algorithm yielded satisfactory results. In [125], a new method of detection of ships for the land contained sea area is proposed. Here an island filter is used to minimize the existing false alarms from the island; use of CFAR is done to get candidates from the big map and finally CNN based classifier is used to separate false alarms from ship object. A new pooling called max-mean pooling is introduced in order to extract effective features in CNN flow. In addition to the above methods adopted in [125], a further extension is carried through use of threshold segmentation for quick execution of candidate detection [126]. Thereafter a two-layer lightweight CNN model-based classifier is designed to separate false alarms from ship objects at last visualization is achieved through ship prediction in vertical–horizontal (VH) and vertical–vertical (VV) polarization. The model is found to perform with great accuracy for ship image with size less than  $32 \times 32$ . For small ship detection and to ensure autonomous ship safety, the authors in [127] proposed a novel hybrid deep learning method by

combining CNN with a modified Generative Adversarial Network (GAN). In [128], the authors proposed a novel ship detection method from remote sensing images based on multi-layer convolutional feature fusion (CFF-SDN). In the paper [129], the authors proposed a deep learning approach for detecting ships in harbour areas using DenseNet architecture as core CNN based classifier.

In this chapter, the detection of ships is done with deep convolution neural networks. The work has been accomplished primarily by classification of the ships present in each image and thereby localizing its presence in the main frame. The classification of ‘Ship’ and ‘No-Ship’ are achieved using a 4-layer 2D CNN and the localization is attained by a bounding box at the specified co-ordinates. Followed by classification, the category of images only with ships is segmented by implementation of an auto-encoder with a pre-processed dataset. Hence the proposed model is a multi-neural network based framework and the results obtained are very satisfactory in terms of parametric studies.

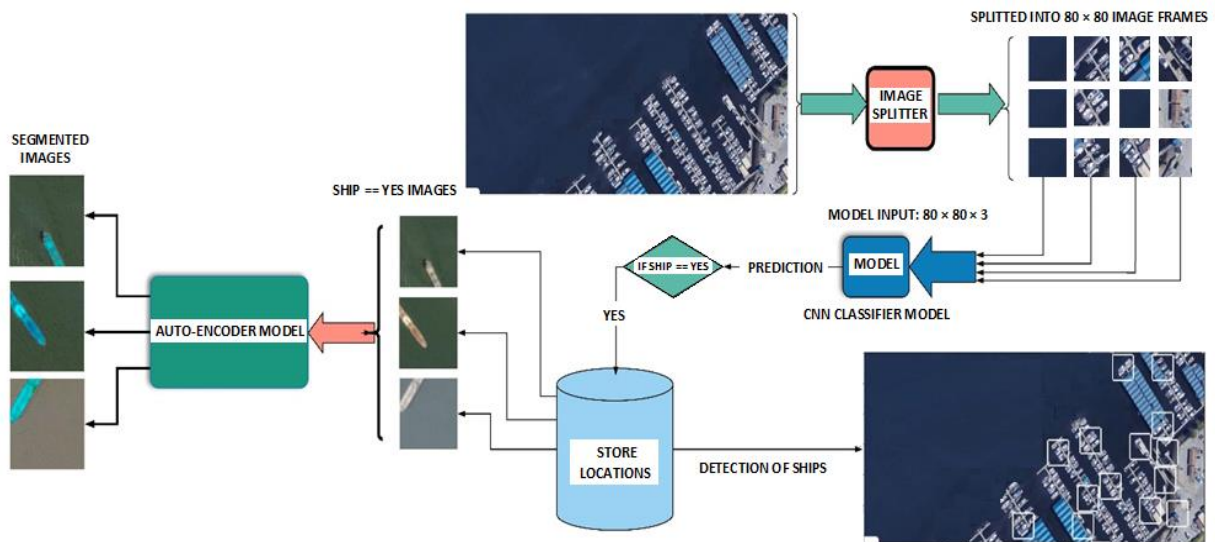
### **4.3. METHODOLOGY**

In this chapter, the detection of ships is accomplished with visible imagery satellite dataset ‘Ships in Satellite Imagery’ [130] consisting of 4000 images. Here a multi-neural network based framework i.e., classification as well as segmentation of the images is presented, so dataset preparation is carried out for the mentioned two tasks. The overall workflow method is depicted in Figure 4.1. The labeled dataset comprising ship and no ship images are fed to the CNN classifier training model; the model gets trained generating training accuracy and loss. Followed by the training module, a test image is provided which is normally bigger in size than the training dataset. The big size image is then splitted into multiple frames of the size same as training images and are fed to the model. The model detects the presence or absence of ships in these frames and the location of the frames in the original image is stored in an array. The locations (pixel co-ordinates) are then matched in the original big size image and in the matched location if ship is present, a bounding box is activated for marking the detection of ships. On the other hand, the classified images with the presence of ships are fed to an auto-encoder where segmentation is

performed with some pre-processing of the images. The segmented outputs of the ships are marked in blue colour.

### 4.3.1 Dataset

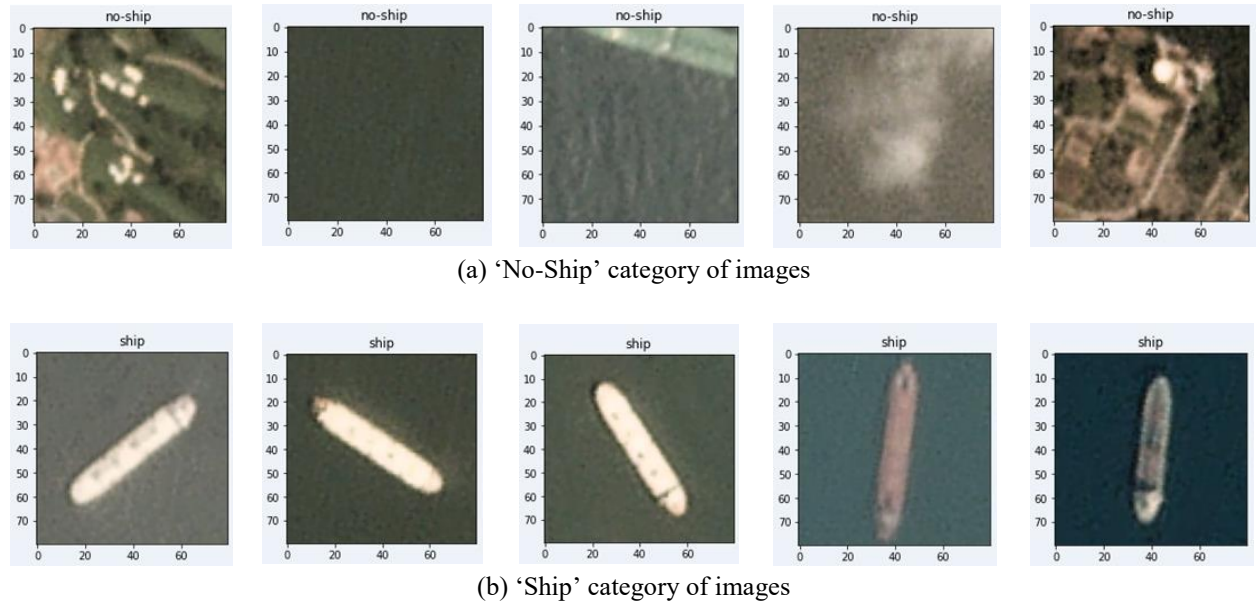
The dataset is taken from ‘Ships in Satellite Imagery’ [130] available in Kaggle. These satellite captured imagery with the help of a constellation of small satellites are made available through Planet, a new commercial imagery provider. The dataset is extracted from Planet satellite imagery collected over the San Francisco Bay and San Pedro Bay areas of California. It consists of total 4000,  $80 \times 80 \times 3$  RGB labeled, images as ‘Ship’ and ‘No-Ship’ categories. The category with ‘Ship’ contains 1000 images and that of ‘No-Ship’ contains 3000 images.



**Figure 4.1** The workflow diagram of the proposed methodology

The images under ‘Ship’ class are all near-centered on a single ship body. Inclusions of different types of ships in their various sizes, orientation and under different atmospheric conditions are also present in this dataset. However, for ‘No-Ship’ class, the dataset comprises of (a). images not having any ship portions but having different land cover features like water, vegetation, bare earth, buildings, etc., (b). images covering certain ship portions only i.e., partial-ships, and (c). images mislabeled owing to bright pixels or strong linear features. Further, for the efficient working of the CNN model, more data are generated from the available dataset with the use of

data augmentation technique. 16000 total images are considered for both classification and segmentation purposes. 8000 images generated with ‘Ship’ and 4800 with ‘No-Ship’ are taken into account for training purpose while the rest 3200 images comprising of a mixture of ship, no-ship and some land cover areas are used for validation of the CNN model. Some typical dataset sample images are exhibited in Figure 4.2.

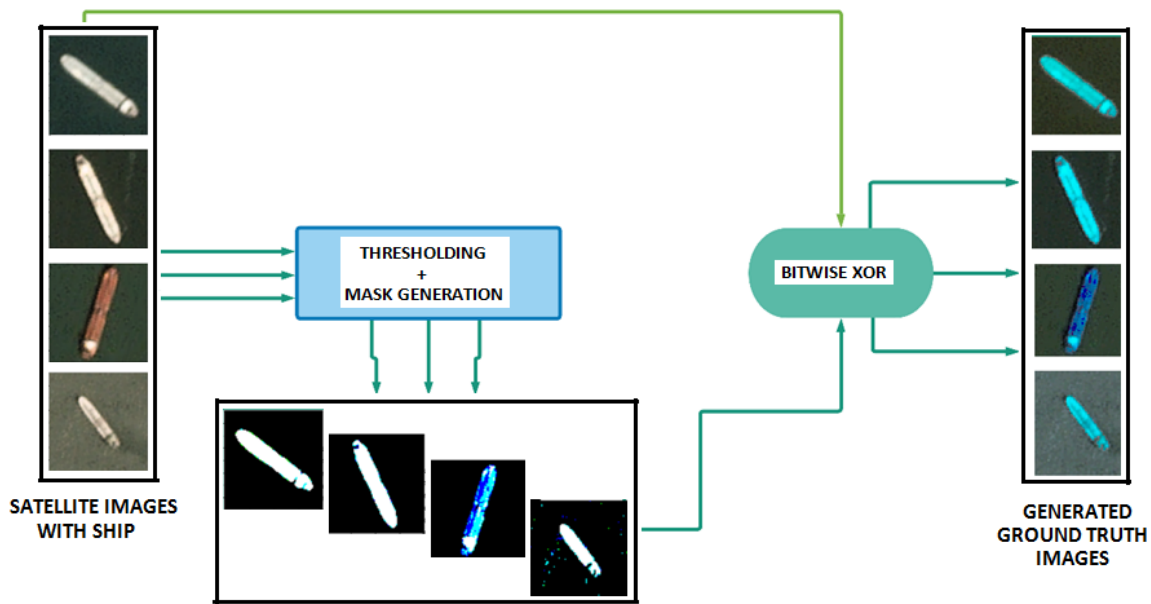


**Figure 4.2** Typical dataset sample images for ships and no-ships

### 4.3.2 Data Pre-processing

The data pre-processing is undergone in two steps for the two tasks – classification and segmentation. For classification purpose, the RGB satellite images are augmented using different operations like shear, rotate and stretch generating 12800 images for training and the remaining 3200 images for validation. The two classes labeled images are shown in Figure 4.2. The RGB images before being fetched by the auto-encoder undergo a processing step as shown in Figure 4.3. The RGB images classified as ships [Figure 4.4(a)] are first converted to grey images and then by means of thresholding and binarization, masks are produced [131] as shown in Figure 4.4(b). Followed by masking, the original ship images and the masks undergo bit XOR operation. After the XOR operation, the images containing ships are marked in blue colour while the images with no-ship remains as it are. These images act as the ground truth data [Figure

4.4(c)] for the auto-encoder. The set of these ground truth images are fed to the training module of the auto-encoder system for segmentation tasks.



**Figure 4.3** Data preprocessing and generation of ground truth images

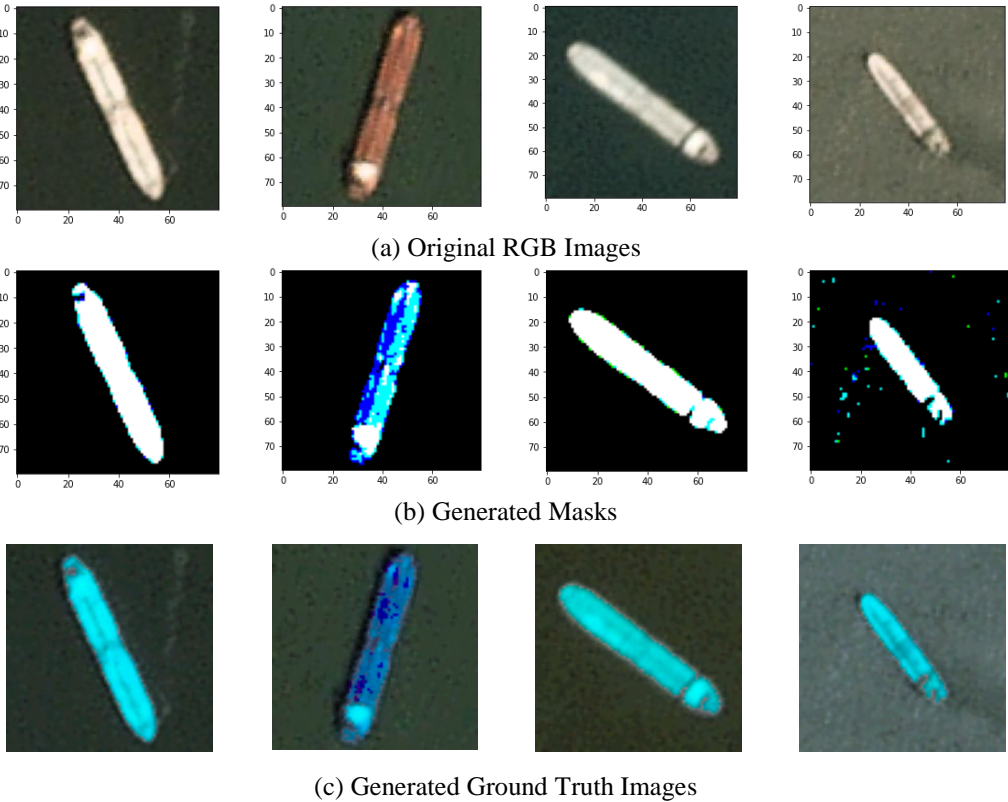
### 4.3.3 Network Architecture

The present work on detection of ships is accomplished using two architectural models – (a). 4-layer CNN for classification purpose and (b). Auto-encoder for segmentation purpose. The CNN used here acts as a binary classifier producing two classes of outputs as ship and no-ship.

#### A. 4-layer CNN for Classification

A 4-layer CNN is implemented for binary classification comprising of CNN 2D, Maxpool 2D and dropout in each layer as shown in Figure 4.5. The output layer comprises of the flattening and the dense layer generating the classified outputs. The CNN classifier has been trained with a typical dataset [130]. The dataset has been augmented and prepared for training and validation. The classifier is trained with the augmented dataset and for the testing procedure; a new test input image is fed to the model. The size of the test input image is not the same as that of the training dataset of  $(80 \times 80 \times 3)$  size and hence the big size test input image is splitted into

multiple grids of size  $80 \times 80 \times 3$ . The portions of the big sized image are now fed to the system model and the system verifies for the presence or absence of ships in the particular grid. If the classification result comes true then the particular location of the grid in the original frame is stored in an array. Followed by accumulation of the co-ordinates (the grids), the localization of ships in the original image is accomplished by bounding boxes in the locations (stored in the array) of the original test image. The bounding boxes established at different points in the image shows the presence of any ship or vessel in the water body. The network architecture of the CNN classifier is illustrated in Table-4.1.

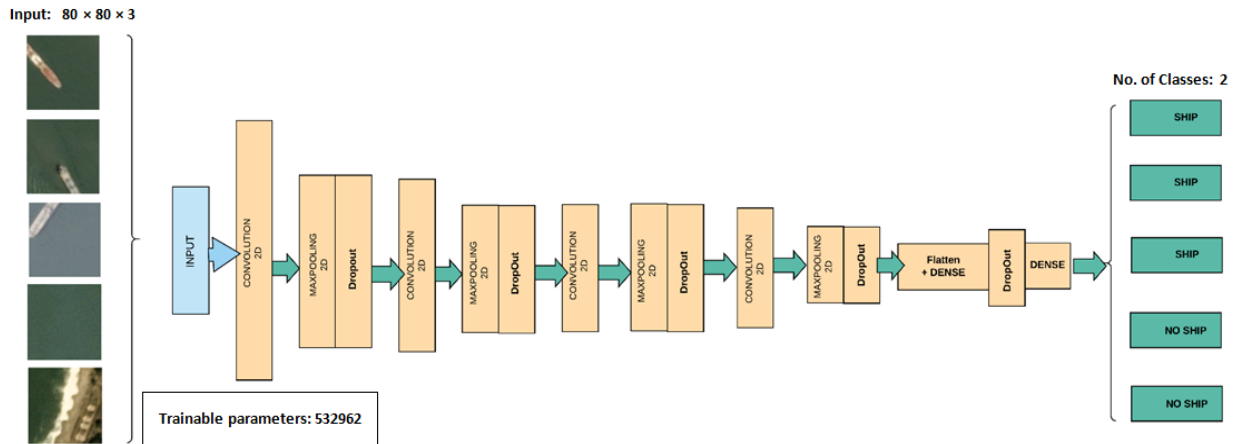


**Figure 4.4** Created dataset: Conversion of RGB (original images) to corresponding binary masks and ground truth images



**Table-4.1 The architecture of 4-layer 2D CNN**

Layer (type)	Output Shape	Parameter Num.
conv2d_10 (Conv2D)	(None, 80, 80, 32)	896
max_pooling2d_6 (MaxPooling2D)	(None, 40, 40, 32)	0
dropout_5 (Dropout)	(None, 40, 40, 32)	0
conv2d_11 (Conv2D)	(None, 40, 40, 32)	9248
max_pooling2d_7 (MaxPooling2D)	(None, 20, 20, 32)	0
dropout_6 (Dropout)	(None, 20, 20, 32)	0
conv2d_12 (Conv2D)	(None, 20, 20, 32)	9248
max_pooling2d_8 (MaxPooling2D)	(None, 10, 10, 32)	0
dropout_7 (Dropout)	(None, 10, 10, 32)	0
conv2d_13 (Conv2D)	(None, 10, 10, 32)	102432
max_pooling2d_9 (MaxPooling2D)	(None, 5, 5, 32)	0
dropout_8 (Dropout)	(None, 5, 5, 32)	0
flatten_1 (Flatten)	(None, 800)	0
dense_2 (Dense)	(None, 512)	410112
dropout_9 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 2)	1026
Total parameters: 532962, Trainable parameters: 532962, Non-trainable parameters: 0		

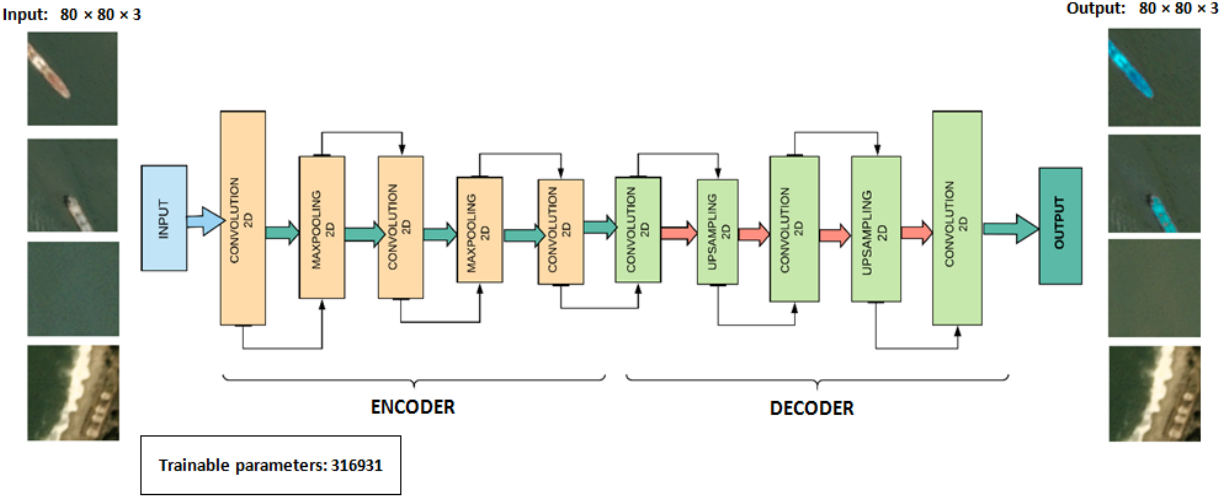


**Figure 4.5 Block diagrammatic representation of 4-layers CNN**

## B. Auto-encoder

A very useful property which the auto-encoder possesses is the automatic learning from the data examples. Auto-encoders do not require any innovative engineering but requires an appropriate training data. The auto-encoder undergoes mainly the compression and decompression functions

implemented by the neural networks. It performs segmentation tasks effectively and shows promising results. In this chapter, along with classification of ships, the semantic segmentation in ships is also carried out using auto-encoder model. In semantic segmentation, different clusters of pixels are assigned to a specific class. A processed dataset has been created for training of the auto-encoder as shown in Figure 4.4. Creation of such masked dataset is of great importance while undergoing semantic segmentation task with auto-encoders. The images which have been classified in the earlier step as ‘ship’ are now segmented implementing the auto-encoder for a detailed representation. The auto-encoder in general includes the encoder and the decoder blocks. The encoder after fetching the input image data maps the input data into compressed data or latent space where similar points remain close together. On the contrary, the decoder blocks maps those data points to target size output. The decoder block performs the up-sampling of the data points at different layers and reaches to the softmax layer. The working model output is shown in Figure 4.6. The network architecture of the auto-encoder is illustrated in Table-4.2.



**Figure 4.6** Block diagrammatic representation of Auto-encoder

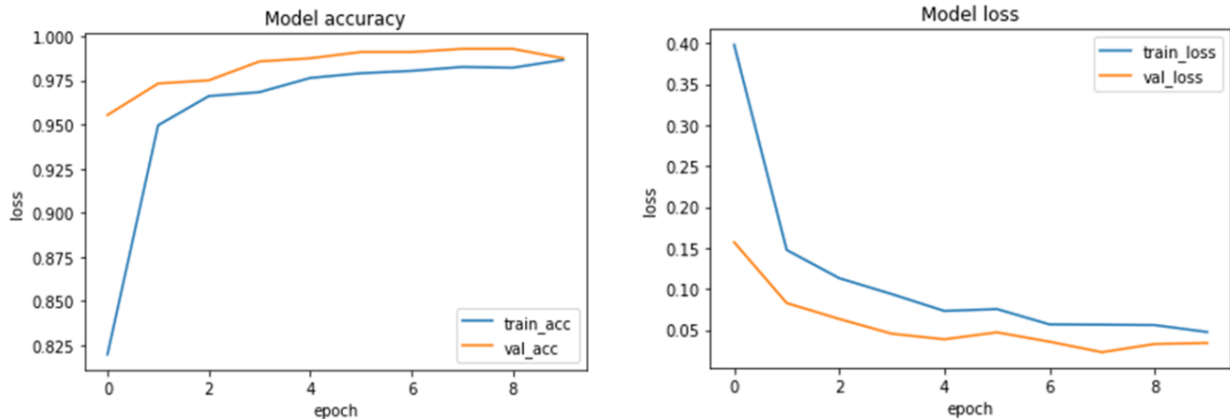
**Table-4.2 The architecture of Auto-encoder**

Layer (type)	Output Shape	Parameter Num.
input_2 (InputLayer)	(None, 80, 80, 3)	0
conv2d (Conv2D)	(None, 80, 80, 32)	896
max_pooling2d (MaxPooling2D)	(None, 40, 40, 32)	0
conv2d_1 (Conv2D)	(None, 40, 40, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 20, 20, 64)	0
conv2d_2 (Conv2D)	(None, 20, 20, 128)	73856
conv2d_3 (Conv2D)	(None, 20, 20, 128)	147584
up_sampling2d (UpSampling2D)	(None, 40, 40, 128)	0
conv2d_4 (Conv2D)	(None, 40, 40, 100)	115300
up_sampling2d_1 (UpSampling2D)	(None, 80, 80, 100)	0
conv2d_5 (Conv2D)	(None, 80, 80, 3)	2703
Total parameters: 358835, Trainable parameters: 358835, Non-trainable parameters: 0		

## 4.4 RESULTS AND DISCUSSIONS

In this chapter two types of model architectures have been considered for classification and segmentation procedures. The architecture models have been executed in a system with 4GB DDR5 Graphics Memory and GPU - NVIDIA 740mx. The implemented 4-layer CNN classifier incorporates the stochastic gradient descent (SGD) optimizer with a learning rate of 0.01 and momentum of 0.9. Cross-entropy specifies the loss function of this typical model. The total number of trainable parameters demonstrated by the model is 532962. The model summary has been reported in Table-4.1. The classification model has been trained and validated against the RGB satellite images. The CNN model has generated very promising results in terms of training and validation accuracy and loss. 99.5% of validation accuracy and 99.2% of training accuracy are achieved with the model for classification task. Apart from training and validation accuracies, other performance parameters have also been studied viz., Precision, Recall, F-score and were proved to be very impressive in terms of their values. The accuracy and loss graphs of the classifier model are shown in Figure 4.7(a) and Figure 4.7(b) respectively and the parametric studies are reported in Figure 4.8. The confusion matrix for the binary classifier model is also shown in Figure 4.8 predicting 590 times true-positive (TP), 2 false-negative (FN), 1 false-positive (FP) and 210 true-negative (TN). From the confusion matrix it is quite

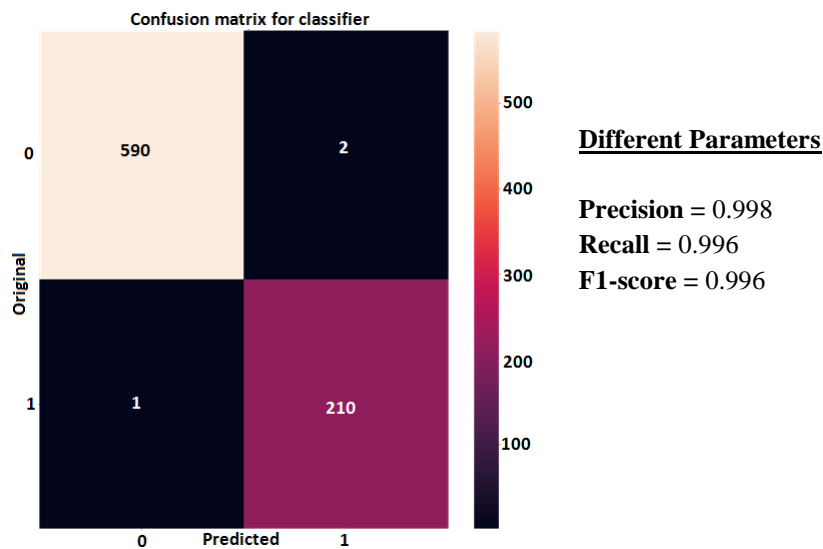
evident that the classification for ship and no-ship has been highly accurate with higher values of TP and TN. The Precision, Recall and F-score have also come up with satisfying values validating the work. The output image with the detected ships and the corresponding original image are shown in Figure 4.9.



(a) Training and Validation Accuracy of CNN classifier

(b) Training and Validation Loss of CNN classifier

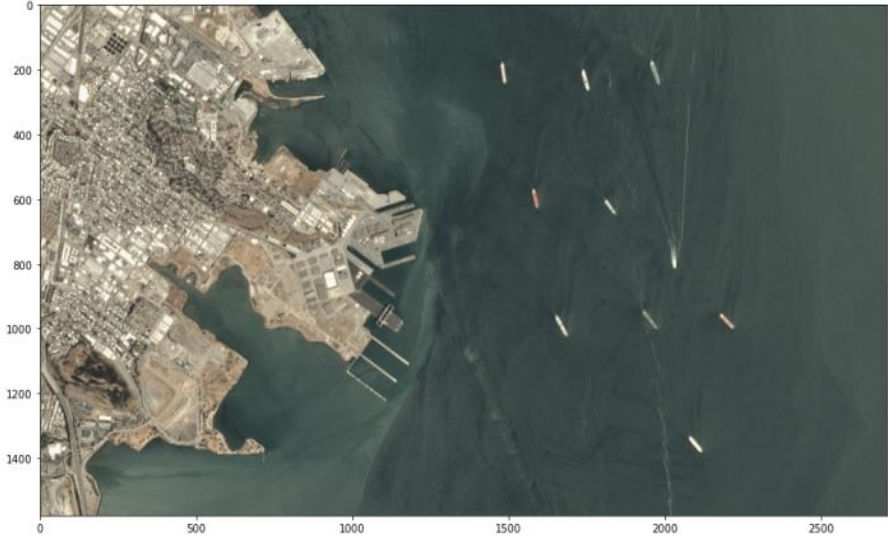
**Figure 4.7 Accuracy and loss plot of CNN classifier model**



**Figure 4.8 Confusion Matrix of CNN classifier model**

In this chapter, it has already been mentioned that the work is accomplished using two approaches, i.e., classification and segmentation. The system is designed with a robust systematic approach to localize as well as segment the anomalies on water bodies from low

resolution satellite or drone images. The auto-encoder's performance in terms of accuracy and loss are shown in Figure 4.10 attaining 84.2% training and 85.1% validation accuracies which are quite remarkable.



(a) A test input image



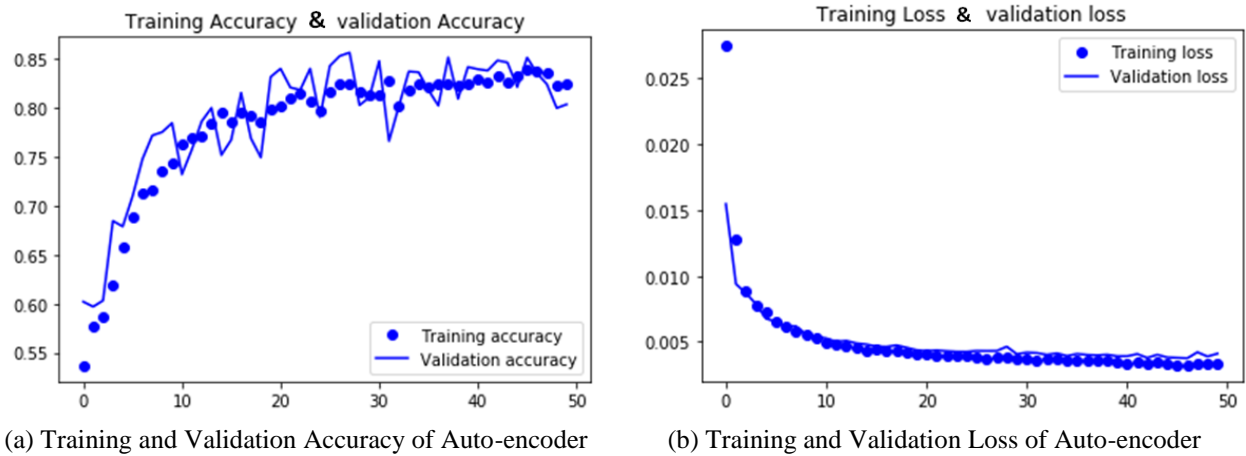
(b) Ships detected in the image

**Figure 4.9** Detection of ships in a test input image

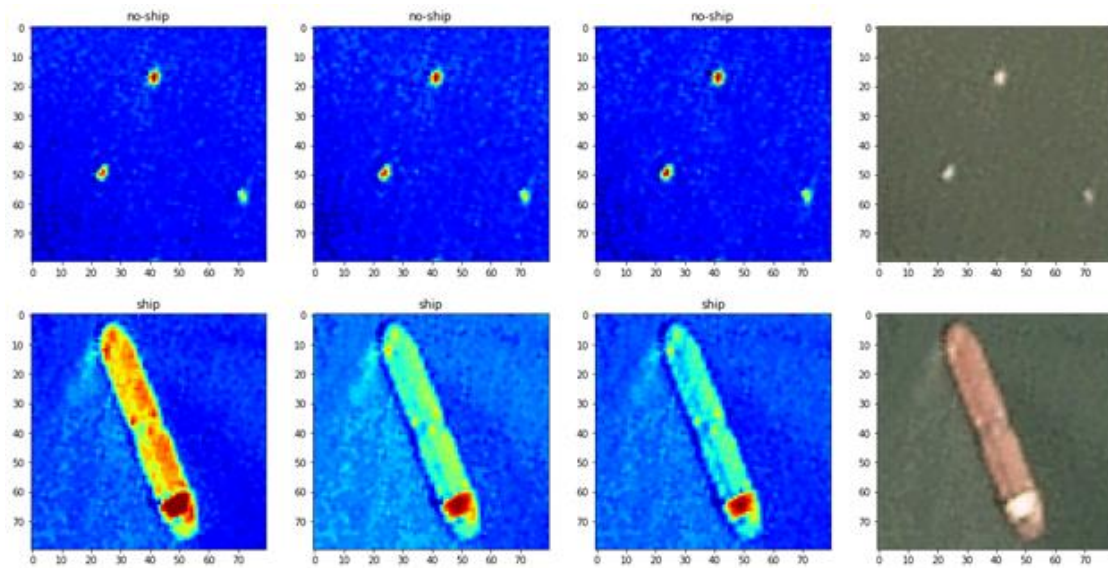
IoU metric, also termed as Jaccard Index, is very effective and commonly used for evaluation of any semantic segmentation model. The segmented image should have a match of the ground truth masks from the satellite input image data. The output obtained after segmentation best

matches with the generated masks depicting notable output for the segmentation of the ship as shown in Table-4.3. The IoU score is calculated for each sample output as witnessed from Table-4.3. The best IoU achieved is 0.77 with this dataset using the auto-encoder model. For the auto-encoder, the total trainable parameters are 358835. Sigmoid activation function is exercised in the model at the last stage. The model has been compiled in 50 epochs with batch size of 128. The time complexity of the model is 2.5 hours approximately for the training with the mentioned specification of the GPU.

Interpretation by the neural network about the decision making regarding classification and segmentation is very important in terms of generated output. Heat maps are the result of the interpretation made by neural network about the accomplishment of the task done by the networks. It is very trivial to know that where the network is looking into or on what features the network is relying for making a decision. Hence in practical real world cases it is absolutely essential to monitor the visualization of the neural network. A sliding window technique is used here and the probability of predictions at each point is calculated. A heat map of the corresponding image is generated using those probability values. The heat maps generated for the sample images are shown in Figure 4.11. A comparison of the proposed model with other models for detection of ships as available in literature is enlisted in Table-4.4.



**Figure 4.10 Accuracy and loss plot of Auto-encoder model**



**Figure 4.11** Typical Heat maps of the sample images



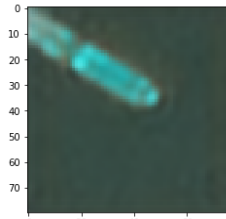


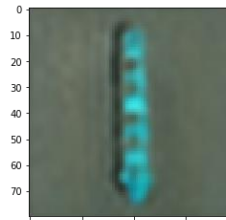


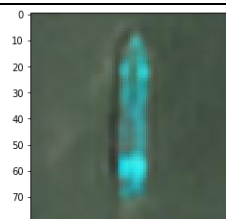


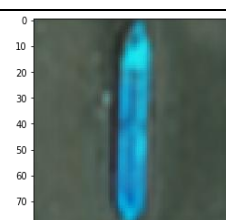
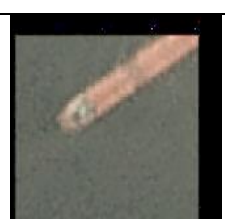

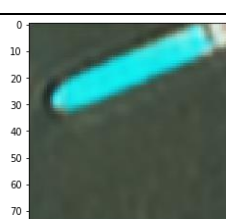
## 4.5 CONCLUSION

Detection and segmentation of ship are carried out in the water body based on satellite images (remote sensing imagery). The satellite RGB images of ships and no-ships are assembled over the San Pedro Bay and San Francisco Bay areas in California. The set of images are fed to the model for classification and thereafter localization and segmentation of the ships in the water body is realized. Results obtained from the CNN classifier model illustrate a validation accuracy of 99.5% and that of the auto-encoder is 85.1%. Other parameters like precision, recall and F1-score have also been studied and extracted in case of the classifier model with very good convincing values. The best IoU score attained by the segmentation model (auto-encoder) is 0.77. The ability of the proposed model can be very much useful in the areas of maritime security management and surveillance, monitoring of traffic, finding and rescuing ships in heavy cyclones, maritime search and rescue operations, navigation management control, etc. The future scope of the present work primarily concerns about the superimposition of the segmented ship images over the original images. Also ship detection can be materialized through remote sensing using high resolution SAR images. But as these images suffer from noise, hence these images can be first de-noised using deep learning approach, often used in dehazing technique, and they can be then further processed to yield the desired result. Hence



this approach can also be considered as a future scope of the present work. The sophistication of the model lies in its novel design, ability to classify, locate and segment the object of interest. Real time implementation of such automated system design will reduce the complexity in terms of time, cost, installation and maintenance.

**Table 4.3: IoU metric of the segmented images**

Original Image	Ground Truth Image	Segmented Output	IoU
			0.4
			0.77
			0.35
			0.39
			0.4



**Table-4.4 Performance analysis of the proposed model with that of other models for ship detection**  
**(Parameters: A – Accuracy, P – Precision, R - Recall)**

Detection of ships	Architecture / Dataset	Parameters
Ship detection based on YOLOv2 [15]	Faster R-CNN / SSDD	A = 70.63% , P = 70.63%
	YOLOv2 / SSDD	A = 90.05% , P = 90.05%
Ship detection in optical remote sensing images [25]	Faster R-CNN / DSSDD	A = 68.43%
	YOLOv2 / DSSDD	A = 89.13%
Ship detection using satellite imagery [42]	DenseNet / Satellite imagery	A = 96.93%
Proposed model	CNN / Satellite imagery	A = 99.5% , P = 99.8%, R = 99.6%

#### 4.6. IMPORTANCE OF AI / ML IN AGRICULTURE

Agriculture is one of the oldest occupations practiced worldwide in the majority of the countries. It forms an important aspect of sustainability and survivability. India being a country primarily having an agriculture-based economy, hence a major part of India’s population is either directly or indirectly linked to agriculture. Moreover, agricultural exports majorly contribute to the country’s GDP. According to the available data, the global population is expected to reach 10 billion by 2050 which in turn would require agricultural productivity to increase by at least 70%. With increase in world population, the demand for agricultural products has increased manifold. As further cultivable land area cannot be increased, hence the only way to increase the amount of agricultural production is to enhance the productivity of the existing lands. Agriculture in many parts of India is still practiced manually in a traditional manner, involving lots of manpower and man-hours. Among several aspects of agriculture, keeping crops disease-free is an important facet. This is done manually and may turn out to be inaccurate owing to some limitations and incorrect judgments of humans which in this case might result in catastrophic consequences; the worst being the whole crop getting ruined. This is a problem that no farmer can afford and hence resort to Artificial Intelligence (AI) based solutions. AI in agriculture serves a better pathway to analyze real-time problems faced by the farmers in day-to-day life. One of the common problems

faced is the invasion of pests, which deteriorates the quality of the crops. The main challenge lies in detecting the diseases caused due to the attack of these pests and for which farmers need innovative technologies to combat such attacks. The joint venture of Computer Vision and Artificial Intelligence makes it a great way to solve such kinds of problems. The most promising factor that rules AI is that it uses real-time data not only to predict the emergence but also the identification of the pest and diseases before it takes a huge shape. Therefore, the main motto of developing such automated systems using AI is to reduce the vulnerability of pest attack and to preserve the quality of production.

Plant disease detection using Deep Learning techniques such as classification and detection has become a crucial aspect in monitoring and analyzing the productivity of each and every specific species of plant. As compared to traditional classification networks, techniques involving deep learning yields better results for real-time identification of plant leaf diseases [132]. These are all headed under the latest improvements in computer vision aided systems to efficiently provide solutions for multiple plant diseases as the existing method for disease detection is through naked eyes which require lots of effort and is time consuming as well [133]. Therefore to reduce this problem Deep Learning has been introduced which involves a robust process with higher accuracy for accurate diagnosis of the respective diseases [134].

In this chapter, the detection of Tomato leaf diseases is accomplished with a Deep Learning approach combined with Machine Learning technique. The overall framework depicts the classification of Tomato leaf images into healthy and diseased ones and then implementation of the images for different categories of disease detection. The entire work has been implemented using Deep Neural Networks, especially using CNN architectures and Principal Component Analysis (PCA) and this new model is named as PCA DeepNet. The PCA work as the primary feature extractor followed by the customized deep neural networks for classification and detection purposes. The convolutional deep learning networks are basically chosen to reduce computational cost and for smooth classification; thereby helping in development of an intelligent systems assisted tomato leaf disease detection. SSD and F-RCNN are used for detection purposes. In F-RCNN, the detection steps are carried out in two steps unlike SSD and hence a more accurate detection is obtained using F-RCNN [135].

The major contributions of the work are listed below which differentiates the presented PCA DeepNet from the existing works available in literature:

- The work is methodized by a hybridized framework consisting of GANs, conventional PCA, customized CNN classifier and detection architecture for respective purposes.
- The system consists of a customized neural network which is basically structured using 10 convolutional neural layers. This is formulated utilizing a feed forward network where the method of down-sampling and up-sampling is predominant and the work is tuned using several hyper-parameters like Learning rate, Optimizers, Activation Functions to enhance the performance of the model.
- The changes made in the dropout values, pooling layers and max pooling layers gives the proposed work a better formulation and thus reduces time complexity of the entire system. The lesser time consumption is compared with respect to the hardware and software specification mentioned later in the results and discussion section.
- The newer system which has not been implemented earlier generates better accuracy and other performance metrics like precision, recall, F1-measure scores.
- The novelty of the overall framework is materialized through a customized hybridized model involving classical machine learning model along with deep neural networks which are structured in such a way so as to outperform any of the existing works.

## **4.7. RELATED WORKS**

Detection and classification methods for different diseases in leaves, plants and crops have been exhaustively studied in the recent past. Even studies involving the causes of diseases in crops through attack by different pests have also been investigated to reduce the crop yield losses [136]. The classification of leaf diseases using deep learning is mainly done with the help of Transfer learning techniques. In [137] a light weight transfer learning based approach is adopted for efficient detection of tomato leaf diseases. Herein a pretrained MobileNetV2 architecture along with a classifier network is used for prediction yielding 99.3% accuracy. Also the MobileNetV2 architecture has been used in [138] for Bean leaf diseases classification yielding more than 97% and 92% accuracy results on training and testing dataset respectively. In [139] a brief analysis shows that the VGG family provides better accuracy while validating the dataset

for classification as well as detection and VGG 16 gave an accuracy of 99.25%. The state-of-the-art results was also generated when several other CNN architectures like Resnet-50, Xception, Mobilenet, ShuffleNet, Densenet121\_Xception were used for feature extraction and studies on the comparative detailing of these architectures [140] revealed that Densenet121\_Xception gave the best training accuracy of 97% followed by ShuffleNet which could only give 83.68. Many more CNN classification architectures like AlexNet and SqueezeNet highlighted some of the good parameters for a detailed study. In [141] a compact CNN is proposed by the authors for tomato leaf disease identification involving six layers network. Another new model based on CNN Architecture was developed with Adam optimizer and with the help of Image Augmentation an accuracy of 96.55% was achieved. Also by using CaffeNet Architecture and fine-tuning the data a similar accuracy of 96.3% was obtained by authors in [142]. Leaf disease detection has been subjected to several other comparative studies using Alexnet and ResNet by [143] where AlexNet fetched a better accuracy of 97%.

Authors in [144] have developed their own feature extraction technique using Gray-Level Co-occurrence Matrix (GLCM), Complex Gabor Filter, Curvelet and Image moments; further they have trained their Neuro-Fuzzy logic classifier with feature extractor using MATLAB simulation tool. Another approach to implementing MATLAB is used in [145], where Image-Segmentation and feature extraction using Color co-occurrence method is proposed and finally classification is done using Back Propagation Neural Network (BPNN). In recent years many new upcoming techniques have been formulated using deep learning, some of which are Precision farming to enhance production and Soft Computing Technologies involving Segmentation processes.

In recent years, machine learning has enabled creation of newer ways for efficient disease detection. In [146] a survey on different machine learning classifiers like SVM, k-nearest neighbor and fuzzy logic has been carried out to get an overview of these algorithms. Many changes in algorithms and techniques were done later on to create better output using clustering and classification techniques. Algorithms such as SVM and K-mean Segmentation also generated better accuracy of 90%. A new method presented in [147] used low-level features of Luminance and colour along with multi-scale analysis for determining saliency maps and then using k-means algorithm for soybean leaf disease detection. Another approach [148] is adopted using SIFT for extraction of features and analyzing the results using an SVM classifier. Some

researchers use different architectures and algorithms to acquire the desired results. In [149] the images were segmented using K-Means Clustering, extracted the features with GLCM and LBP and finally classified them using SVM. Regression techniques were used by researchers as reported in [150] where SVR (Support Vector Regression) and GPR (Gaussian Process Regression) have been used. SVR was used in the estimation of biochemical and biophysical parameters of the plant while GPR, a kernel-based machine learning method, was used for nonlinear regression problems.

GANs (Generative Adversarial Networks) are widely used nowadays by researchers as tools for image augmentation. GANs increase efficiency of classification models to a greater margin. Researchers in [151] proposed the first work using GANs for synthetic augmentation of the dataset in improving performance of plant disease recognition. Authors optimized the activation reconstruction loss (ARL) function that put forward an enhanced AR-GAN, comparing it with prominent existing models.

The proposed model provided a significant increase of about 5.2% in the classification accuracy as compared to the classical ones. As reported in [152], using DCGAN to augment the dataset, researchers were able to achieve 20% higher accuracy than those using conventional tools for augmentation. GANs were also able to solve another major problem of data imbalance or class imbalance; also providing endless high-quality data. In [153], Double GAN approach is adopted wherein two GANs are used to obtain a pre-trained model and SRGAN is used to increase the residual network to prevent overfitting. Researchers in [154] have generated augmented images using C-DCGAN (Conditional Deep Convolutional GAN) as the input to VGG16 and found average accuracy to be around 28% higher than conventional methods like rotation and translation. Authors in [155] augmented four types of grape leaf disease with a novel Leaf GAN model. The experimental results revealed that the Leaf GAN model could make the images highlight the disease and could also generate enough synthetic grape leaf disease images, proving that the method is superior in comparison to DCGAN and WGAN.

## 4.8. METHODOLOGY

The proposed PCA DeepNet is meticulously designed for performance where each and every step of data preparation and analysis has been optimized for best results. The pre-processed image data is first made to go through an augmentation process using GANs, where the data is refined and made more trainable for further processes. Then the data is processed with a feature extraction technique performed using conventional PCA. Thereafter, the classification of the data, which is the highlight of PCA DeepNet, is done using a customized CNN Classifier specifically designed to process the data. At last, the classified outputs are detected using a faster region-based CNN. The overall system workflow is shown in Figure 4.12.

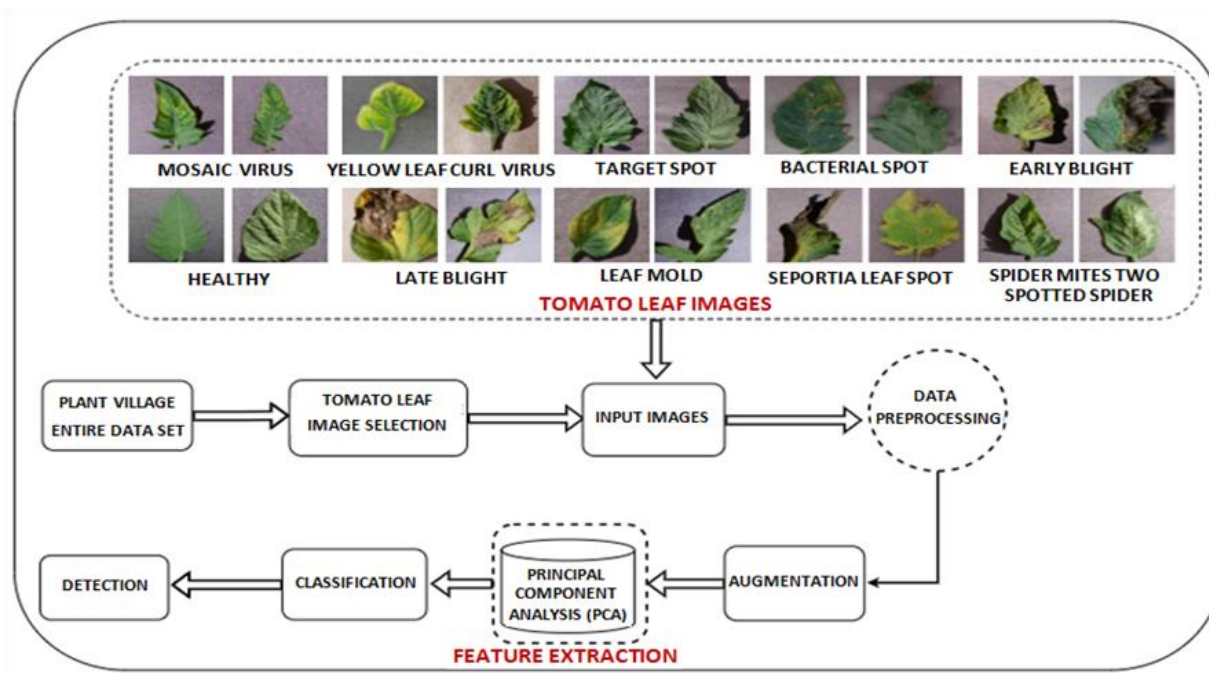
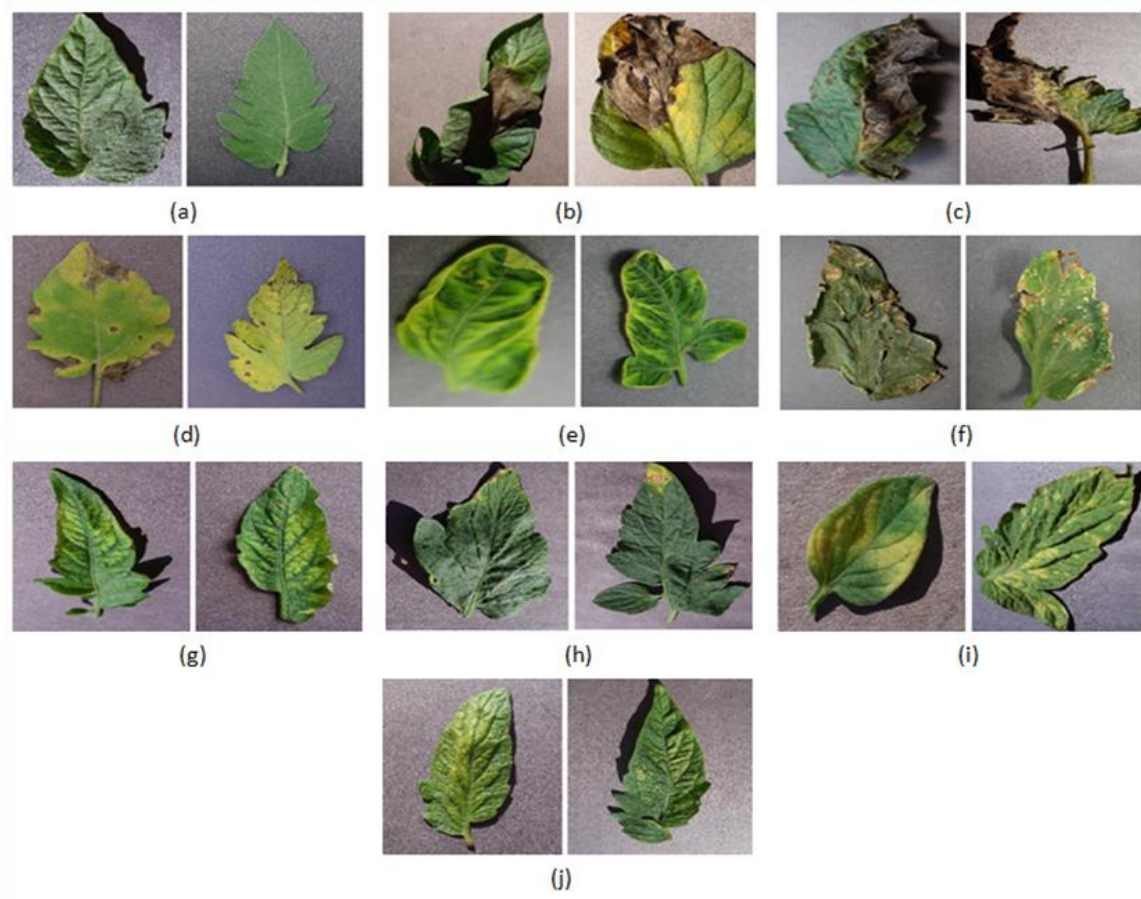


Figure 4.12 Block diagram of overall System

### 4.8.1 Dataset

For training a Deep Learning model to be able to classify precisely and with high accuracy, an image dataset with proper and balanced image samples is very essential [156]. The larger the dataset size, the more accurate the deep learning model can be obtained. Here, the Plant Village dataset [157] is selected which is an open-source agricultural disease dataset. It is a collection of

more than 56000 images divided into 38 classes consisting of 19 crops (apple, grapes, tomato, potato etc.). The dataset consists of high-quality images of leaves in .jpeg format with a width of 5472 pixels and a height of 3648 pixels. Among the 19 crops, only tomatoes are taken into consideration and some of them are shown in Figure 4.13. The tomato data is distributed into 10 different classes namely Late\_blight, Healthy, Early\_blight, Seportia\_leaf\_spot, Yellow\_leaf\_curl\_virus, Bacteria\_spot, Target\_spot, Mosaic\_virus, Leaf\_mold and Spider\_Mites\_two\_spotted\_sider which consists of a total of 18,128 images of tomato leaves. Table 4.5 highlights the overall data employed in this work. The enlisted table also throws light on the entire images utilized to carry forward the current framework. Thus a multi-class classification of the dataset is performed in the presented work.



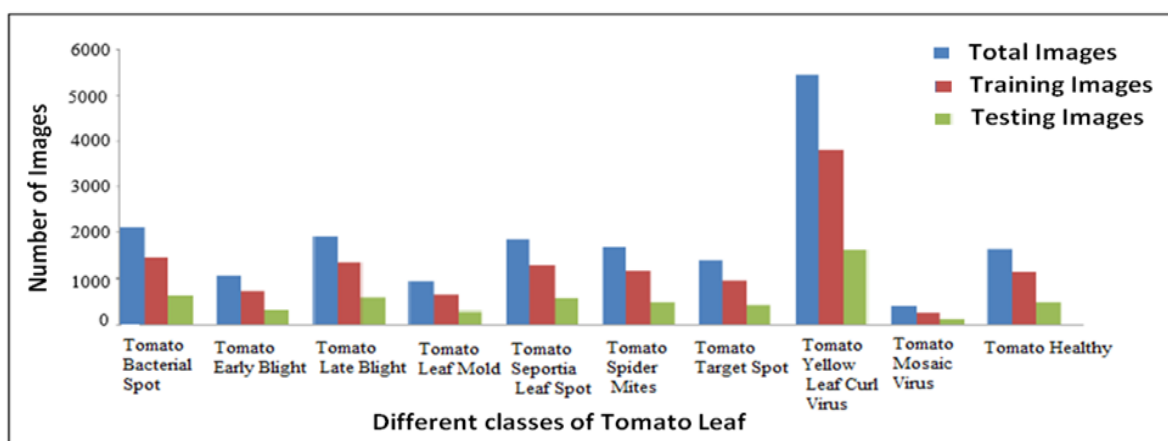
**Figure 4.13 Dataset images – (a) Healthy (b) Late\_blight (c) Early\_blight (d) Seportia\_leaf\_spot (e) Yellow\_leaf\_curl\_virus (f) Bacteria\_spot (g) Target\_spot (h) Mosaic\_virus (i) Leaf\_mold (j) Spider\_Mites\_two\_spotted\_spider**

**Table 4.5 Detail Information of Dataset employed**

Tomato leaf Diseases Classes	Total images Per classes	Training images	Testing images
Tomato Bacterial Spot	2126	1488	637
Tomato Early Blight	1057	740	317
Tomato Late Blight	1938	1356	582
Tomato Leaf Mold	952	667	285
Tomato Septoria Leaf Spot	1860	1302	558
Tomato Spider Mites	1700	1190	510
Tomato Target Spot	1386	971	415
Tomato Yellow Leaf Curl Virus	5456	3819	1637
Tomato Mosaic Virus	400	280	120
Tomato Healthy	1662	1163	499

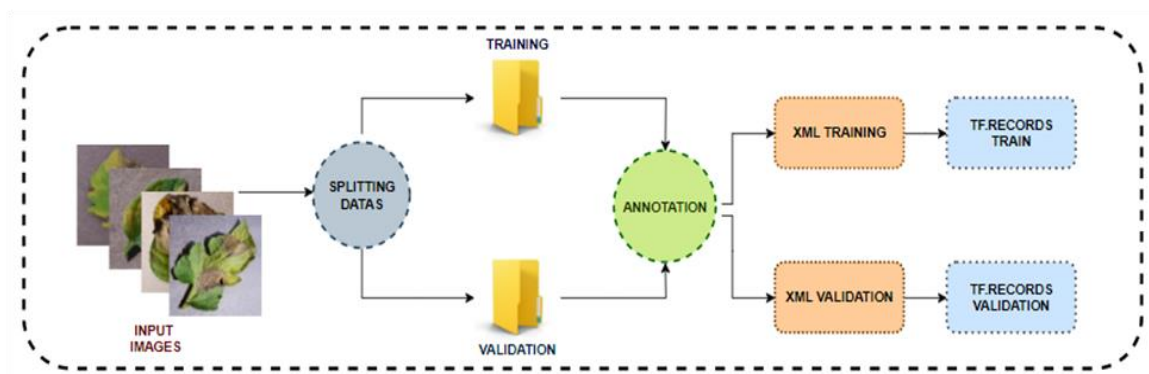
### 4.8.2 Data Pre-processing

The collected data which is used as input images are pre-processed to make it trainable with the proposed model. The whole dataset consisting of 18,128 images is split into training and validation sets in ratio of 7:3, where each set consists of all the 10 classes. After the segregation of the dataset, each and every image is combined separately according to the diseases. The graphical representation of the overall data pre-processing technique is exhibited in Figure 4.14 while its schematic representation is shown in Figure 4.15. In the proposed work a classical method of splitting the data into train and test is carried out to grasp the convenience of easy compatibility with the current dataset.



**Figure 4.14 Graphical representation of data pre-processing**





**Figure 4.15 Schematic diagram of Data Pre-processing**

### 4.8.3 Annotation

Annotation of the images plays a major role in detection of different diseases. The annotation of the images in this work is mainly done using labeling software, an open source graphics annotation tool. The images are labeled using bounding boxes; the most commonly used type of annotation in object detection and localization tasks. Figure 4.16(a) and Figure 4.16(b) depict the picture and the XML document associated to label these images using python. The different kinds of diseased images have been identified by professional experts in the agriculture domain. Detailed information about the annotation of each class is given in Table 4.6. The entire table contains all the detail information regarding all the images which are annotated for computing the detection process. The annotation is done in PASCAL format since the detection architecture used in the presented work is Faster Region Based Convolutional Neural Network.

**Table 4.6 Detail information of Image annotation**

Tomato leaf Classes from Plant Village Dataset	Annotation Labeling	Total images Per classes	Training images	Testing images
Tomato Bacterial Spot	Tomato_BS	2126	1488	637
Tomato Early Blight	Tomato_ER	1057	740	317
Tomato Late Blight	Tomato_LR	1938	1356	582
Tomato Leaf Mold	Tomato_LM	952	667	285
Tomato Septoria Leaf Spot	Tomato_SLS	1860	1302	558
Tomato Spider Mites	Tomato_SMP	1700	1190	510
Tomato Target Spot	Tomato_TS	1386	971	415
Tomato Yellow Leaf Curl Virus	Tomato_YLCV	5456	3819	1637
Tomato Mosaic Virus	Tomato_MV	400	280	120
Tomato Healthy	Tomato_Healthy	1662	1163	499



```

<annotation>
  <folder>Tomato___Bacterial_spot</folder>
  <filename>0d56df83-84fb-4189-a5d2-3a6da18a224d___UF.GRC_BS_Lab...</filename>
  <path>E:\desktop\leaf_data\Tomato___Bacterial_spot\0d56df83-84fb-4189-a5d2-3a6da18a224d___UF.GRC_BS_Lab...</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>256</width>
    <height>256</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>Tomato_BS</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>34</xmin>
      <ymin>124</ymin>
      <xmax>86</xmax>
      <ymax>208</ymax>
    </bndbox>
  </object>
  <object>
    <name>Tomato_BS</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>121</xmin>
      <ymin>64</ymin>
      <xmax>195</xmax>
      <ymax>175</ymax>
    </bndbox>
  </object>
</annotation>

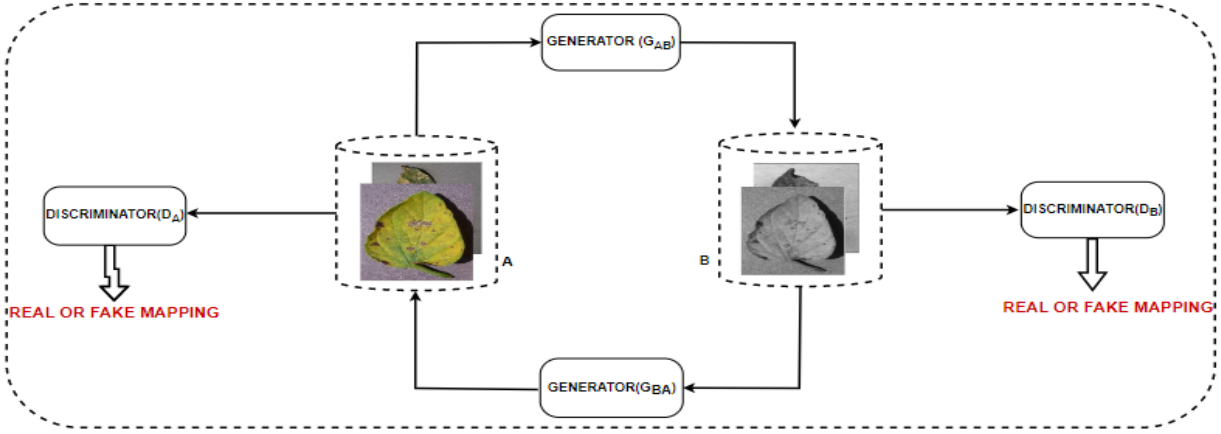
```

**Figure 4.16 Annotation of Tomato Leaf Images (a) Annotated image (b) XML document**

#### 4.8.4 Data Augmentation

Data augmentation is a technique of refining the dataset which facilitates the training of classification models. In agricultural disease datasets like the Plant Village dataset, the onset period of certain diseases is shorter, which makes it difficult to collect enough samples of them. In the field of deep learning, small sample size and data imbalance are major factors leading to poor recognition and classification. As a mitigation technique, data augmentation is applied to artificially increase the amount of data by generating new data points from existing data. Here GANs have been used as a modern approach to data augmentation. Unlike any other conventional augmentation models, GAN aims at learning the distribution of a training dataset to generate new (synthetic) data instances. The GAN model comprises two sub-models: generator and discriminator, which work against each other. The discriminator is trained on both real and fake data. It learns to get better at distinguishing the generated fake data and real data and the

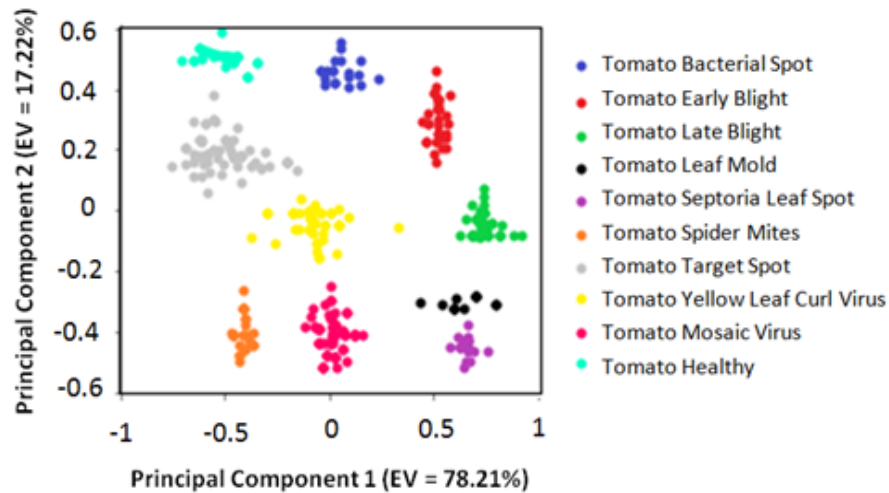
generator learns to generate more realistic new data points from random inputs. The process continues until the generator can create data instances that the discriminator cannot distinguish from real data. Out of the various types of GANs suitable for different purposes, here CycleGAN is used owing to its suitability for image augmentation. The most important feature of CycleGAN is that it can perform image translation on an unpaired image where there is no relation between input and output images. A diagrammatic representation of the above process is shown in Figure 4.17.



**Figure 4.17 Image Augmentation using CycleGAN**

**4.8.5 Feature Extraction**

The datasets commonly available these days have hundreds of features. If the number of features becomes similar to the number of observations stored in a dataset, then this can certainly lead to over fitting of the model. Amongst the different available Feature Extraction techniques, herein the presented work is accomplished using a classical machine learning feature extraction method known as PCA which is an unsupervised linear dimensionality reduction technique mainly used for feature extraction and reduction of image dimensions. PCA is primarily chosen as it aims toward finding the direction of maximum variance in high dimensional data; thereby helping in easy identification of the object. PCA helps to project the original data into a set of orthogonal axes and each of the axes gets ranked in the order of importance [158]. Figure 4.18 shows the projection of different classes of tomato leaf after evaluating the eigen vectors using PCA.



**Figure 4.18** Projection of different classes of leaf after implementing PCA

#### 4.8.6 Image classification

Classification of images has been done using many hybrid models which includes a mixture of machine learning and deep learning models namely viz. AlexNet-SVM [159]. Classification is a major technique for classifying the diseased one amongst the healthy leaves. This step includes the usage of several machine learning models like SVM and KNN [160]. Classification of images can also be performed using pre-trained CNN models like Resnet-50, Xception, Mobilenet, ShuffleNet, Densnet121\_Xception, AlexNet, GoogleNet, VGGNet [161] etc. The pre-trained models are saved networks that had been previously trained on a large scale dataset. It includes all steps like data augmentation, feature extraction, image classification etc. unlike the proposed PCADeepNet where each step is customized according to the dataset for providing best results. For classifying the images, CNN is customized in such a way so that it contains a stack of 10 Conv2D layers where each convolutional layer consists of a pooling layer, dropout, a max-pooling layer and the activation function ReLU.

ReLU is best suited for multi-class classification and does not saturate for the positive value of the weighted sum of inputs. An input image is fed to the classifier after extracting the features using PCA. A series of 10 layers is made which consists of kernel size, stride and padding layer

of different values as shown in Figure 4.19. The entire classifier is structured in such a way that the first five convolutional layers consist of a feed forward neural network that increases in size from 32 to 512. The next set of convolutional layers presenting down-sampling are concatenated with the previous layers of up-sampling. In the following model, the up-sampling method is utilized and conv6, conv7, conv8 and conv9 are designed by concatenating layers (conv5,conv4), (conv6,conv3), (conv7,conv2), (conv8,conv1) respectively. In the present work, Categorical Cross-Entropy is used as the loss function; the formula for the same is given in equations 4.1 and 4.2 respectively. Equation 4.1 is for the Softmax activation function and equation 4.2 stands for the Cross Entropy.

$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \dots\dots\dots (4.1)$$

$$CE = - \sum_i^C t_i \log(f(s)_i) \dots\dots\dots (4.2)$$

Here f(s) is the function, C is the class for which the probability needs to be calculated. The letter t stands for the target vector. The classifier also uses Adam as an optimizer with a learning rate of 0.01 and a momentum of 0.9 respectively. The mathematical formula associated with it is given in equation 4.3.

$$m_n = E [X^n] \dots\dots\dots(4.3)$$

where m is the moment and X is the random variable and n is the expected value of the moment. The outer layer is the fully connected layer which comprises the input from the convolutional layer 9 with a sigmoid activation function. The computation of the classifier using 35 epochs is performed and overall training time reaches to 25 min. However an early stopping [162] function callback is used for resolving the problem of over fitting. The hybridized framework is trained using a huge dataset and the limited GPU helps in consumption of less time; thereby boosting the proposed work with utilization of less computational resources. The inference time calculated for the present classifier is less than any other models used. Thus the work highlights the efficacy of the presented work effectively.

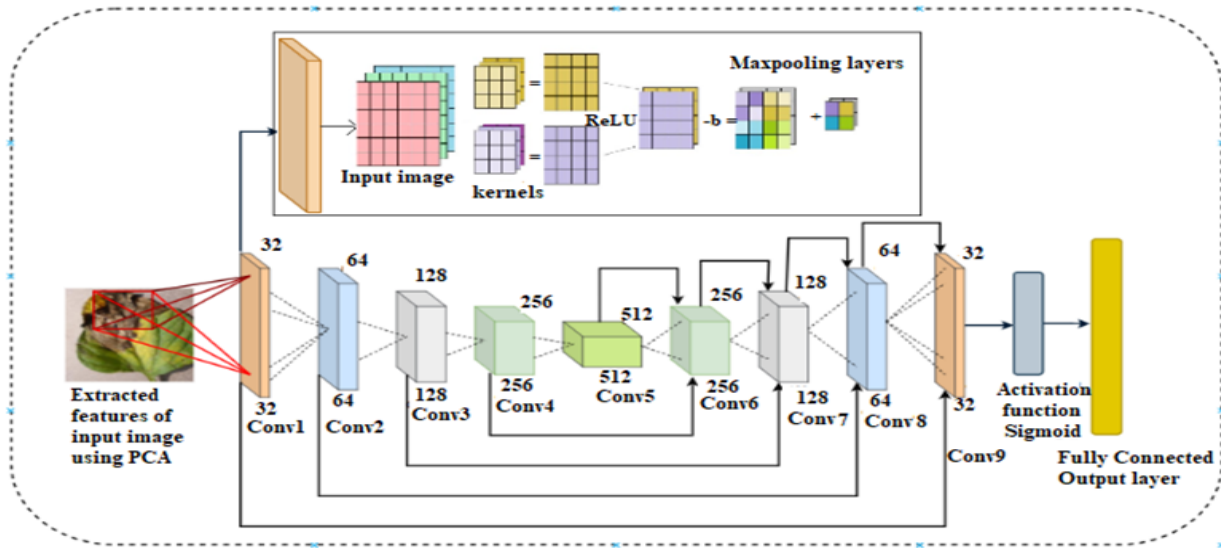


Figure 4.19 Schematic representation of the classifier model

#### 4.8.7 Performance Parameters

The performance of the PCA DeepNet classifier model is evaluated based on several metrics which are calculated using True positive and True negative (Tp & Tn) and False positive and False negative (Fp & Fn) values obtained from the confusion matrix while training the models [163].

1. Accuracy – It is the most crucial and intuitive performance measure which ensures the percentage of correct prediction based on the total number of observations present.

$$Accuracy (ACC) = \frac{Tp+Tn}{Tp+Tn+Fp+Fn} \dots\dots\dots (4.4)$$

2. Precision – Also called Positive Predictive Value, it represents the number of samples actually and predicted as positive from the total number of samples predicted as positive. The precision measures the model’s accuracy in classifying a sample as positive. In some cases, precision is preferred over recall because it does not depend on the false negative values which rules out the problems arising due to class imbalance.

$$Precision (PPV) = \frac{Tp}{Tp+Fp} \dots\dots\dots (4.5)$$

3. Recall – It is the number of samples actually and predicted as positive from the total number of samples actually positive. It gives a measure of how accurately the model is able to identify the data specifically the true positives.

$$Recall (Sensitivity) = \frac{Tp}{Tp+Fn} \dots\dots\dots (4.6)$$

4. F1-Measure – The F1-Score combines the precision and recall of a classifier into a single matrix by taking their harmonic mean. It measures a model’s accuracy for a dataset. It is used to compare the performance of two classifier models.

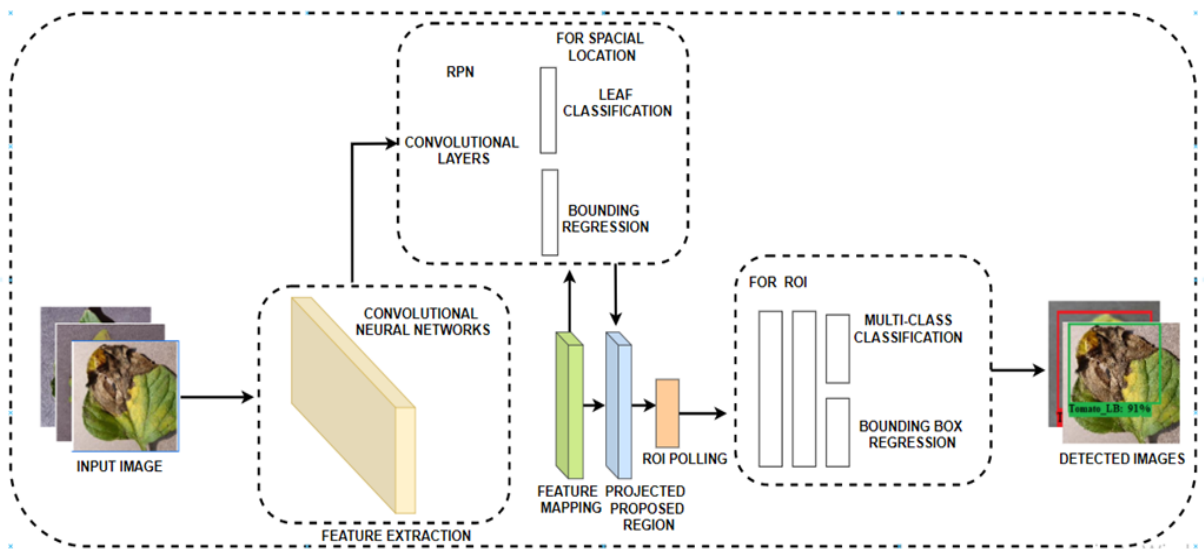
$$F1 - Measure = 2 * \frac{Tp}{2*Tp+Fp+Fn} \dots\dots\dots (4.7)$$

#### 4.8.8 Detection

The proposed approach of the work is to train the data using customized PCA DeepNet model. In case of detection, the authors have used the model F-RCNN. In Faster RCNN, PCA DeepNet is used as the backbone architecture for training. An input image is used to train the model. This detected as well as perfectly localized the images due to its improved architecture. It uses two networks – one for region proposal and another for object detection. It consists of 9 anchors for creating bounding boxes of specific size. The output obtained in this architecture is due to the RPN which easily classifies the diseased and the healthy Tomato Leaf in the form of rectangular bounding boxes by reframing the anchor. The training accuracy of this model is higher. The architecture used produces an IoU score of 0.95 with a threshold score of 0.8. The entire process is diagrammatically represented in Figure 4.20.

### 4.9. RESULTS AND DISCUSSION

A detailed and exhaustive study has been carried out while validating the PCA DeepNet classifier model. The overall training has been processed using Google Colab with GPU specification of Tesla K80 (2496 CUDA cores). The pre-processed augmented data of 10 classes are taken as input and different classification metrics were generated while training it. The software and hardware specifications used in the work are enlisted in Table 4.7.



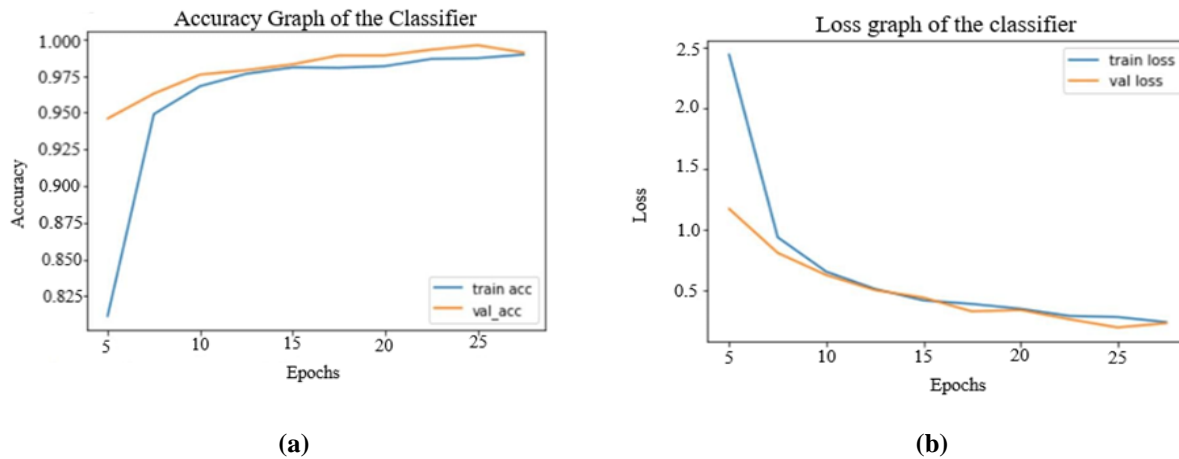
**Figure 4.20** Block diagram of Faster Region-Based Convolutional Neural Networks

**Table 4.7** Software and Hardware specifications

Configuration	Value
CPU	Intel core i5 8th Generation
GPU	Tesla K80 (2496 CUDA Cores)
Hard Disk	1TB
Operating System	Windows 10

The Accuracy and loss graph of PCA DeepNet classifier is highlighted in Figure 4.21(a) and Figure 4.21(b). The graphs are validated against 50 epochs although for optimized fitting of the model for accuracy and loss an early stopping function named callback of 30 epochs was implemented. The resulting graph generated into a well fitted optimized curve. In [164], the authors trained the apple leaves disease data using DenseNet and EfficientNet which resulted in inconsistent accuracy and loss graphs. The overall time taken to compile the process is 25 minutes.

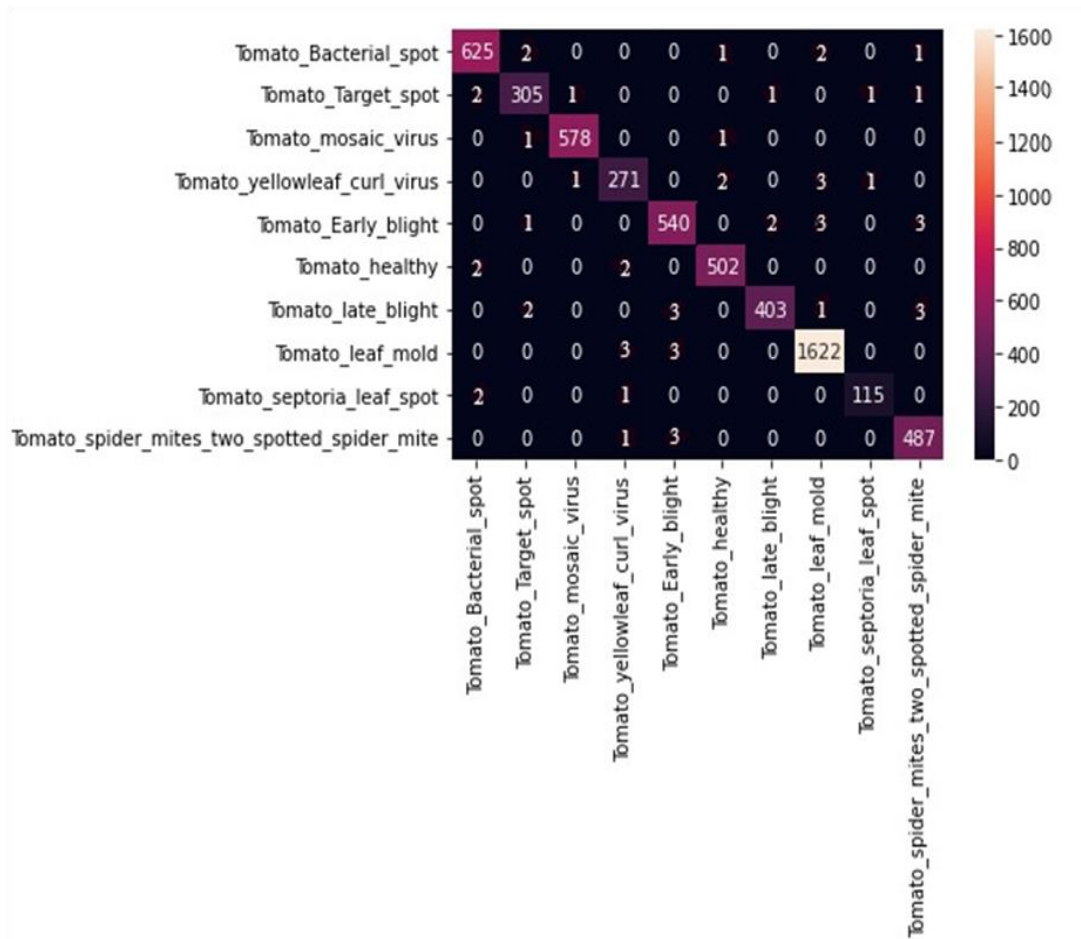




**Figure 4.21 (a) Accuracy graph of the classifier (b) Loss graph of the classifier**

The custom PCA DeepNet generates the confusion matrix, which qualifies the classifier's performance and provides necessary data for calculating the performance parameters like precision, recall, F1-Score etc. The diagonal elements of the confusion matrix indicate the number of points for which the predicted label is equal to the true label. Figure 4.22 represents the confusion matrix generated by the classifier which has high diagonal element values indicating a large number of correct predictions. The overall summary of the result gives a concrete inference that the classifier generated the best values for each class.

The other parameters like precision, recall and F1-measure are also generated and give a promising value indicating the novelty of the classifier in specifically classifying each and every class without getting confused with others. In [165] where the authors trained their data with ResNet34, the results generated in terms of accuracy, F1-Score etc. were less as compared to the proposed AIS. The entire structure of the presented PCA DeepNet is enlisted in Table 4.8 for smooth understanding of the overall work. Brief analyses of the overall training and its resulting accuracies and performance metrics have been entitled in Table 4.9. The main objective is to get the result which includes Accuracy, Precision, Recall and F1-measure scores obtained while training the novel PCA DeepNet model. It can be seen that the proposed architecture detected each and every class of tomato leaf disease with higher accuracy. The study is accomplished using Adam optimizer at a learning rate of 0.01; however, the application of different optimizers on the classifier is also performed in the experiment.



**Figure 4.22** Confusion matrix generated by PCA DeepNet

Table 4.10 highlights the results which are generated when PCA DeepNet is compiled using different optimizers respectively. The best results are listed in the table to set a comparison between different optimizers using different learning rates and predicting the outcomes efficiently. The work proves that mostly all the optimizers could perform well using the proposed PCA DeepNet classifier. The performance parameters include Accuracy, Precision, Recall and F1-Score. Thus it can be said that in terms of Optimization, the model is fully optimized to perform the training process.

**Table 4.8 Detail structure of PCA DeepNet**

Name	Kernel	Pooling Size	Number of Filters	Stride	Padding	Dropout	Activation
Input Image	-	-	-	-	-	-	-
Conv_1	5		32	3	Same	0.2	ReLU
Pool_1		2					
Conv_2	3		64	3	Same	0.25	ReLU
Pool_2		2					
Conv_3	3		128	3	Same	0.3	ReLU
Pool_3		2					
Conv_4	3		256	3	Same	0.3	ReLU
Pool_4		2					
Conv_5	3		512	3	Same	0.3	ReLU
Pool_5		2					
Up_6	2		256	2			
Conv_6	1		256	3	Same	0.3	ReLU
Up_7	2		128	2			
Conv_7	1		128	3	Same	0.3	ReLU
Up_8	2		64	2			
Conv_8	1		64	3	Same	0.3	ReLU
Up_9	2		32	2			
Conv_9	1		32	3	Same	0.3	ReLU
Conv_10		1		1			Sigmoid
Fully Connected layer							
Total Training Parameters	7,759,521						
Trainable Parameters	7,759,521						
Non-Trainable Parameter	0						

In the presented work, further experiment is carried out using different pre-trained Deep Learning models. The transfer learning process is applied for each and every DL classifier using the Plant Village dataset. The various models are utilized based on several hyper-parameters to compare the efficiency of the presented customized classifier PCA DeepNet. Table 4.11 enlists the various ranges of the hyper-parameters values used in different architectural models while compiling the same using Tomato leaf diseases dataset. Several parameters like the trainable, non-trainable layers, optimizers, learning rate, batch size, activation function, epochs and dropout values are altered. In the VGG16 model, first the base model is freezed and after training

the same it is again unfreezed in order to reduce the computational time of the model. The entire model takes 40 min to execute the results. The InceptionV3 model is trained altering several hyperparameters like hidden trainable layers, optimizers, learning rate, activation function, batch size, epochs and dropout values respectively. However in this model freezing of any layers is not done and the compilation is carried out using transfer learning technique. The training time consumption of the model is 35 min. After successful completion of the process the corresponding results for the same are calculated.

**Table 4.9 Performance parameters obtained from the Confusion Matrix**

Leaf Classes	Precision (%)	Recall (%)	F1-measure (%)	Accuracy (%)
Tomato Bacterial Spot	99.04	99.04	99.04	99.20
Tomato Early Blight	98.07	98.07	98.07	98.07
Tomato Late Blight	99.48	99.82	99.65	99.65
Tomato Leaf Mold	97.48	97.48	97.48	97.48
Tomato Septoria Leaf Spot	98.36	98.36	98.36	99.36
Tomato Spider Mites	99.01	99.01	99.01	99.01
Tomato Target Spot	98.05	97.81	97.93	97.90
Tomato Yellow Leaf Curl Virus	99.51	99.51	99.51	99.51
Tomato Mosaic Virus	98.29	97.45	97.87	97.92
Tomato Healthy	98.78	98.78	98.78	98.78

**Table 4.10 Performance parameters obtained using different Optimizer on PCA DeepNet model**

Optimizers	Learning Rate	Accuracy	Precision	Recall	F1-Score
SGD	0.01	96.90	94.32	94.30	94.31
RMSProp	0.001	94.40	93.32	93.20	93.26
Adadelta	0.001	92.31	92.30	92.30	92.30
Adagrad	0.0001	95.32	92.32	92.30	92.31
Adamax	0.001	97.32	96.90	96.92	96.91
Adam	0.01	99.60	98.55	98.49	98.52

**Table 4.11 Details of Hyper-Parameters used in different architectural models**

Hyperparameters	Experimental Value Range			
	VGG 16	Inception V3	Inception Resnet V2	Resnet 152V2
No. of Hidden trainable layers	1-3	1-3	0-3	0-3
Nodes per trained layer	64 - 2048	128 - 2048	64 - 2048	64 - 2048
Learning rate	0.01 - 0.00001	0.001 - 0.000001	0.001 - 0.000001	0.001 - 0.000001
Learning Rate Decay	Yes	-	-	-
Drop out	0.0 - 0.5	0.0 - 0.75	0.0 - 0.7	0.0 - 0.7
Batch size	64	64	64	64
Optimizer	Adam	Adam	Adam	Adam
Activation Function	Softmax	Softmax	Softmax	Softmax
Epochs	30-50	30-50	30-50	30-50
Base model	Freeze, Unfreeze	Unfreeze	-	-
Momentum	-	-	0.9	0.9

The presented work also comprises of an exhaustive study of the different hyperparameters incorporated while training the dataset using InceptionResnetV2 architecture as shown in Table 4.11. Different parameters like Hidden trainable layers, nodes per trained layer, learning rate, momentum, optimizer, activation function, batch size, epochs and dropout values are changed. The top layer of this model was freezed so as to generate good results while compiling it. This helps in smooth training of the entire model. The time taken to complete the process is noted to be 32 min. Experimentation is also done using Resnet152V2 and the overall hyper-parameters used in training the model are listed in Table 4.11. The variation in the parameters like hidden trainable layers, optimizers, learning rate, momentum, activation function, batch size, epochs and dropout values resulted in valuable results for the same. The total training time for the DL architecture is 30 min and none of the layers are freezed in the architecture.

The entire approach highlights a detail analysis on how different deep learning models work on the present Tomato leaf diseases dataset. The different models used in the present experiment sets a good comparative report on the different parameters and how much time each and every model takes to generate the results. It is hereby observed that Resnet152V2 takes lesser time for

computation as compared to other Deep Learning models used in the current work. Hence it can be concluded that different DL models perform differently when several parameters are altered, which is primarily due to the different model complexities that are mostly generated owing to the number of layers in it.

A detailed study has also been conducted where the Plant village dataset is utilized to validate other pre-trained Deep learning architectures. The resulting performances as well as the inference time are enumerated in Table 4.12 which indicates that the proposed customized classifier is better than any other state-of-the-art described and thus establishes the proof of its excellence. The paper also proposes a good utilization of the Plant Village dataset and the same is further used to train some existing Machine Learning Algorithms. Table 4.13 summarized the different values of hyper-parameters used in compiling the ML algorithms on Plant village dataset. The performance results in terms of Accuracy, Precision, Recall and F1-Score are mentioned in Table 4.14 while the graphical representation of the same is shown in Figure 4.23. The present work is computed on Machine Learning algorithms as well as Deep Learning algorithms including the proposed PCA DeepNet architecture. From Table 4.12 and Table 4.15 it is clearly revealed that deep learning architecture generates promising results. The present scores give an insight that the proposed PCA DeepNet classifier is best among all the Machine Learning as well as Deep Learning classifiers. This clearly states that the proposed PCA DeepNet generates a sharp rise in the performance parameters which is more than 15% compared to Machine Learning algorithms and more than 5% compared to Deep Learning classifiers respectively. Table 4.15 represents a comparative analysis of different works based on the Plant Village Dataset. From the table it can be concluded that the proposed CNN architecture yields superior results as compared to other works. Detection of diseases is a major purpose of this current work. Detection of plant diseases using Deep Learning based CNN has achieved great success in the Agricultural Industry for reducing the risk of infected diseases [166]. The authors in [167] for detecting Apple leaf diseases proposed CNN based model with GoogleNet Inception structure using Rainbow concatenation based on SSD Architecture, which generated detection performance of 78.80% mAP with a high detection speed of 23.13 FPS. The detection of the following 10 class Tomato leaf disease was formulated using F-RCNN. Some of the detected class images are shown in Figure 4.24. The detection image consists of the individual annotated

class names, bounded boxes and the detection scores. The detection scores and bounded boxes define how perfectly each class is detected. An intersection over union score of 0.95 is obtained while processing it. The present work is also utilized in real time detection of tomato leaves to draw a major conclusion of the proposed work. In this paper we have used some more dataset and Figure 4.25 shows some of the detected images of a single class in multiple positions to prove the major utility of the presented work.

**Table 4.12 Comparison of existing pre-trained DL classifiers with the proposed work using Plant Village Dataset**

<b>Pretrained CNN Classifier</b>	<b>Accuracy (%)</b>	<b>Precision (%)</b>	<b>Recall (%)</b>	<b>Inference Time (ms)</b>
VGG 16	97.86	97.19	97.19	2.15
Inception V3	98.20	98.10	98.10	1.98
InceptionResnet V2	99.01	98.45	97.34	4.60
Resnet152V2	98.60	97.41	97.41	3.67
<b>PCA DeepNet</b>	<b>99.60</b>	<b>98.55</b>	<b>98.49</b>	<b>1.21</b>

**Table 4.13 Hyper-parameters of the Machine Learning Algorithms**

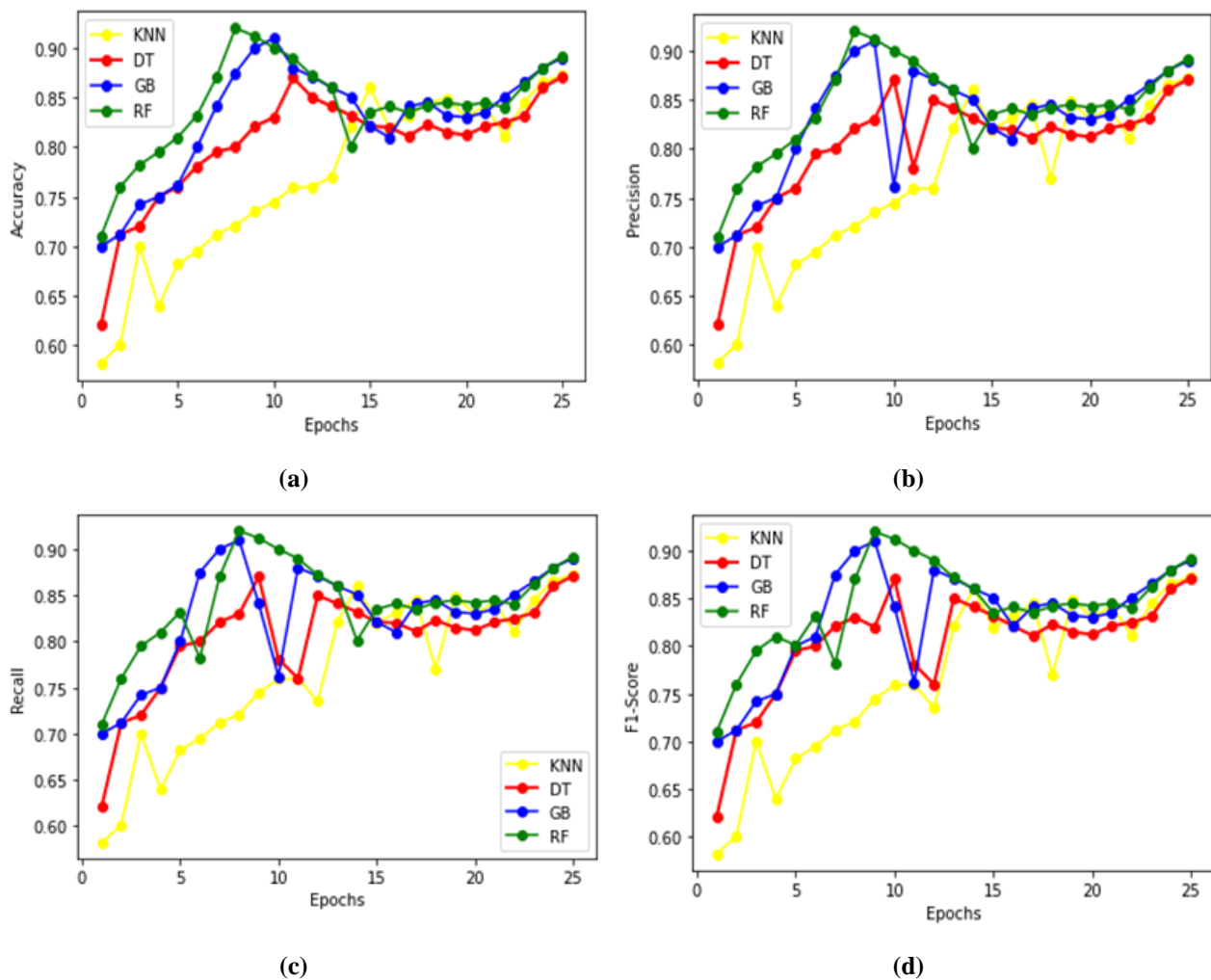
<b>Machine Learning(ML) Algorithms</b>	<b>Hyper-parameters</b>
K-Nearest Neighbor	Leaf_size = 5, p=1, n_neighbors = 7
Decision Tree	Max_depth = 200
Random Forest	n_estimators = 200, random_state = 5, max_depth = 200
Gradient Boosting	n_estimators = 200, random_state = 5, max_depth = 200

**Table 4.14 Comparison of existing pre-trained DL classifiers with the proposed work using Plant Village Dataset**

<b>ML Algorithms</b>	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>
K-Nearest Neighbor (KNN)	0.767	0.767	0.767	0.767
Decision Tree (DT)	0.807	0.807	0.807	0.807
Gradient Boosting (GB)	0.830	0.830	0.830	0.830
Random Forest (RF)	0.842	0.842	0.842	0.842

**Table 4.15 Comparison of existing pre-trained DL classifiers with the proposed work using Plant Village Dataset**

Different Techniques	Accuracy (%)	Precision (%)	F1-Measure (%)
MobileNetV2 [6]	99.30	-	-
CNN [10]	99.70	-	98.49
InceptionV3+DCGAN [22]	92.60	-	-
VGG-INCEP [13]	97.14	78.80	-
SSD-InceptionV2 [3]	-	73.07	-
ResNet-34 [29]	97.2	-	96.5
<b>Proposed PCA DeepNet</b>	<b>99.60</b>	<b>98.55</b>	<b>98.5%</b>



**Figure 4.23 Performance evaluation of the different Machine Learning Classifiers on the Plant Village Dataset**



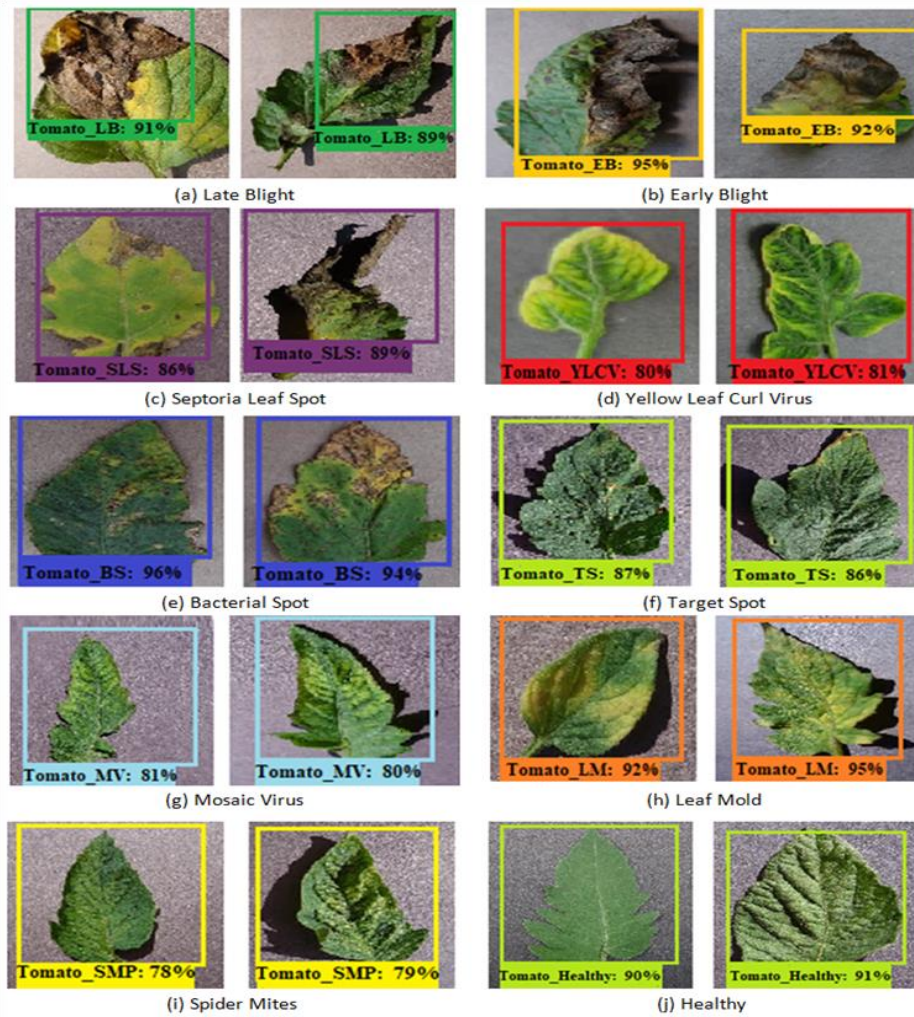


Figure 4.24 Detected images of all the classes

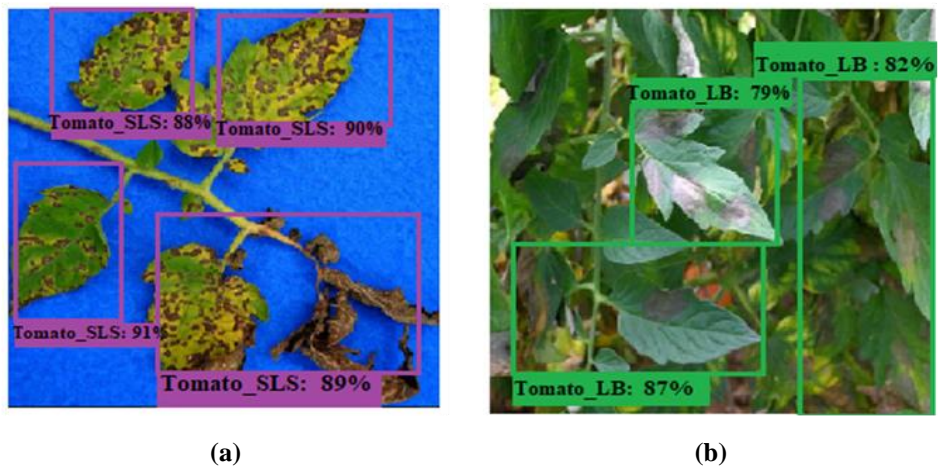


Figure 4.25 Detected image of single class in Multiple position

The novelty of the pipeline is methodized using GANs, PCA, CNN and F-RCNN hence the name is suggested as PCA DeepNet which is used for the classification as well as detection of Tomato Leaf diseases. The entire framework is a hybridized structure of all the modules mentioned with appropriate and optimized customization. The classifier is designed to carry out a specific task of classifying each and every disease efficiently. The main challenge is to reduce the time complexity of the entire model so as to make it very reliable for proper classification process. The presented work proposes a framework where proper detection of Tomato Leaf Diseases is done. Hence, a detection process is introduced with required changes in different parameters of F-RCNN model for efficient detection of the diseases. The integration of these two helps in fulfilling the desired framework. The main objective of the current work is to classify the images based on different diseases and then detect the exact disease. This is methodized owing to the fact that if it is automated into a system then this can be very beneficial for the farmers and other people associated with agricultural work. The system will automatically analyze the different Tomato Leaf Disease images and can detect the diseases within a fraction of second. Thus this can serve a better purpose for the early detection of the Tomato Leaf diseases so as to prevent excessive loss that would have been otherwise occurred due to these diseases. This would thus strengthen agro-based industries for better production.

The presented work has huge utility in the agricultural domain; however, it possesses some limitations. The architecture of the proposed work consists of 10 CNN layers which can further be reduced to 5 – 6 layers in order to reduce computational time and complexity, still maintaining a higher value of Accuracy. Moreover, the proposed method is only limited to tomato leaf diseases detection whereas in real world, several other crops exist which require same types of pipeline for disease detection. Thus the same work can be extended for other crops as well. Also the current work is only a software based framework, while its hardware can be developed for accomplishment of the same specific task which will help to resolve the problem easily. Hardware interfacing for real time applications is encouraged to have a finished product for smart agriculture and growth of agro-based industries

## **4.10 CONCLUSION**

Agriculture has secured a great place in our day-to-day lifestyle and plays a significant role in the economic growth of the country. India, being an agricultural country, employs about 60% of its population in the agricultural sector. Thus, it is very crucial to have healthy and disease free crops. The implementation and deployment of modern methods using Deep learning and Computer vision in agriculture facilitates early detection of diseases which is very important for implementing remedial solutions and sustaining agro-based industries. The proposed work using PCA DeepNet makes disease detection very fast and accurate thereby eliminating the factor of human error. The major utility of the present work lies in helping farmers and other people engaged with agriculture to eradicate great losses incurred due to pest and other bacterial invasion on crops. Thus easy detection of such situations can enhance production to a greater extent. In near future this work can be automated into a real time system so that this can directly help farmers and others associated with agriculture. Moreover it can also be implemented as pest resistant equipment in large nurseries to detect early arrival of the diseases for its recovery. It is thus concluded that this novel framework creates a greater scope in enhancing its quality of analyzing and detecting different diseases in a better way than any other state-of-the-art.

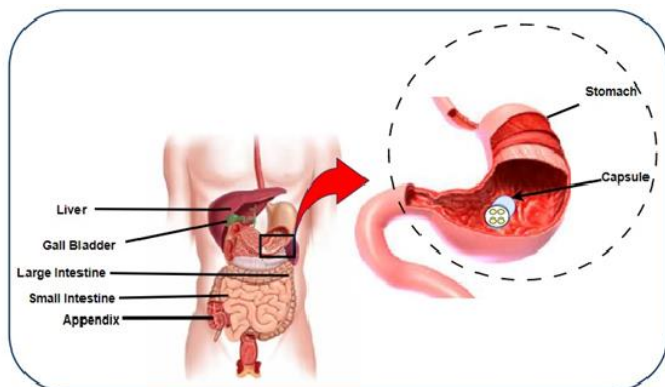
## **4.11. IMPORTANCE OF AI / ML IN BIO-MEDICAL IMAGE ANALYSIS**

In the field of medical science, the novel technique involving the use of Endoscopy has revolutionized the approach of diagnosing intestinal diseases, including small bowel disorders and other problems, for Gastroenterologists. The gastrointestinal tract comprises the stomach, liver, pancreas, gall bladder, small intestine, and large intestine, as shown in Figure 4.26. Unlike Gastro-Duodenoscopy, Capsule Endoscopy (CE) enables a more detailed investigation of conditions like Celiac disease. CE is also effective in diagnosing hereditary polyposis syndrome, small bowel tumors, and intestinal damage caused by non-steroidal anti-inflammatory drugs. With the use Endoscopy, healthcare professionals can assess the entire digestive system through an average of 60,000 generated frames. This technology has significantly advanced the diagnosis and understanding of various intestinal diseases, leading to improved patient care and treatment outcomes. The Wireless Capsule Endoscopy (WCE) detection can turn out to be more

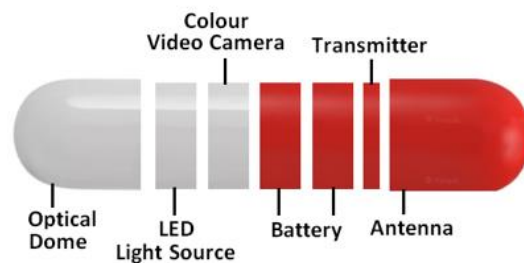
challenging at times [168]. The advancement of deep learning helps to resolve the challenges and is proven to predict more accurately; even better than those interpreted by radiologists [169]. Evolving facts have shown that WCE is a very reliable, efficient and non-invasive method for examining small intestinal anomalies [170-171]. Early detection of diseases such as polyps [172], ulcers [173], bleedings [174], crohns [175-177] and tumors [178-179] help in reducing the complications and widens the scope of treatment. The advantages of early detection lie not only in curing or controlling the spread of these diseases, but also prevent them from getting worse leading to cancer or other fatal diseases. For analysis and findings of these diseases in their early stages, numerous indirect technologies have evolved for gastro- intestinal (GI) tract disease detection viz., ultrasonography, X-radiography, scintigraphy, angioplasty, etc. However, these techniques have been reported with low diagnostic yields [180-181]. Examination with the traditional invasive wired endoscopic method does not allow the whole GI tract for diagnosis owing to its non-penetration in the small intestine acting as dead zone. Moreover, the wired endoscopic method increases the risk of intestinal damage and high probability of cross contamination. These limitations are overcome through WCE based detection.

Capsule endoscopy is a non-invasive technique hence has a lot of advantages for extracting gastrointestinal disorder images. Moreover, it is an effortless process as it only requires a capsule which automatically captures the images. Although wireless capsule endoscopy is bit expensive as compared to normal endoscopy, still its uniqueness in allowing painless imaging causes it to replace the traditional endoscopy technique. Primarily the capsule ingredients comprise four light emitting diodes, a color camera, CMOS imager lens, batteries, antennas and a radio-frequency transmitter [182-183]. However, variations in the arrangement of capsule ingredients differ from different manufacturers. Figure 4.27 shows the mixture of different components which is present in a basic capsule used for endoscopy. Typically, eight skin antennas are tied to the anterior abdominal wall of the patients before the capsule is swallowed. GI tract images are captured while the capsule moves inside and the captured images are sent through radio-frequency transmitters and sensors to the data logger. Images are finally downloaded into a computer as video images after completion of the entire internal procedure [184]. The evolving prosperity of this new technology has shown quality progress in terms of high- resolution images and higher frame rate. Hence, analysis and localization of the GI tract abnormalities with the pill camera

generated digital images becomes easier than processing its analog image counterparts. In this paper, seven different types of GI tract anomalies have been considered viz., Crohn’s disease, Erosion, Tapeworm, Vascular Ectasia, Polyp, Esophagitis and Ulcerative-Colitis. The digital endoscopic images of the mentioned irregularities are chosen and are used for detection via a proposed framework named WCE Entention (Encoder-Attention) DeepNet.



**Figure 4.26 Digestive system along with the capsule in the GI tract**



**Figure 4.27 Different components of Capsule**

GI bleeding [185-187] within the digestive tract is a serious problem. The bleeding region may indicate a small amount of blood loss to life threatening hemorrhage. Hence, premature recognition of a bleeding region will help out a patient from its serious consequences. Inflammation in the digestive tract when experienced, leads to severe diarrhea, fatigue, malnutrition, weight loss and abdominal pain. Crohn, a kind of inflammatory bowel disease (IBD), is taken here into consideration for automatic identification exploiting machine learning techniques. The painful Crohn’s disease sometimes causes life-risking complications spreading into the deeper layers of the bowel. Nonetheless, long-term diminution and healing of the inflammation can be accomplished if signs and symptoms of the disease are discovered at an early stage [188-190]. The stomach and duodenum are often affected under some pathologic condition known as erosion. Essentially it is a severe lesion that brings out a prompt reaction [191-192]. In many cases, erosions cannot be identified and located by naked eyes and hence detection of the affected region with machine intelligence is found to be very efficient. A major percentage of the world population suffers from ulcer. A mucus coating is present in the digestive tract which gives protection from acid. If the mucus reduces and the acid increases then ulcers are formed [193-194]. Ulcer is found majorly in the duodenum or lining of the stomach.

Peptic ulcers are basically sores and an individual suffering from chronic peptic ulcers is diagnosed for Peptic Ulcer disease (PUD). Negligence with the kind of pathology concerning ulcers results in histologic damage to the patient. Iron deficiency anemia or gastrointestinal bleeding occurs due to vascular ectasia. In the pyloric antrum, the dilated small blood vessel results in intestinal bleeding which is often termed as watermelon stomach or honeycomb stomach. The examination is mainly based on endoscopic images. Treating the disease with endoscopic therapy has proved to be efficient and safer rather than surgery. The therapy is termed as Argon Plasma Coagulation which can be considered for patients as first-line treatment with vascular ectasia related bleeding [195-197]. About 50 million of people worldwide are infected with tapeworm from beef or pork meat. Several studies have shown that consumption of under-cooked beef, eating raw liver, and drinking contaminated water are the main reasons for tapeworm infestation. The regular clinical symptoms are abdominal pain, nausea, anorexia and modification in appetite. Tapeworm causes several damages to the intestine as well as abnormal gut motility [198-200]. In this case, some- times stool microscopy fails to detect whereas CE provides a promising result in identification of the eggs present.

The impact of Deep Learning has remarkably changed the dimension of medical diagnosis and has certainly uplifted the world into another level [201]. Deep Convolutional Neural Network (DCNN) [202] acquired a great expertise in identification of diseases. Several new techniques in deep learning are evolving especially for the detection of Endoscopic images. Authors in [203] have used Modified Salp Swarm Algorithm infused with Deep Learning for classifying different gastrointestinal diseases. The hybrid fusion of features using pre-trained architectures like VGG-16, SVM and DenseNet121 with Artificial Neural Network [204] plays a remarkable role in diagnosing different GI anomalies. ResNet101 is employed by authors in [205] to detect and classify GI abnormalities. Some approaches also include denoising CNN [206] models to classify different Gastrointestinal disorders. Thus, Deep learning has helped to easily detect and classify various GI anomalies. The rise of Deep learning has played a key role in massive transformation to recent solutions of serious problems. A large integration of newer CNN models finds great applications in each and every domain irrespective of the type of data that the proposed work is dealing with. Different deep learning frameworks, mainly Convolutional Neural Networks (CNN), are being utilized extensively to generate best possible results. CNN

are used for classification and detection of object [207-208] in each and every domain. An intelligent method to classify different alimentary disorders like esophagitis, polyps, hemorrhoids and ulcerative colitis integrating empirical wavelet transform (EWT) and CNN was shown by the authors in [209]. The implementation of CNN in medical disease diagnosis for classification of diseases is re-ported by the authors in [210]. Transfer learning [211], another effective technique of classifying different anomalies and then detecting it for health monitoring systems, bridges the gap and enhances the accuracy as compared to conventional machine learning algorithms. Recent methods in detection are also proposing meta learning [212] for anomaly detection using fundus images, requiring a more balanced dataset for computation. In [213], a contrastive lifelong learning model for image anomaly detection was implemented using vision transformers (ViT) for extracting the visualization. The amalgamation of feature encoder, deep learning and machine learning in machine health surveillance systems [214] stimulates better analysis of data for detection of anomalies. Experimental studies are performed by the authors in [215] for generating high-quality images through multi-scale dilated CNN. The amalgamation of two-stage CNN along with an autoencoder was addressed for detecting anomalies [216]. The following method was implemented to enhance the utility of the presented CNN over conventional CNN and autoencoder. The work in [217] presents the practice of utilizing Convolutional Neural networks on image dataset for the diagnosis of diabetes using iris dataset. Moreover, the signal processing operations are done with several pre- trained CNN models namely ResNet-101, Xception and EfficientNet [218] for performing classification of different time- series data. Time-frequency analysis using high precision CNN model for classifying biomedical health disorders like cardiac arrhythmia was shown in [219]. CNN and Bi-directional Long-Short Term Memory (BiLSTM) [220] networks are also highly processed for assessing ECG signal qualities. Thus, rigorous employment of Convolutional Deep networks is prevalent in every sphere of work.

Several segmentation algorithms have been implemented for the detection of pathologies in the digestive system and are widely available in the literature [221-233]. In [221], the authors highlight tumor, polyp and ulcer as major anomalies present in the GI tract. They emphasized WCE imaging as a fast and involuntary pathway for abnormal findings in the video frames. They portrayed computer aided highly sophisticated machine algorithms as booming efficient methods

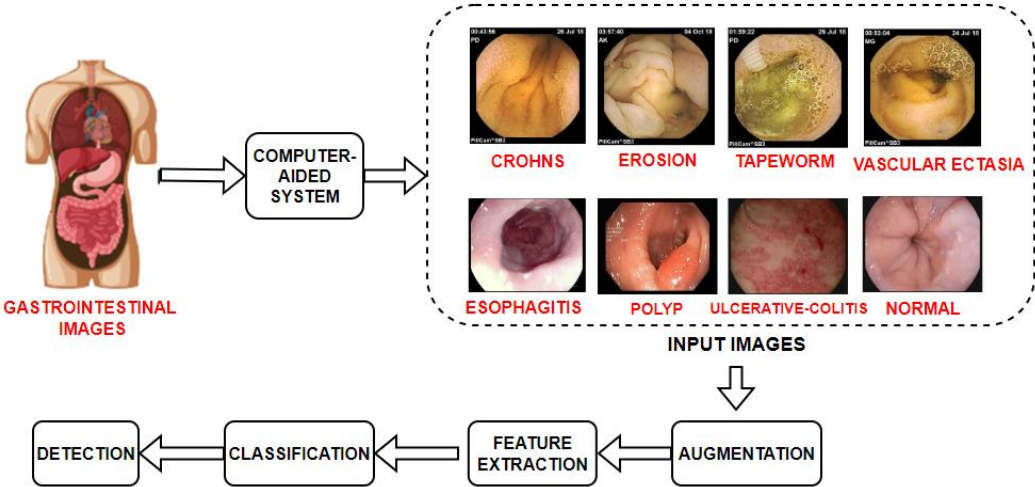
for anomalies detection with high accuracies and also handling huge data with less complexity. In the segmentation algorithm reported in [222], texture and color-based descriptors are applied for identification of pathologies in the WCE frames. Herein, the Vector Supported Convex Hull algorithm is compared against SVM taking two different feature selection methods in consideration. The block-based segmentation method is implemented for segmenting out the bleeding regions [223]. The algorithm encompasses HSI model's average saturation skewness and kurtosis as features for the SVM classifier. In [224], the authors brought out the popularity of Machine Learning with the aid of computer vision for exploration of disease identification with highly sophisticated algorithms in terms of precision and recall. They revealed that automation for the analysis of the GI tract video is an instance of this possibility. In [225], WCE is reflected as a new technology for examining the GI tract; hence detailed information of the orientation of the endoscopic capsule is much needed while traveling in the GI tract. A novel procedure based on computer vision speed estimation analyzing the consecutive video frame is proposed and established to get the speed information of the capsule. In [226], bleeding region detection through capsule endoscopic images is accomplished considering the color information. The shape and size of the blood spot regions are also considered for validation of the localized area. In [227], the authors dealt with colon polyps as another major pathology, reported to occur in 1.2 million new cases in 2008. The colon capsule endoscopy is depicted as an emerging non-invasive procedure as an alternative to the traditional colonoscopy. Processing the colonoscopy images with computer vision techniques yields good results extracting quality traits. In the paper, gradient features histogram and multilayer perceptron neural networks have been realized for performing the task of distinguishing the polyp regions. In [228], Frame Abnormality Index (FAI) using densities of training and testing data are reported. Thus, the abnormal frames containing abnormalities can be uncovered easily. The burden of handling huge data (55000 frames of RGB color data) produced by WCE can be easily handled with the aid of digital image processing algorithms designed for automatic functioning and identification of abnormalities in the GI tract [229]. Segmentation of mucosa regions in the WCE frames and videos are undergone using Split Bregman method and generate promising results. In [230], segmentation of lesions (Crohn's disease) has been reported with the implementation of the MPEG-7 visual descriptor and Hanalick texture feature. The MPEG-7 comprises adaptation of Dominant Color Descriptor (DCD), Homogeneous Texture Descriptor (HTD) and Edge Histogram Descriptor (EDH). In



[231], WCE has been exploited with many algorithms viz., SVM and neural network classifiers producing more than 90% accuracy. The severity of disease like Crohn’s can be graded with an index like CEC-DAI (Capsule Endoscopy Crohn’s Disease Activity Index) as described in [232]. The CEC-DAI score as analyzed may serve as a reproducible diagnostic and reliable method for the use of the endoscopists. Thus, machine learning methodologies are very popular nowadays for analysis in the domain of medical imaging.

**4.12. METHODOLOGY**

Image segmentation and detection are prime research areas in the field of digital imaging. Researchers of various domains have addressed image segmentation for a plethora of applications. A number of major issues are related to digital images with special emphasis on medical imaging like uneven intensity distribution, poor contrast, ambiguous boundaries, poor illumination etc. which makes the problem even more difficult to analyze [233-236]. Hence WCE detection confirms a major diagnostic modality in recognizing the diseases [237]. Eight classes of the WCE images are taken namely Vascular Ectasia, Tapeworm, Erosion, Crohns, Polyp, Esophagitis, Ulcerative-colitis along with the Normal ones. The entire workflow is methodized using a novel framework Encoder-Attention DeepNet which is named as WCE-Entention-DeepNet that comprises of Data acquisition, Data preprocessing, Data Augmentation, Feature Extraction, Classification and Detection as shown in Figure 4.28.



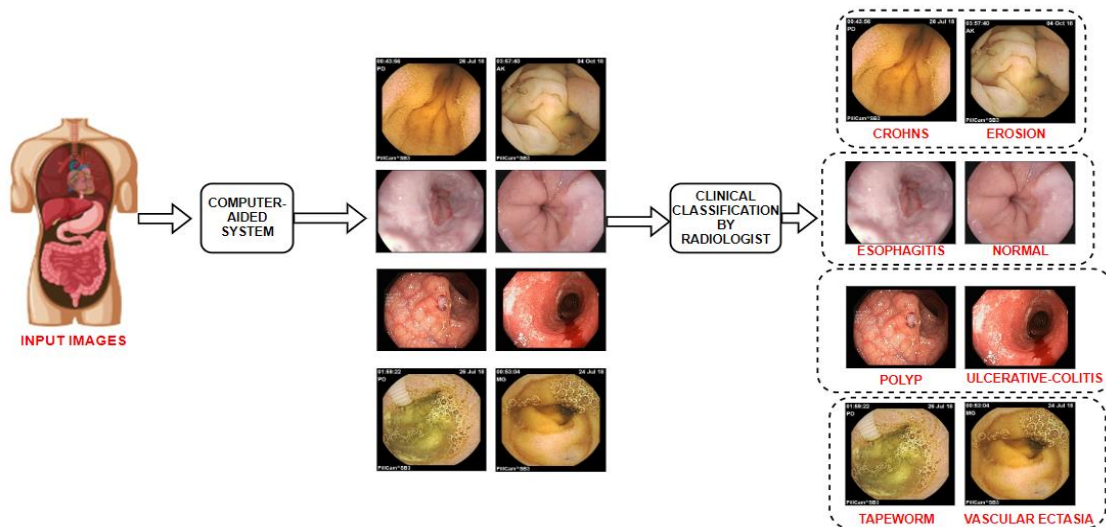
**Figure 4.28 Proposed Workflow Diagram**

### **4.12.1 Data Acquisition**

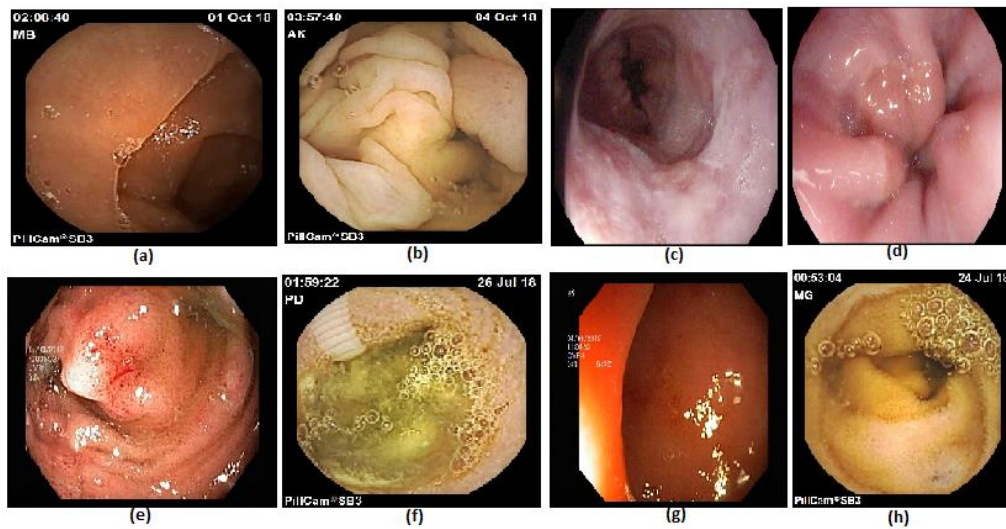
WCE is becoming one of the leading methods in diagnosing gastrointestinal problems. The study is usually done using a camera embedded capsule. This capsule passes through the entire GI tract and captures various internal images of the throat, esophagus, stomach, small intestine, large intestine, rectum and anus. This entire process helps in collection of data for further processes and is beneficial for several research works. Data Acquisition is a technique of extracting the raw data. The present dataset is collected partially from [238] and partially from an open source platform named Kaggle [239] and each of the diseases is grouped separately. The dataset is basically clinically classified by a radiologist for smooth labeling of the diseases. The different orientation of the disease images helps for easy identification of different classes. The extracted data is then preprocessed for further continuation of the work. The quality of the images is brilliant and hence gives an easy outlet to process for subsequent stages. Figure 4.29 shows the entire block diagram of data acquisition through a computer aided system.

### **4.12.2 Dataset**

Endoscopy is a procedure of looking inside the internal organs of the Human body. This technique helps the medical practitioners and researchers to easily look into the types of diseases a person is suffering from. The different GI tract disorders create critical problems for which detection and classification of these diseases are of utmost need in the present world. Such detection is only possible with a good mixture of data to enable an automated detection device. The current dataset is a combination of data obtained from [238] and from an open source platform Kaggle [239]. The dataset is organized into separate groups based on different diseases. The total images taken for the experiment is nearly 3448 which is divided among eight different classes respectively. The dataset is grouped separately into different classes namely Vascular Ectasia, Tapeworm, Erosion and Crohn, Polyp, Esophagitis, Ulcerative-colitis along with the Normal ones. The entire dataset images are shown in Figure 4.30.



**Figure 4.29 Data Acquisition**



**Figure 4.30 Dataset images (a) Crohns (b) Erosion (c) Esophagitis (d) Normal (e) Polyp (f) Tapeworm (g) Ulcerative-colitis (h) Vascular Ectasia**

### 4.12.3 Data Preprocessing

The collected data which is used as input images are pre-processed to make it trainable with the proposed model. The whole dataset consisting of 3448 images is split into training and validation sets in a ratio of 70:30, where each set consists of all the 8 classes. After the segregation of the dataset, each and every image of all the classes is annotated by identifying the characteristic feature of the particular disease of each individual class. The schematic diagram of data-

preprocessing process used in this work is depicted in Figure 4.31. Detailed information about the pre-processed of each class is given in Table 4.16. The graphical representation of the overall data pre-processing technique is shown in Figure 4.32. The images are labeled using bounding boxes; the most commonly used type of annotation in object detection and localization tasks, some of which are shown in Figure 4.33. After labeling the images, the annotation data obtained as XML files are further converted into TensorFlow Records for smooth performance in object detection. The annotation naming done per classes is enlisted in Table 4.17.

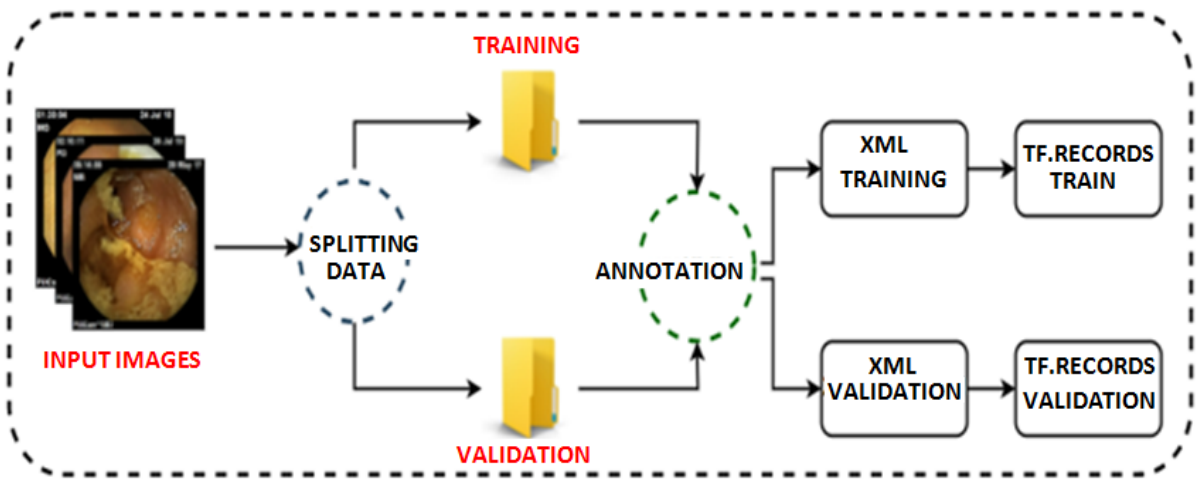
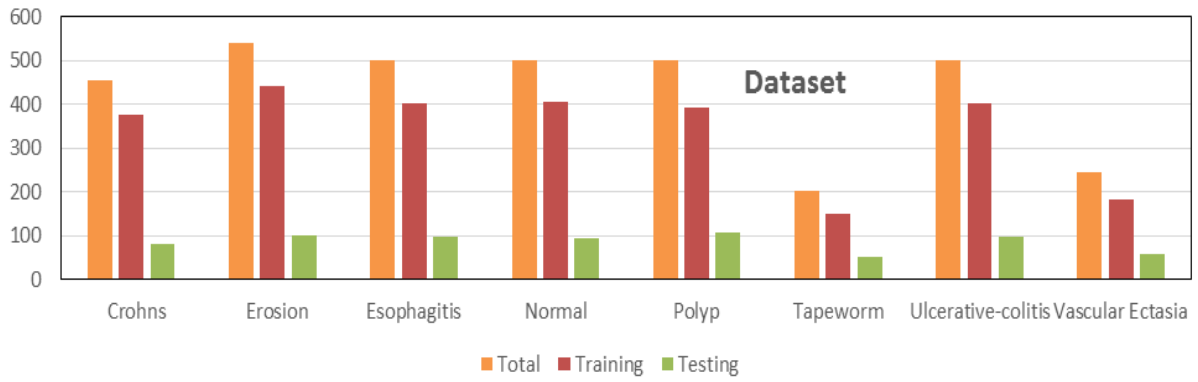


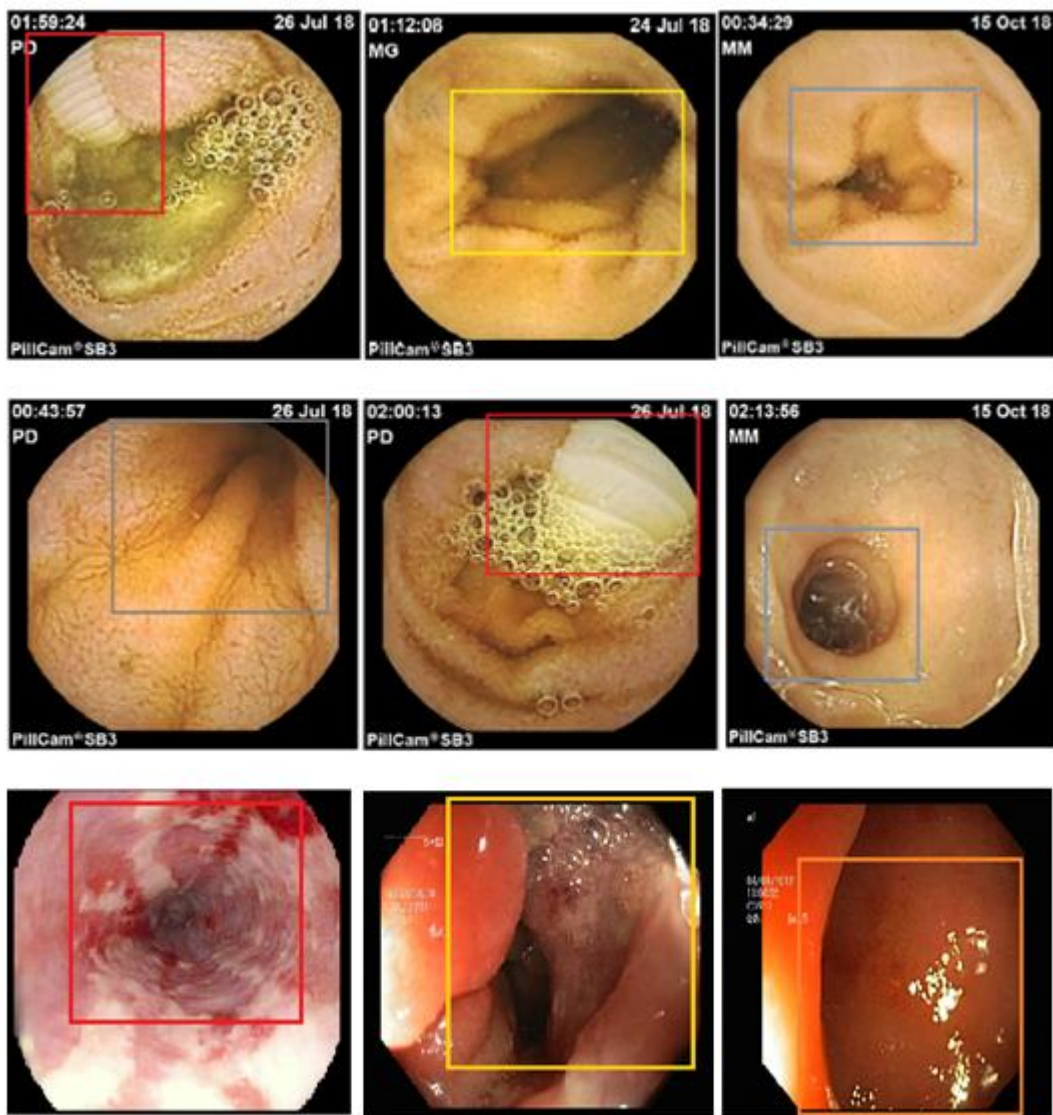
Figure 4.31 Schematic diagram of Data Preprocessing

Table 4.16 Total images in the Dataset

WCE Dataset	Total Images	Training Images	Testing Images
Crohns	457	376	81
Erosion	542	442	100
Esophagitis	500	402	98
Normal	500	406	94
Polyp	500	392	108
Tapeworm	203	152	51
Ulcerative-colitis	500	402	98
Vascular Ectasia	245	185	60



**Figure 4.32 Graphical Representation of the Original Dataset**



**Figure 4.33 Annotated WCE images**

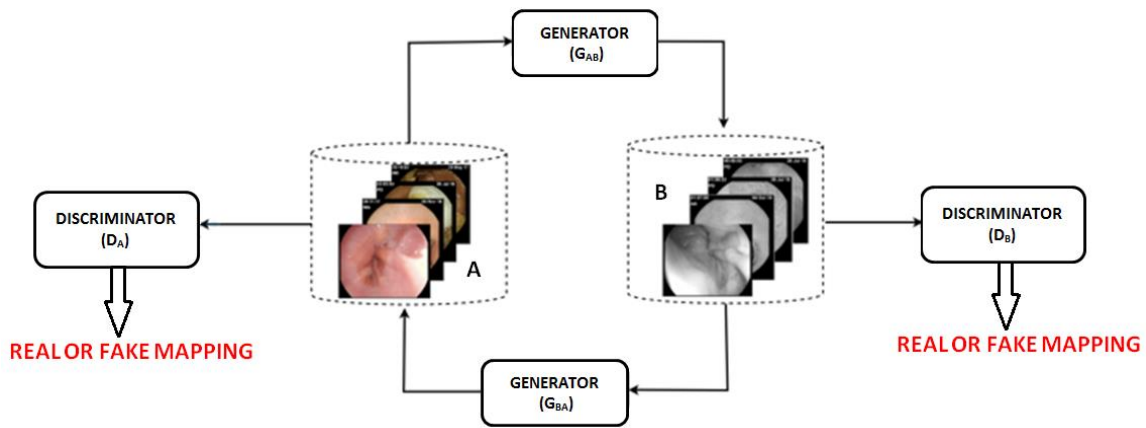
**Table 4.17**      **Detail information of Data Annotation per class**

<b>Wireless Capsule Endoscopy Diseases</b>	<b>Annotated images</b>
Crohns	WCE_Crohn
Erosion	WCE_Erosion
Esophagitis	WCE_Esophagitis
Normal	WCE_Normal
Polyp	WCE_Polyp
Tapeworm	WCE_Tapeworm
Ulcerative-colitis	WCE_UlcerativeC
Vascular Ectasia	WCE_VascularE

#### **4.12.4 Image Augmentation**

Data augmentation is a technique of refining the dataset which facilitates the training of classification models. In WCE, the embedded capsule is inserted directly which is indeed not a painful process, so it is not difficult to collect enough image samples. In the field of deep learning, small sample size and data imbalance are major factors leading to poor recognition and classification. The data augmentation is applied to artificially increase the amount of data by generating new data points from existing ones. Here GANs have been used as a modern approach to data augmentation. Unlike any other conventional augmentation models, GAN aims at learning the distribution of the training dataset to generate new (synthetic) data instances. The GAN model comprises two sub-models: generator and discriminator, which work against each other. The discriminator is trained on both real and fake data. It learns to get better at distinguishing the generated fake data from the real ones and the generator learns to generate more realistic new data points from random inputs. The process continues until the generator can create data instances such that the discriminator cannot distinguish it from real data. Out of the various types of GANs suitable for different purposes, CycleGAN is chosen here owing to its suitability for image augmentation. The most important feature of CycleGAN is that it can perform image translation on unpaired images where there is no relation between input and output images. A diagrammatic representation of the above process is shown in Figure 4.34.





**Figure 4.34 Image augmentation using CycleGANs**

#### 4.12.5 Feature Extraction

Feature Extraction is a technique that provides an alternate way of modeling the data into a tabular format. The arrangement helps to reduce the dataset into a valuable key feature that contains transformed data extracted from the raw dataset. It also helps to reduce the amount of redundant data from the data set. Apart from overfitting risk reduction, feature extraction has other added advantages like accuracy improvement, improved data visualization and reduced runtime of the model. In this paper Autoencoder is utilized for extracting the features. Features are often necessary to scale up the information which may get lost while handling large amounts of data. Using deep learning a semi-supervised feature extraction method using Convolutional autoencoder is generated that can overcome the problem of losing essential features from the data. Figure 4.35 shows the diagrammatic representation of the autoencoder. The figure represents the presence of a hidden feature which is mainly used to filter out noises instead of removing the relevant features.

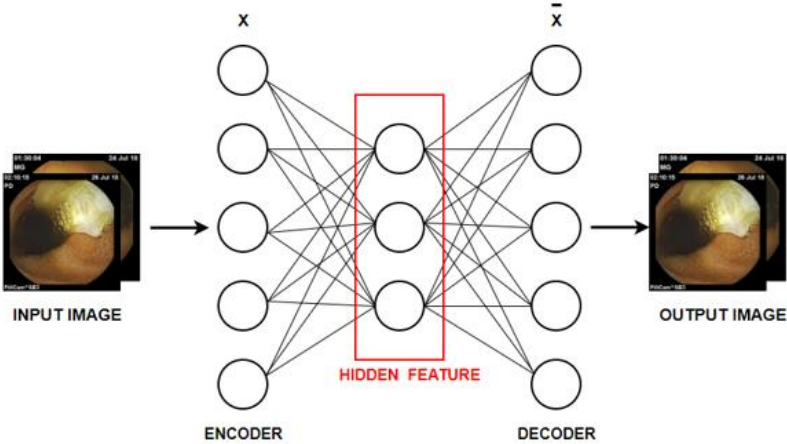
#### 4.12.6 Classification

Classification is a major technique of classifying different classes of images. The current WCE-Enttention-DeepNet framework proposes an attention-based CNN model for classifying the 8 different classes of WCE images namely Normal, Vascular Ectasia, Tapeworm, Crohns, Erosion, Esophagitis, Polyp and Ulcerative-Colitis. The present work mainly highlights a VGG 16 network which uses attention modules to exactly differentiate different classes accurately. The

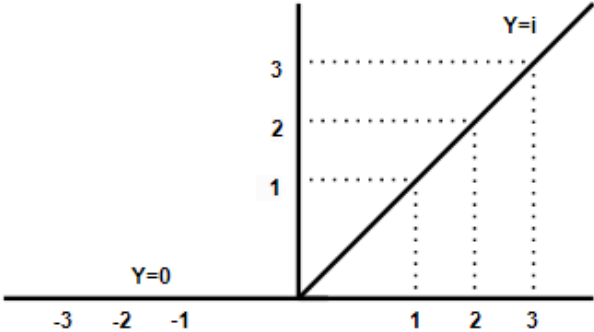
VGG networks contain convolutional neural networks along with other layers. An input image of tensor  $224 \times 224 \times 3$  indicating the size of the images is fed into the architecture. The work uses mostly  $3 \times 3$  convolutional layers with an activation function ReLU to support the model. Figure 4.36 shows the graphical representation of the activation function ReLU.

$$Y = \max(0, i) \dots\dots\dots (4.8)$$

Here, Y is the output depicted in equation 4.8 of the function performed by ReLU. The output function points to zero for all negative values whereas for the positive ones it remains constant.



**Figure 4.35 Feature extraction using Auto encoder**



**Figure 4.36 ReLU Function**

The model contains a VGG network but only the dense layers are removed. This approach reduces images and thereafter the feature vectors are computed via global average pooling and are concatenated together to form the final feature vector, which serves as the input to the classification layer. The feature vectors are generated from the auto encoder mentioned in the above segment. The feature vectors are the output and the global feature vector acts as the input



to the attention layer. Both the feature vector and the global feature vector pass through a series of CNN layers; hence actual classification of different WCE images takes place. In the current work Categorical Cross-Entropy is used as the loss function, the formula for the same is referred in equations 4.9 and 4.10 respectively. Equation 4.9 is for the Softmax activation function and equation 4.10 stands for Cross Entropy.

$$f(s)_i = \frac{e^{s_i}}{\sum_{j=1}^C e^{s_j}} \dots\dots\dots (4.9)$$

$$CE = -\sum_{i=1}^C t_i \log(f(s)_i) \dots\dots\dots (4.10)$$

Here f(s) is the function; C is the class on which the probability needs to be calculated. The letter t stands for the target vector. In this work, Adam is used as an optimizer and Softmax is used as the Activation function.

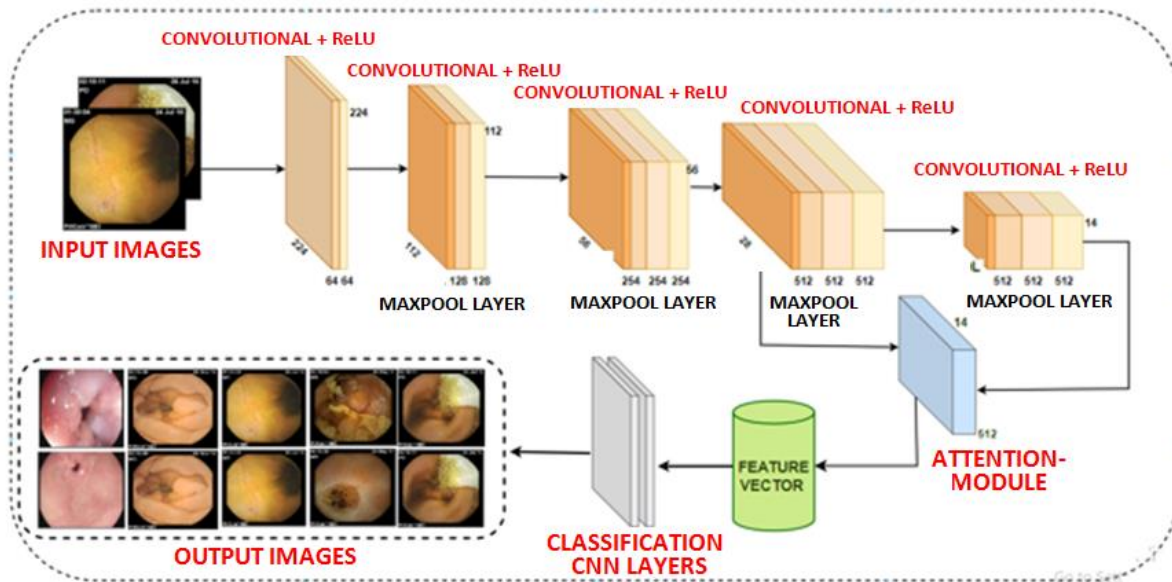
$$v_t = \beta_1 v_{t-1} - (1 - \beta_1) g_t \dots\dots\dots (4.11)$$

$$s_t = \beta_2 s_{t-1} - (1 - \beta_2) g_t^2 \dots\dots\dots (4.12)$$

$$\omega_t = -\eta \frac{v_t}{\sqrt{s_t + \epsilon}} g_t \dots\dots\dots (4.13)$$

$$\omega_t = \omega_t + \delta \omega_t \dots\dots\dots (4.14)$$

Adam uses the combination of heuristics of both momentum and RMSProp. Equations 4.11 – 4.14 show the mathematical basis of Adam works in which  $\eta$  is Initial learning rate,  $g_t$  is gradient at time t along  $\omega^j$ ,  $v_t$  is exponential average of gradients along  $\omega^j$ ,  $s_t$  is exponential average of square of gradient along  $\omega_t$  and  $\beta_1, \beta_2$  are the hyper parameters. Thus, in the current work, the reduction of the extra dense layer in the Attention based CNN provides a lesser computational time; hence, helping in easy compilation with the limited available resources. The improved classification is obtained by tuning the hyper parameters and the loss function supports the work diligently which is shown in Figure 4.37. The proposed model performs much better than other models discussed later.



**Figure 4.37** Diagrammatic representation of the Attention based CNN classifier

#### 4.12.7 Evaluation Metrics

The presented WCE-Entention-DeepNet uses an attention- based CNN model and the performance of it is evaluated based on several metrics which are calculated using the True positive and True negative ( $T_p$  and  $T_n$ ) and False positive and False negative ( $F_p$  and  $F_n$ ) values obtained from the confusion matrix while training the models. These are mainly the Accuracy, Precision, Recall, F1-Score and Cohen kappa score. The presented work solves the problem of multi class classification. The values of  $T_p$ ,  $T_n$ ,  $F_n$  and  $F_p$  decide the parametric metrics results. In case of multi-class classification, the values of the True positives ( $T_p$ ), True Negatives ( $T_n$ ), False Positives ( $F_p$ ) and False Negatives ( $F_n$ ) are determined as shown in Figure 4.38. Equations 4.4 – 4.7 represent the formulas of Accuracy, Precision, Recall and F1-Score respectively.

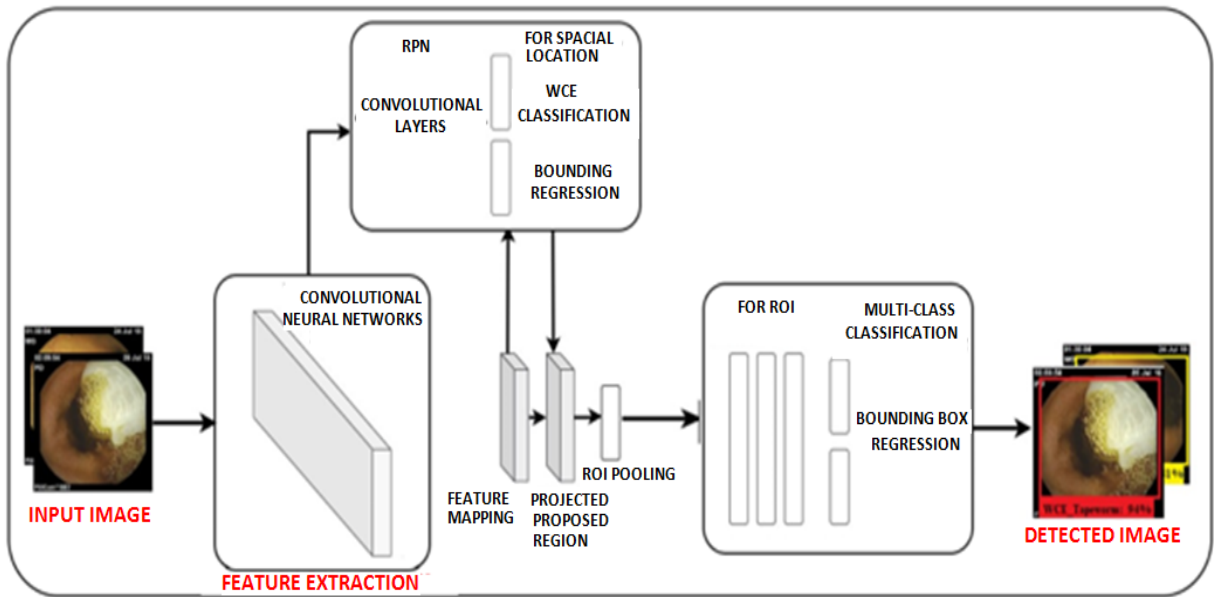
#### 4.12.8 Detection

The last part of the overall framework comprises the detection section carried out using F-RCNN. Figure 4.39 shows the block diagram of the detection architecture where fine-tuned backbone architecture is fed for further detection. The specialty of this architecture is that instead of selective search it uses two networks – one for Region proposal and another for object

detection. The F-RCNN Network comprises three parts. The first part constitutes the Convolutional layers which are useful for feature extraction as they contain filters in them which help to capture diverse information in microscopic images. The convolutional networks naturally consist of convolutional layers, pooling layers and fully connected layers which are beneficial for classification of images. The convolution is mainly done by sliding filters along the input image which generates a 2D matrix termed as Feature map. The pooling is performed by reducing the number of features by eliminating the low pixel values. The second part consists of a RPN which decides whether there is an object for detection and also predicts the bounding box of that precise object. The last part is composed of a fully connected neural network that takes input to predict the classes and the bounding box around for detecting the precise object which in our case are the affected WCE images. These layers help in easy detection and classification of WCE images and the normal images. The model is further fined-tuned using several hyperparameters for the detection of different diseases in the WCE images. Faster R-CNN evaluates 4 losses namely RPN classification, RPN regression, Fast RCNN Classification and Fast RCNN Regression. The architecture of F-RCNN has a high IoU score of 0.9 for a threshold score of 0.8. The test time consumed per image for the detection purpose is 1 sec which gives a concrete conclusion of its excellence. Moreover, it shows that the detection technique generates the results at 10x speed as compared to other detection architectures.

		Predicted class				
		NORMAL	DISEASE_1	DISEASE_2	DISEASE_3	DISEASE_4
Actual class	NORMAL					
	DISEASE_1		True Negatives (TN)		False Positives (FP)	TN
	DISEASE_2				False Positives (FP)	
	DISEASE_3		False Negatives (FN)		True Positives (TP)	FN
	DISEASE_4		TN		FP	TN

**Figure 4.38 Confusion Matrix for Multi-Class Classification**



**Figure 4.39** Detection Architecture of F-RCNN

### 4.13. RESULTS

A detailed and an exhaustive study have been carried out while validating the novel WCE-Enttention-DeepNet that uses an Attention-based Deep Learning classifier model. The overall training has been processed using Google Collaboratory with GPU specification of Tesla K80 (2496 CUDA cores). The hardware and software specifications are enlisted in Table 4.18. The work has been computed using Keras and Tensorflow in python. The pre-processed augmented data of eight classes are taken as input and different classification metrics are generated while training it. The entire architectural description of the classifier is enlisted in Table 4.19 respectively.

**Table 4.18** Hardware and Software Specifications

Configuration	Value
CPU	Intel core i5 8 <sup>th</sup> Generation
GPU	Tesla K80 (2496 CUDA cores)
Hard Disk	HDD (1 TB)
Operating System	Windows 10
RAM	8 GB

**Table 4.19** Details of the Attention-based classifier used in WCE-Enttention-DeepNet

Layer	Kernel	Stride	Input Size	Output Sizes
Conv1_64	1	3 x 3	224 x 224 x 3	224 x 224 x 64
Conv1_64	1	3 x 3	224 x 224 x 64	224 x 224 x 64
Maxpool	2	2 x 2	224 x 224 x64	112 x 112 x 64
Conv2_128	1	3 x 3	112 x 112 x 64	112 x 112 x 128
Conv2_128	1	3 x 3	112 x 112 x 128	112 x 112 x 128
Maxpool	2	2 x 2	112 x 112 x 128	56 x 56 x 128
Conv3_256	1	3 x 3	56 x 56 x 128	56 x 56 x 256
Conv3_256	1	3 x 3	56 x 56 x 256	56 x 56 x 256
Conv3_256	1	3 x 3	56 x 56 x 256	56 x 56 x 256
Maxpool	2	2 x 2	56 x 56 x256	28 x 28 x 256
Conv4_512	1	3 x 3	28 x 28 x 256	28 x 28 x 512
Conv4_512	1	3 x 3	28 x 28 x 512	28 x 28 x 512
Conv4_512	1	3 x 3	28 x 28 x 512	28 x 28 x 512
Maxpool	2	2 x 2	28 x 28 x 512	14 x 14 x 512
Conv5_512	1	3 x 3	14 x 14 x 512	14 x 14 x 512
Conv5_512	1	3 x 3	14 x 14 x 512	14 x 14 x 512
Conv5_512	1	3 x 3	14 x 14 x 512	14 x 14 x 512
Maxpool	2	2 x 2	14 x 14 x 512	7 x 7 x 512
Fc	-	1 x 1	1 x 1x 25088	1 x 1x 4096
Fc	-	1 x 1	1 x 1x 4096	1 x 1x 4096
Fc	-	1 x 1	1 x 1x 4096	1 x 1x 1000

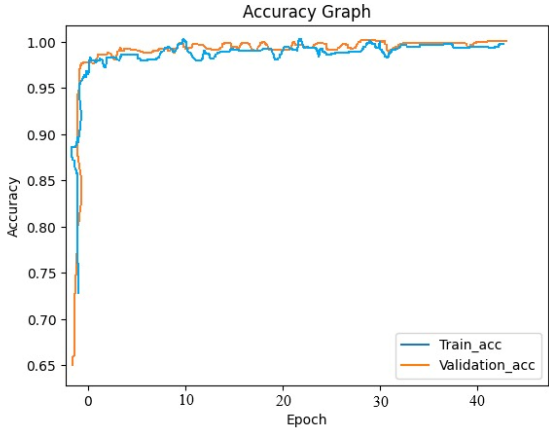
### A. Experiment using Attention-based CNN

The experiment is carried out using attention-based CNN classifier which throws light to the overall framework effectively. The presented process is trained using Adam optimizer having a learning rate of 0.001,  $\beta_1$  of 0.9,  $\beta_2$  of 0.999 and  $\epsilon$  of 1e-08. The validation done is for 50 epochs with a batch size of 32. The accuracy graph and loss graph shown in Figure 4.40 and Figure 4.41 supports the fact that a good fitted graph is generated when an augmented dataset was utilized. The confusion generated from the graphs in Figure 4.42 proves that a balanced dataset is more likely to be used in case of deep learning. The precision-recall curve for the attention-based classifier is shown in Figure 4.43 respectively. The rise in the values of accuracy, precision, recall and F1-score are enlisted in Table 4.20. The augmentation process is applied to all the classes. The total numbers of images per classes is listed above. Same are divided into train and test respectively and the training images were utilized to obtain the confusion matrix

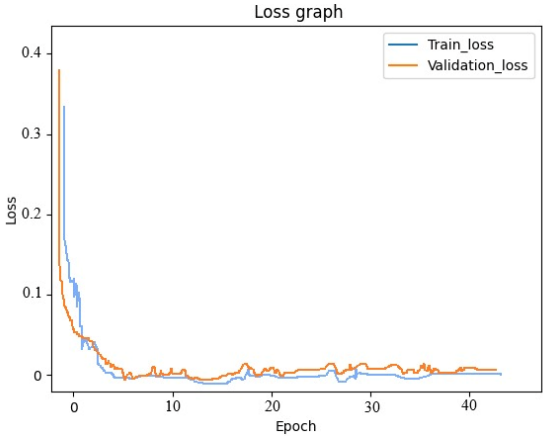
respectively. Figure 4.44 represents the ROC curve generated by the classifier. The graph highlights the true positive and the false positive rates for the Attention based CNN which further concludes the true prediction of the presented work. The graphical representations of the parameters generated by different classes are depicted in Figure 4.45 respectively. The proposed model proves that attention-based classifier generates fair scores while classifying the eight different classes of wireless endoscopy diseases.

**Table 4.20 Results of Attention-based CNN**

WCE Dataset	Accuracy	Precision	Recall	F1- Score
Crohns	0.989	1.00	0.91	0.95
Tapeworm	1.00	1.00	1.00	1.00
Vascular Ectasia	1.00	1.00	1.00	1.00
Erosion	0.989	0.93	1.00	0.97
Esophagitis	0.99	0.99	1.00	0.99
Normal	0.988	0.95	0.97	0.96
Polyp	0.97	0.93	0.90	0.92
Ulcerative-colitis	0.981	0.93	0.94	0.93



**Figure 4.40 Accuracy graph of the Attention-based CNN**

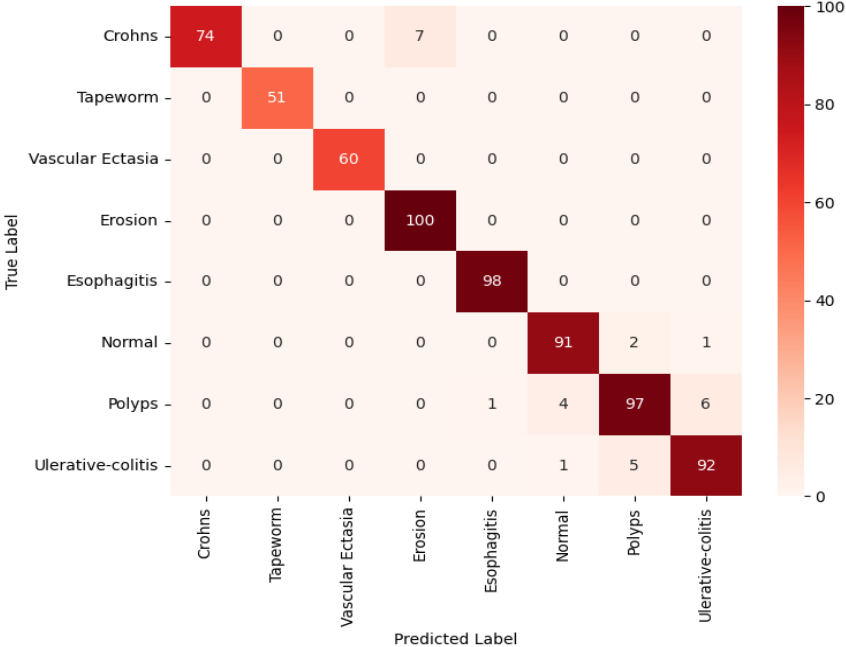


**Figure 4.41 Loss graph of the Attention-based CNN**

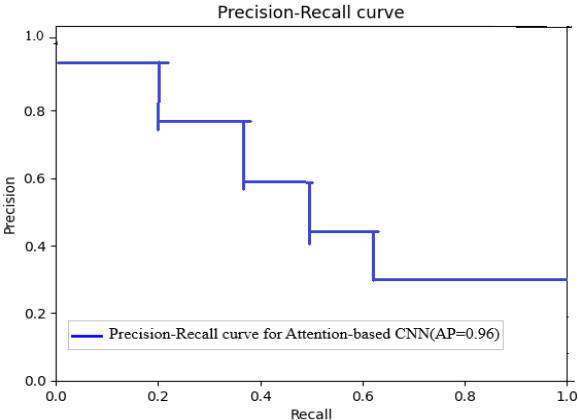
**B. Performance evaluation of different pre-trained Deep Learning models**

This section contains some pre-trained DL models validated with the augmentation dataset. The performance metrics are listed in Table 4.21 and the graphical representation of the same is shown in Figure 4.46. The overall analysis shows that the pre-trained models also serve as good classifiers; however, the Attention based CNN is still far better as compared to it. The

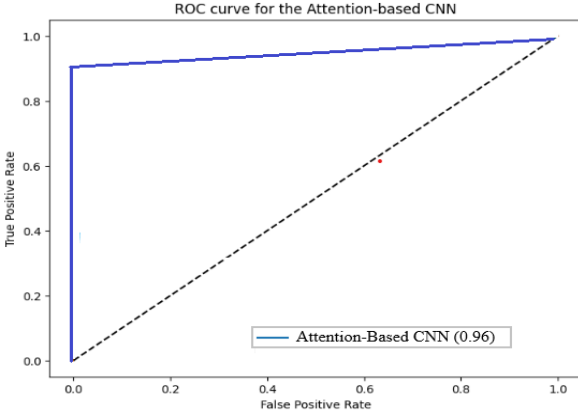
performance generated by them provides promising results and even proves that the pre-trained DL architectures are also good for WCE image classification.



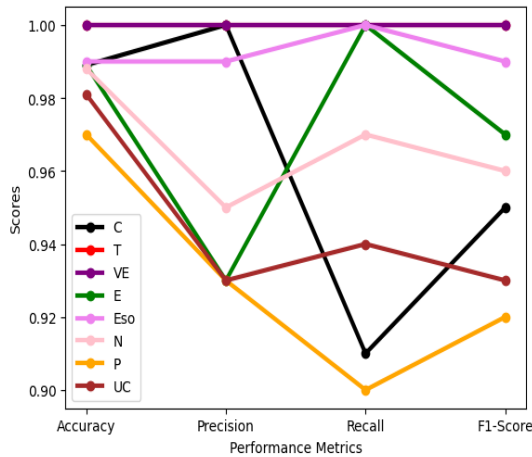
**Figure 4.42 Confusion matrix of the Attention-based CNN**



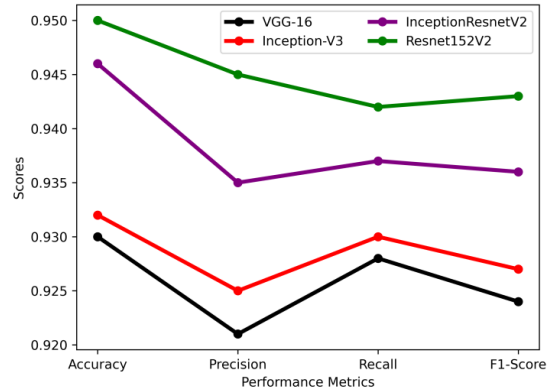
**Figure 4.43 Precision-Recall curve for the Attention-based CNN**



**Figure 4.44 ROC Curve for the Attention-based CNN**



**Figure 4.45** Graphical representation of parameters using Attention-based CNN.



**Figure 4.46** Graphical representation of results for DL Classifiers

(C: Crohns, T: Tapeworm, VE: Vascular Ectasia, E: Erosion, Eso: Esophagitis, N: Normal, P: Polyp, UC: Ulcerative-colitis)

**Table 4.21** Comparison of different performance parameters generated using state-of-art DL Classifiers and proposed model

Pre-trained DL Classifier	Accuracy	Precision	Recall	F1-Score
VGG-16	0.930	0.921	0.928	0.924
Inception V3	0.932	0.925	0.930	0.927
InceptionResnetV2	0.946	0.935	0.937	0.936
Resnet152V2	0.95	0.945	0.942	0.943
WCE-Entention-DeepNet	0.988	0.966	0.965	0.965

### C. Evaluation of performance parameters

The WCE dataset is again preprocessed using several machine learning algorithms. The hyper-parameters of the selected machine learning are changed accordingly and are listed in Table 4.22. The different algorithms evaluate results based on performances and are enlisted in Table 4.23. The use of machine learning models is done to compare with the Attention based CNN model. The implementation of K-Nearest Neighbor is done using three hyper-parameters namely Leaf\_size, probability value and n\_neighbors. The Decision tree is used by incorporating a Max\_depth of 200. The n\_estimator of 200 indicates that 200 decision trees will be created under the Random Forest algorithm. In case of Gradient Boosting a n\_estimator of 200, random\_state



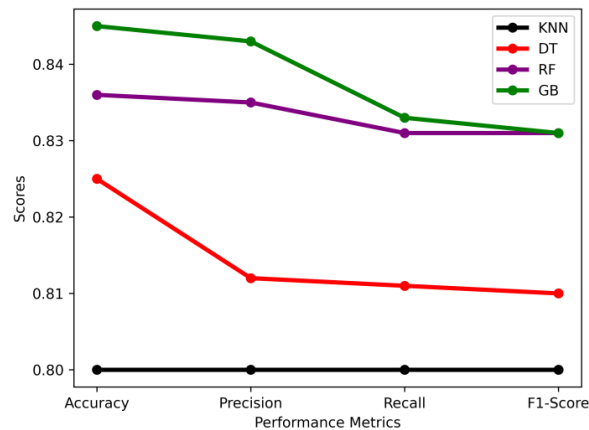
of 5 and a max\_depth of 200 is utilized. The exhaustive study summarizes that Machine learning models performed well on WCE dataset for classification of images. It is especially observed from the results that Gradient Boosting, Random Forest and Decision Tree performed better than K-Nearest Neighbor. This again confirms that the tree-based models are more efficient for classification of WCE images than other algorithms. The graphical representation of the results is shown in Figure 4.47.

**Table 4.22 Hyper-parameters of the Machine Learning Algorithms**

ML Algorithms	Hyper-parameters
K-Nearest Neighbor	Leaf_size=5,p=1,n_neighbors=7
Decision Tree	Max_depth=200
Random Forest	n_estimators=200,random_state=5,max_depth=200
Gradient Boosting	n_estimators=200,random_state=5,max_depth=200

**Table 4.23 Performance parameters obtained from Machine Learning Algorithms**

ML Algorithms	Accuracy	Precision	Recall	F1-Score
K-Nearest Neighbor (KNN)	0.80	0.80	0.80	0.80
Decision Tree (DT)	0.825	0.812	0.811	0.810
Random Forest (RF)	0.836	0.835	0.831	0.831
Gradient Boosting (GB)	0.845	0.843	0.833	0.831



**Figure 4.47 Graphical representation of results for ML classifiers**  
(KNN: K-Nearest Neighbor, DT: Decision Tree, RF: Random Forest, GB: Gradient Boosting)

## 4.14. DISCUSSION

This section proposes an in-depth analysis of the entire work done in the current experiment. The presented WCE-Enttention-DeepNet framework comprises of five sections: Data Acquisition, Augmentation, Feature Extraction, Classification and Detection. The detailed study focuses on the novel attention-based CNN model which is utilized for classification of Endoscopy images of 8 classes. The overall process depends on the Accuracy, Precision, Recall and F1-Score obtained from the proposed model. The performance scores obtained from the Attention-based CNN model used in the novel framework WCE-Enttention-DeepNet have been experimented in two ways – one with the imbalanced data and the other with the balanced ones. The Attention-based classifier resulted in overall accuracy of 98.8%, Precision of 96.6%, Recall of 96.5% and F1-Score of 96.5% respectively. Thus, the experiment throws light into the fact that attention-based classifier enhances the performance score effectively. The study also concludes that a good fitted accuracy graph is generated when augmented dataset is used. The paper proposes a detailed study with the other available deep pre-trained models. Some of the famous classifiers like VGG 16, InceptionV3, InceptionResnetV2 and Resnet152V2 were taken into consideration. The above-mentioned classifier generated a good amount of performance metrics in the form of accuracy, precision, recall and F1 score respectively. The exhaustive study helped to draw a conclusion that the proposed Attention based CNN classifier using augmented dataset generated an enhanced accuracy of 4% more than the DL classifier. Thus, the novelty of the architecture is justified in this case. Machine Learning models are also used for the classification of different WCE diseases, hence, some of the popular ML algorithms are chosen for testing the same using the augmented dataset. Although it is observed that some Machine Learning models like Gradient Boosting, Random Forest and Decision Tree performed well when utilized for classification of WCE images, still the performance is much more enhanced when Attention Based CNN classifier is applied for classification of different classes of images. There is a sharp rise of 15% in accuracy when compiled using the WCE dataset. The entire study supports the fact that the amount of data plays an effective role in smooth classification as well as accurate detection of diseases. Hence the work clarifies a way in which smooth classification and detection can be obtained after proper augmentation of the images. The efficacy and performance of the proposed WCE-Enttention-DeepNet model lies in resolving the problem of multi class classification which is often found in case of WCE images. Till now, as reported in literature in

section I, in case of multi class classification category only 3 classes have been taken whereas in the present work the robustness of the output results has been enhanced to another level by generating 8 classes of classified outputs. The information regarding the same is listed in Table 4.20. The detailed work is concluded with the detection of different WCE images as shown in Figure 4.48. The detection images diagnose the exact location of the anomaly in the Different WCE diseases. The attention-based CNN classifier is fed as backbone architecture for the same and the detection is effectively materialized. The detection results in an IoU score of 0.9 which proves the major utility of the presented work. The work even out performs the other state-of-the-art algorithms as evident from the performance parameters of the present model enlisted in Table 4.24. Moreover, the generated results were also verified by a medical practitioner for its actual authentication and validation in order to establish the correctness of the detected results. The presented work can even be automated into a real time system for smooth detection of WCE anomalies. The presented model implies better performance of the presented framework for detection of GI tract anomalies using WCE-Enttention-DeepNet. India being one of the largest populated countries often suffers from lack of medical facilities and early diagnosis of different diseases – hence a real-time automated system like this can reduce the problem of disease detection. Thus, this can be of immense help to the medical practitioners and common people suffering from deadliest diseases to get its earliest cure easily. Hence the novel WCE-Enttention-DeepNet framework presented can be very beneficial for the people for the detection of the gastrointestinal disorder.

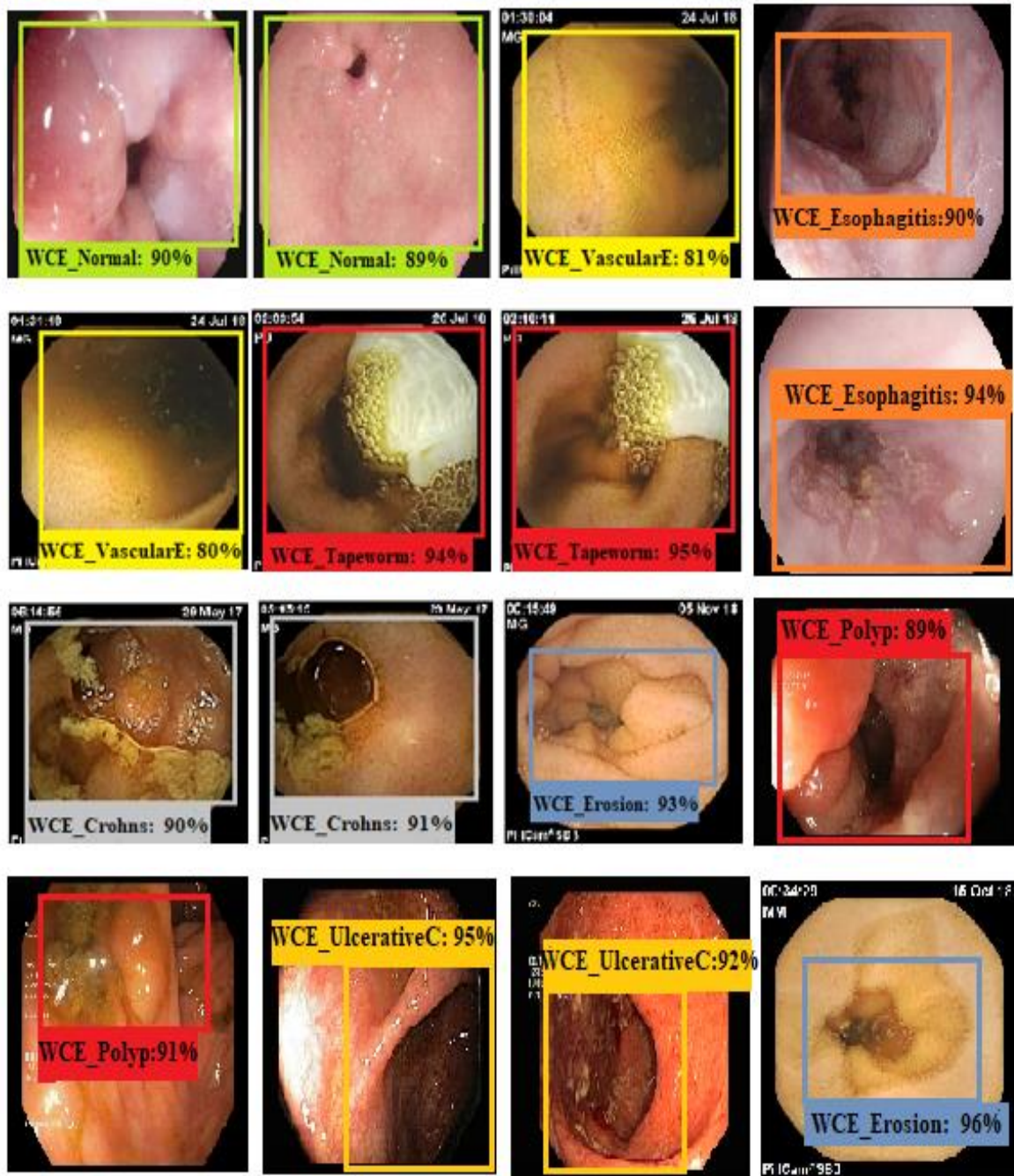


Figure 4.48 Detection results generated by F-RCNN

**Table 4.24 Comparison of the proposed work with other state-of-art works reported in literature**

References	Vascular Ectasia	Esophagitis	Tapeworm	Crohns	Polyp	Erosion	Ulcerative -colitis
Detection of Crohns in WCE images using deep CNN [8]	-		-	A=97.64% R=98.98%	-	-	-
Detection of Crohns in WCE images using deep NN [9]	-	-	-	A=93.5%	-	-	-
Detection of Crohns using Conv. Recurrent Attention Neural Network [10]	-	-	-	P=93.7% R=93%	-	-	-
Comparison of Crohns disease in WCE and CT [22]	-	-	-	-	-	A=90.8% R=88.2%	-
Utility of WCE for Crohns disease [23]	-	-	-	-	-	A=95.34% R=93.67%	-
Detection of Gastric Vascular Lesions in WCE using CNN [50]	A=93.7% R=86.2% P=96.4%	-	-	-	-	-	-
Modified Salp Swarm Algorithm (MSSADL-GITDC) [36]	-	A=97.75% P=93.16% R=88.50% F=90.77%	-	-	A=98.0% P=89.81% R=95.10% F=92.38%	-	A=98.56% P=92.92% R=96.10% F=94.48%
ANN (VGG+SVM) [37]	-	A=95.50% P=93.60% R=95.34%	-	-	A=95.50% P=96.0% R=95.10%	-	A=94.0% P=93.50% R=93.85%
ANN(DenseNet-121+SVM) [37]	-	A=96.5% P=96.50% R=96.35%	-	-	A=96% P=96.50% R=96.09%	-	A=93.50% P=92.60% R=94.41%
Proposed work	A=100% R=100% P=100% F=100%	A=99.9% R=99.9% P=100% F=99.99%	A=100% R=100% P=100% F=100%	A=98.9% R=100% P=91% F=95%	A=97% R=93% P=90% F=92%	A=98.9% R=93% P=100% F=97%	A=98.1% R=93% P=94% F=93.5%

A: Accuracy, R: Recall, P: Precision, F: F1score

## 4.15. CONCLUSION

Intestinal diseases are a significant concern in gastroenterology and Wireless Capsule Endoscopy (WCE) has been recognized as a valuable clinical tool for their detection. In this study, the authors proposed a novel framework, named WCE-Enttention-DeepNet, for detecting various types of intestinal diseases using WCE images. The framework aims to identify classes such as Vascular Ectasia, Tapeworm, Crohn's disease, Erosion, Esophagitis, Polyp, Ulcerative-Colitis as well as Normal cases. The study primarily utilizes an Attention-Based Convolutional Neural Network (CNN) classifier, which achieves an impressive average accuracy of 98.80%. Additionally, detection techniques are applied using the Faster Region-Based Convolutional Neural Network, resulting in an Intersection over Union score of 0.9. This framework demonstrates its potential for automatic classification and detection of WCE images, significantly reducing the time required for analysis. While the presented work focuses on seven classes of Intestinal diseases in the gastrointestinal tract and including the normal case, it is worth mentioning the inclusion of an additional class, namely Bleeding. By incorporating this class, a total of eight different Endoscopy diseases can be considered, enabling comprehensive insight and precise detection of specific regions in endoscopic images. Looking ahead, future research can explore optimizing the framework using different optimizers and adopting innovative approaches to enhance performance. The presented work can also be transformed into a real-time system for direct detection of various endoscopy diseases. Moreover, by leveraging Internet of Things (IoT) devices, the framework can be utilized remotely, enabling individuals to seek medical advice from doctors through the internet. This methodology holds potential for efficient and accurate detection of different endoscopic disorders, offering a valuable tool for medical practitioners in their diagnostic efforts.

# Chapter-5

---

## CONCLUSION AND FUTURE SCOPE

### 5.1. CONCLUSION

Deep learning, a subset of machine learning, has the potential to bring about significant societal benefits in various domains. Its utility can be observed in the areas of healthcare, autonomous vehicles, natural language processing, environmental monitoring, manufacturing and quality control, security and surveillance, education, agriculture, disaster response, language translation and communication, etc.

In the present research work, few of the areas viz., healthcare, horticulture, agriculture and maritime surveillance are focused with the various novel deep learning models. The DL models are primarily based on classification and thereafter followed by localization. Classification is an important step for categorization of classes whether it be semantic segmentation, normal classification or detection.

OCT-based retinal images have been examined for the purpose of identifying a particular disease. The OCT retinal images are used as the input data for six different CNN models, which produce classified results. According to Figure 2.11, the output is correctly classified for each of the four classes of retinal images. Except for the model Inception v3 without transfer learning, the proposed model with and without a pre-trained data set has provided good accuracy percentages. Therefore, it follows from the current research that models that incorporate transfer learning produce more accurate results than otherwise. The saliency maps and occlusion maps at the CNN architecture's output layers also display how the output is categorized based on the region of interest.

Fresh apples are distinguished from rotten ones through real-time semantic segmentation. Two architectures, UNet and En-UNet, are fed with the RGB images of the rotten and fresh apples. Binary masks are created as ground truths before the image data is fed to the networks, and the predicted outputs are then checked against the ground truths. Both models are compared in terms of accuracy and IoU score. According to the findings, UNet's validation accuracy was 95.36%, while En-UNet's was 97.54%. Both En-UNet and UNet achieved the best mean IoU score of



0.866 and 0.66 respectively; both below a threshold of 0.95, respectively, at 0.866 and 0.66. With a quick automation system designed for the detection of high-quality fruit and the production of high-quality food products, the proposed work can be very helpful in the horticulture and fruit product manufacturing industries.

The last segment of the research is based on classification followed by localization which is often referred to as detection. Although the outcome is detection but the backbone is the optimized classification in every case. Even though there are state-of-the-art detection algorithms with deep learning but in the present research work novel hybridized framework of the deep learning models are designed and implemented. In this section three of the applications are chosen viz. maritime surveillance, agriculture and healthcare (anomaly detection with WCE images). In all the case studies, the designed models are compared with the existing state-of-the-art algorithms and are proven to be superior in terms of performance matrices.

In maritime surveillance, the CNN classifier model detects and segments ships in the water body using satellite images. The model achieves a validation accuracy of 99.5% and 85.1% for the auto-encoder, with a best IoU score of 0.77. This model can be useful in maritime security management, traffic monitoring, heavy cyclone rescue operations, and navigation management control. Future work focuses on superimposition of segmented ship images over original images and remote sensing using high resolution SAR images. The model's sophistication lies in its ability to classify, locate, and segment objects, reducing complexity in terms of time, cost, installation, and maintenance.

The second case study is the plant leaf disease detection. A hybridized framework with PCA and deep learning algorithms has been executed and named as PCA-deepNet. The proposed method makes plant disease detection extremely quick and precise, removing the possibility of human error. The primary benefit of the current work is to assist farmers and other agricultural professionals in eliminating significant losses brought in by pest and other bacterial invasion on crops. So simple detection of such diseases can increase production rate. This work can be automated into a real-time system, directly assisting farmers and other people involved in agriculture. Additionally, it can be used as pest-resistant equipment in sizable nurseries to identify diseases before they spread and thus act as aid in their recovery.

In the last case study, anomaly detection in the GI tract is carried out using WCE images. A hybridized framework is used with an attention based classifier. The WCE-Attention-DeepNet framework is a novel approach for detecting intestinal diseases using WCE images, focusing on eight classes and seven different diseases. The framework uses an Attention-Based CNN classifier, achieving an average accuracy of 98.80%. By incorporating bleeding, the framework can detect eight different endoscopy diseases, providing comprehensive insight and precise detection of specific regions. Future research can explore optimizing the framework and leveraging IoT devices for remote diagnostic assistance. This approach offers a valuable tool for medical practitioners in their diagnosis of diseases.

The proposed works deal with huge amount of data. A good mixture of data is very important to obtain good values of the performance matrices. Majorly the execution of deep neural networks faces constraints with the data sets and class imbalance problem. In order to solve class imbalance problem and to come up with good mixture of data set, data augmentation is very much essential and thus GANs are employed for each of the cases. The state-of-the-art architectures are very good at generating outputs but very high in terms of computation cost and architecture complexity. On the other hand, if customized CNN classifiers are clubbed with classical machine learning techniques for generating a hybrid model, the computation cost will be low as well as the model complexity.

## **5.2. FUTURE SCOPE**

Deep learning has emerged as a transformative technology in the field of computer vision, enabling significant advancements in tasks such as classification, segmentation, and object detection. These tasks play a crucial role in various applications ranging from autonomous vehicles and medical imaging to surveillance and natural language processing. As technology continues to evolve, the future scope of deep learning in classification, segmentation, and detection models promises even more remarkable possibilities. Future models are expected to focus on combining various modalities, such as text and sensor data in order to create more comprehensive and accurate models.

Reducing the dependence on labeled data is a significant challenge. Future research might focus on developing models that can learn with less annotated data or even weak annotations, such as image-level labels. This would significantly reduce the manual labeling effort required for training deep learning models. Self-supervised learning techniques, where models learn from unlabeled data, are gaining traction. Future research might refine these techniques to create models that can leverage large amounts of unlabeled data to improve performance on various tasks, including classification, segmentation, and detection.

Future models with few shot and zero shot learning might be designed to learn from very few examples or even from completely new classes with zero examples available during training. This is especially important when dealing with rapidly evolving domains, where obtaining large labeled datasets can be of real challenge.

To cope with changing environments and concepts, future models are likely to exhibit better adaptability. Continual learning techniques will enable models to acquire knowledge over time while avoiding catastrophic forgetting of previously learned information. As deep learning models become more complex, there is a growing need for interpretability and explain ability. Future models are likely to incorporate mechanisms that provide insights into their decision-making processes, which is crucial for applications in medical diagnosis and legal systems.

As deep learning models become larger and more resource-intensive, concerns about their environmental impact are growing. Future research will likely focus on developing more efficient architectures that achieve similar or better performance while consuming fewer resources. Future models will need to be more robust to adversarial attacks and noisy data. Research will likely explore techniques to enhance the security and reliability of deep learning models, especially in safety-critical applications. Future models could allow easier customization and adaptation to individual industries, such as healthcare, agriculture, and manufacturing.

As deep learning models influence various aspects of society, ethical and legal implications will become more pronounced. Future research will likely delve into creating models that adhere to ethical guidelines and regulatory frameworks, ensuring responsible and fair use. Classification algorithms to be optimized more in terms of layers, trainable parameters and data set so that the GPU computation requirement becomes less. The classification segmentation and detection

models which has been used in the present work in some particular applications can be implemented in other domains as the application field.

The models developed for classification, segmentation and detection of images in real life applications can be implemented in hardware resulting in finished products or can be realized and implemented in IoT frameworks.

The future scope of deep learning in classification, segmentation, and detection models is incredibly promising. Researchers and practitioners are poised to explore hybrid models, weakly supervised learning, self-supervised learning, and more so to tackle challenges and expand the capabilities of these models. With a focus on interpretability, robustness, and ethical considerations, the next era of deep learning is expected to drive transformation across industries and shape the way humans interact with technology.

# Bibliography

---

1. V. Veeramani and L. Mohan, "Image category classification using 12-Layer deep convolutional neural network," *Multimedia Tools and Applications*, May 2023, Published, doi: 10.1007/s11042-023-15631-3.
2. V. Adithya and R. Rajesh, "A Deep Convolutional Neural Network Approach for Static Hand Gesture Recognition," *Procedia Computer Science*, vol. 171, pp. 2353–2361, 2020, doi: 10.1016/j.procs.2020.04.255.
3. R. Fu, B. Li, Y. Gao and P. Wang, "CNN with coarse-to-fine layer for hierarchical classification," *IET Computer Vision*, vol. 12, no. 6, pp. 892–899, May 2018, doi: 10.1049/iet-cvi.2017.0636.
4. E. Pintelas, I. E. Livieris, S. Kotsiantis and P. Pintelas, "A multi-view-CNN framework for deep representation learning in image classification," *Computer Vision and Image Understanding*, vol. 232, p. 103687, Jul. 2023, doi: 10.1016/j.cviu.2023.103687.
5. S. Singh, "Systematic Review on Machine Learning and Deep Learning Approaches for Mammography Image Classification," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, no. 7, pp. 337–350, July 2020, doi: 10.5373/jardcs/v12i7/20202015.
6. A. Sharma and G. Phonsa, "Image Classification Using CNN," *SSRN Electronic Journal*, 2021, Published, doi: 10.2139/ssrn.3833453.\
7. S. Murugan and M. Jeyakarthic, "Optimal Deep Neural Network based Classification Model for Intrusion Detection in Mobile Adhoc Networks," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 11, no. 10-SPECIAL ISSUE, pp. 1374–1387, Oct. 2019, doi: 10.5373/jardcs/v11sp10/20192983.
8. N. Meng, E. Y. Lam, K. K. Tsia and H. K.-H. So, "Large-Scale Multi-Class Image-Based Cell Classification with Deep Learning," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 5, pp. 2091–2098, Sep. 2019, doi: 10.1109/jbhi.2018.2878878.
9. W. Mi, J. Li, Y. Guo, X. Ren, Z. Liang, T. Zhang and H. Zou, "Deep Learning-Based Multi-Class Classification of Breast Digital Pathology Images," *Cancer Management and Research*, vol. 13, pp. 4605–4617, Jun. 2021, doi: 10.2147/cmar.s312608.

10. V. Tiwari, R. C. Joshi and M. K. Dutta, “Deep neural network for multi-class classification of medicinal plant leaves,” *Expert Systems*, vol. 39, no. 8, May 2022, doi: 10.1111/exsy.13041.
11. B. K. Harsha and G. Indumathi, “Skin Segmentation at the Pixel Level Using Fully Convolutional Neural Network,” *International Journal of Science and Research (IJSR)*, vol. 11, no. 2, pp. 234–237, Feb. 2022, doi: 10.21275/sr22202155120.
12. L. Yan, B. Fan, H. Liu, C. Huo, S. Xiang and C. Pan, “Triplet Adversarial Domain Adaptation for Pixel-Level Classification of VHR Remote Sensing Images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3558–3573, May 2020, doi: 10.1109/tgrs.2019.2958123.
13. M. Khairalseed, S. Reddy, J. Song, G. Rijal and K. Hoyt, “Abstract PS3-29: Pixel-level tissue classification from ultrasound images of breast cancer and direct comparison to matched histological measurements,” *Cancer Research*, vol. 81, no. 4\_Supplement, pp. PS3-29, Feb. 2021, doi: 10.1158/1538-7445.sabcs20-ps3-29.
14. Z. Li, S. Xuan, X. He and L. Wang, “Global weighted average pooling network with multilevel feature fusion for weakly supervised brain tumor segmentation,” *IET Image Processing*, vol. 17, no. 2, pp. 418–427, Sep. 2022, doi: 10.1049/ipr2.12642.
15. Z.-Q. Zhao, P. Zheng, S.-T. Xu and X. Wu, “Object Detection with Deep Learning: A Review,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/tnnls.2018.2876865.
16. M. E. H. Chowdhury, T. Rahman, A. Khandakar, M. A. Ayari, A. U. Khan, M. S. Khan, N. Al-Emadi, M. B. I. Reaz, M. T. Islam and S. H. Md Ali, “Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques,” *AgriEngineering*, vol. 3, no. 2, pp. 294–312, May 2021, doi: 10.3390/agriengineering3020020.
17. K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, Feb. 2018, doi: 10.1016/j.compag.2018.01.009.
18. T. Nazir, A. Irtaza, A. Javed, H. Malik, D. Hussain and R. A. Naqvi, “Retinal Image Analysis for Diabetes-Based Eye Disease Detection Using Deep Learning,” *Applied Sciences*, vol. 10, no. 18, p. 6185, Sep. 2020, doi: 10.3390/app10186185.

19. X. Chen, H. Li, Q. Wu, K. N. Ngan and L. Xu, "High-Quality R-CNN Object Detection Using Multi-Path Detection Calibration Network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 2, pp. 715–727, Feb. 2021, doi: 10.1109/tcsvt.2020.2987465.
20. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.
21. M. Mateen, J. Wen, Nasrullah, S. Song and Z. Huang, "Fundus Image Classification Using VGG-19 Architecture with PCA and SVD," *Symmetry*, vol. 11, no. 1, p. 1, Dec. 2018, doi: 10.3390/sym11010001.
22. O. Ronneberger, P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18* (pp. 234-241). Springer International Publishing.
23. S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/tpami.2016.2577031.
24. T. Hassan, M. U. Akram, B. Hassan, A. Nasim and S. A. Bazaz, "Review of OCT and fundus images for detection of Macular Edema," in *Proceedings of IEEE International Conference on Imaging Systems and Techniques (IST-2015)*, Macau, pp. 1-4, 2015.
25. A. M. Bagci, R. Ansari and M. Shahidi, "A method for detection of retinal layers by optical coherence tomography image segmentation," in *Proceedings of IEEE / NIH Life Science Systems and Applications Workshop*, Bethesda, MD, pp. 144-147, 2007.
26. A. F. Fercher, C. K. Hitzenberger, W. Drexler, G. Kamp and H. Sattmann, "In-Vivo Optical Coherence Tomography," *American Journal of Ophthalmology*, vol. 116, pp. 113–115, 1993.
27. E. A. Swanson, J. A. Izatt, M. R. Hee, D. Huang, C. P. Lin, J. S. Schuman, C. A. Puliafito and J. G. Fujimoto, "In-vivo retinal imaging by optical coherence tomography," *Optics Letters*, vol. 18, pp. 1864–1866, 1993.

28. A. F. Fercher, "Optical coherence tomography," *Journal of Biomedical Optics*, vol. 1, pp. 157–173, 1996.
29. J. G. Fujimoto, "Optical coherence tomography for ultrahigh resolution in vivo imaging," *Nature Biotechnology*, vol. 21, pp. 1361–1367, 2003.
30. J. G. Fujimoto, M. E. Brezinski, G. J. Earney, S. A. Boppart, B. Bouma, M. R. Hee, J. F. Southern and E. A. Swanson, "Optical biopsy and imaging using optical coherence tomography," *Nature Medicine*, vol. 1, pp. 970–972, 1995.
31. C. Bowd, L. M. Zangwill, C. C. Berry, E. Z. Blumenthal, C. Vasile, C. Sanchez-Galeana, C. F. Bosworth and P. A. Sample, "Detecting early glaucoma by assessment of retinal nerve fiber layer thickness and visual function," *Investigative Ophthalmology & Visual Science*, vol. 42, pp. 1993–2003, 2001.
32. C. Bowd, L. M. Zangwill, E. Z. Blumenthal, C. Vasile, A. G. Boehm, P. A. Gokhale, K. Mohammadi, P. Amini, T. M. Sankary and R. N. Weinreb, "Imaging of the optic disc and retinal nerve fiber layer: the effects of age, optic disc area, refractive error, and gender," *Journal of the Optical Society of America a-Optics Image Science and Vision*, vol. 19, pp. 197–207, 2002.
33. T. Otani, S. Kishi and Y. Maruyama, "Patterns of diabetic macular edema with optical coherence tomography," *Am. J. Ophthalmol.*, vol. 127, pp. 688–693, 1999.
34. W. Drexler and J. G. Fujimoto, "State-of-the-art retinal optical coherence tomography," *Prog. Retin. Eye Res.* vol. 27, pp. 45–88, 2008.
35. S. C. Lee, M. Doug and Y. L. Aaron, "Deep Learning is Effective for Classifying Normal versus Age-Related Macular Degeneration Optical Coherence Tomography Images," *Ophthalmology Retina*, 2017.
36. C. S. Lee, A. J. Tying, N. P. Deruyter, Y. Wu, A. Rokem and A. Y. Lee, "Deep-learning based, automated segmentation of macular edema in optical coherence tomography," *Biomedical Optics Express*, vol. 8, pp. 3440-3448, 2017.
37. T. Schlegl, S. M. Waldstein, H. Bogunovic, F. Endstraßer, A. Sadeghipour, A-M. Philip, D. Podkowinski, B. S. Gerendas, G. Langs and U. Schmidt-Erfurth, "Fully Automated Detection and



- Quantification of Macular Fluid in OCT Using Deep Learning. Ophthalmology," *American Academy of Ophthalmology*, pp. 549-558, 2017.
38. D. S. Kermany, et. al., "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning," *Cell*, vol. 172, p. 1122-1131, 2018.
  39. S. P. Karri, D. Chakraborty and J. Chatterjee, "Transfer learning based classification of optical coherence tomography images with diabetic macular edema and dry age-related macular degeneration," *Biomed Opt Express*, vol. 8, pp. 579-592, 2017.
  40. R. R. Bourne, J. B. Jonas, S. R. Flaxman, J. Keeffe, J. Leasher, K. Naidoo, M. B. Parodi, K. Pesudovs, H. Price, R. A. White, T. Y. Wong, S. Resnikoff and H. R. Taylor, "Vision Loss Expert Group of the Global Burden of Disease Study, Prevalence and causes of vision loss in high-income countries and in Eastern and Central Europe: 1990-2010," *Br. J. Ophthalmol.* vol. 98, pp. 629–638, 2014.
  41. P. Romero-Aroca, "Current status in diabetic macular edema treatments," *World J. Diabetes.* vol. 4, pp. 165–169, 2013.
  42. C. B. Rickman, S. Farsiu, C. A. Toth and M. Klingeborn, "Dry age-related macular degeneration: Mechanisms, therapeutic targets, and imaging dry AMD mechanisms, targets, and imaging," *Investigative Ophthalmol. Vis.Sci.* vol. 54, ORSF68, 2013.
  43. N. Sengar, M. K. Dutta, R. Burget and L. Povoda, "Detection of diabetic macular edema in retinal images using a region based method," in *Proc. of 38<sup>th</sup> International Conference on Telecommunications and Signal Processing (TSP)*, Prague, pp. 412-415, 2015.
  44. J. Sugmk, S. Kiattisin and A. Leelasantitham, "Automated classification between age-related macular degeneration and Diabetic macular edema in OCT image using image segmentation," in *Proceedings of the 7<sup>th</sup> International Conference on Biomedical Engineering*, Fukuoka, pp. 1-4, 2014.
  45. G. Quellec, K. Lee, M. Dolejsi, M. K. Garvin, M. D. Abramoff and M. Sonka, "Three-Dimensional Analysis of Retinal Layer Texture: Identification of Fluid-Filled Regions in SD-OCT of the Macula," *IEEE Transactions on Medical Imaging*, vol. 29, pp. 1321-1330, 2010.

46. S. Naz, A. Ahmed, M. U. Akram and S. A. Khan, "Automated segmentation of RPE layer for the detection of age macular degeneration using OCT images," in *Proc. of 2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA-2016)*, Oulu, pp. 1-4, 2016.
47. D. Xiang, et al., "Automatic Segmentation of Retinal Layer in OCT Images with Choroidal Neovascularization," *IEEE Transactions on Image Processing*, vol. 27, pp. 5880-5891, 2018.
48. S. S. Parvathi and N. Devi, "Automatic Drusen Detection from Colour Retinal Images," in *Proc. of IEEE International Conference on Computational Intelligence and Multimedia Applications (ICCIMA-2007)*, Sivakasi, Tamil Nadu, pp. 377-381, 2007.
49. Y. Zheng, H. Wang, J. Wu, J. Gao and J. C. Gee, "Multiscale analysis revisited: Detection of drusen and vessel in digital retinal images," in *Proc. of IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, Chicago, IL, pp. 689-692, 2011.
50. S. Razavian, H. Azizpour, J. Sullivan and S. Carlsson, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus, OH, pp. 512-519, 2014.
51. Y. Wang, Y. Zhang, Z. Yao, R. Zhao and F. Zhou, "Machine learning based detection of age-related macular degeneration (AMD) and diabetic macular edema (DME) from optical coherence tomography (OCT) images," *Biomed Opt Express*, vol. 7, pp. 4928-4940, 2016.
52. B. Al-Bander, W. Al-Nuaimy, M. A. Al-Taei, B. M. Williams and Y. Zheng, "Diabetic Macular Edema Grading Based on Deep Neural Networks," in *Proceedings of Ophthalmic Medical Image Analysis International Workshop*, pp. 121-128, 2016.
53. F. G. Venhuizen, B. van Ginneken, F. van Asten, M. J. J. P. van Grinsven; S. Fauser, C. B. Hoyng, T. Theelen and C. I. Sánchez, "Automated Staging of Age-Related Macular Degeneration Using Optical Coherence Tomography," *Investigative Ophthalmology & Visual Science*, vol. 58, pp. 2318-2328, 2017.
54. L. Liu, S. S. Gao, S. T. Bailey, D. Huang, D. Li and Y. Jia, "Automated choroidal neovascularization detection algorithm for optical coherence tomography angiography," *Biomedical Optics Express*, vol. 6, pp. 3564-76, 2015.

55. X. Xi, X. Meng, L. Yang, X. Nie, G. Yang, H. Chen, X. Fan, Y. Yin and X. Chen, "Automated segmentation of choroidal neovascularization in optical coherence tomography images using multi-scale convolutional neural networks with structure prior," *Multimedia Systems*, 2018.
56. S. Khalid, M U. Akram, T. Hassan, A. Jameel and T. Khalil, "Automated Segmentation and Quantification of Drusen in Fundus and Optical Coherence Tomography Images for Detection of ARMD," *Journal of Digital Imaging*. vol. 31, 2017.
57. A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, pp. 84-90, 2017.
58. X. Ni, C. Li, H. Jiang and F. Takeda, "Deep learning image segmentation and extraction of blueberry fruit traits associated with harvestability and yield," *Horticult Res*. vol. 7, no. 1, pp. 1–14, 2020.
59. H. Kang and C. Chen, "Fast implementation of real-time fruit detection in apple orchards using deep learning," *Computers Electron Agricult*. vol. 168, pp. 105108, 2020.
60. X. Ni, C. Li and H. Jiang, "Blueberry harvestability trait extraction from 2D images and 3D point clouds based on deep learning and photogrammetric reconstruction. In 2020 ASABE Annual Int. Virtual Meeting (p. 1). American Society of Agricultural and Biological Engineers, 2020.
61. Y. Majeed, M. Karkee, Q. Zhang and L. Fu, "Whiting MD (2020) Determining grapevine cordon shape for automated green shoot thinning using semantic segmentation-based deep learning networks," *Computers Electron Agricult* 171:105308
62. V. Badrinarayanan, A. Kendall and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans Pattern Anal Mach Intell* vol. 39, no. 12, pp. 2481–2495, 2017.
63. H. Kang and C. Chen, "Fruit detection and segmentation for apple harvesting using visual sensor in orchards," *Sensors* vol. 19, no. 20, pp. 4599, 2019.
64. S. Jin, Y. Su, S. Gao, F. Wu, Q. Ma, K. Xu and J. Zhang, "Separating the structural components of maize for field phenotyping using terrestrial lidar data and deep convolutional neural networks," *IEEE Trans Geosci Remote Sens* vol. 58, no. 4, pp. 2644–2658, 2019.

65. D. Rong, L. Xie and Y. Ying, "Computer vision detection of foreign objects in walnuts using deep learning," *Computers Electron Agricult* vol. 162, pp. 1001–1010, 2019.
66. N. Kumari, A. K. Bhatt, R. K. Dwivedi and R. Belwal, "Hybridized approach of image segmentation in classification of fruit mango using BPNN and discriminant analyzer," *Multimedia Tools Applic* pp.1–31, 2020.
67. X. Liu, D. Zhao, W. Jia, W. Ji, C. Ruan and Y. Sun, "Cucumber fruits detection in greenhouses based on instance segmentation. *IEEE Access* vol. 7, pp. 139635–139642, 2019.
68. P. A. Dias, A. Tabb and H. Medeiros, "Multispecies fruit flower detection using a refined semantic segmentation network," *IEEE Robot Autom Lett* vol. 3(4), pp. 3003–3010, 2018.
69. J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," In Proc. of the *IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.
70. L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs.," *IEEE Trans Pattern Anal Mach Intell.* vol. 40(4), pp. 834–848, 2017.
71. H. Noh, S. Hong and B. Han, "Learning deconvolution network for semantic segmentation," In Proceedings of the *IEEE international conference on computer vision.* pp. 1520–1528, 2015.
72. C. Yang, "Colorful fruit image segmentation based on texture feature," *Adv Intell Inform Hid Multimedia Sign Process.* Springer, Singapore, pp 305–311, 2020.
73. A. S. M. Shafi, M. B. Rahman and M. M. Rahman, "Fruit disease recognition and automatic classification using MSVM with multiple features," *Int J Comput Appl.* Vol. 181(10), pp. 0975–8887, 2018.
74. A. Wajid, N. K. Singh, P. Junjun and M. A. Mughal, "Recognition of ripe, unripe and scaled condition of orange citrus based on decision tree classification," In 2018 *International Conference on Computing, Mathematics and Engineering Technologies (iCo-MET)*, pp. 1–4, IEEE, 2018.
75. N. V. Mhapne, S. V. Harish, A. S. Kini and V. G. Narendra, "A comparative study to find an effective image segmentation technique using clustering to obtain the defective portion of an apple," In 2019 *Int. conference on automation, computational and technology management*

(*ICACTM*), pp. 304–309, IEEE, 2019.

76. K. Roy, S. S. Chaudhuri, S. Bhattacharjee, S. Manna and T. Chakraborty, “Segmentation techniques for rotten fruit detection,” In 2019 *International Conference on Opto-Electronics and Applied Optics (Optronix)*, pp. 1–4. IEEE, 2019.
77. K. Roy, A. Ghosh, D. Saha, J. Chatterjee, S. Sarkar and S. S. Chaudhuri, “Masking based Segmentation of Rotten Fruits,” In 2019 *International Conference on Opto-Electronics and Applied Optics (Optronix)*, pp. 1–4. IEEE, 2019.
78. Y. Yu, K. Zhang, L. Yang and D. Zhang, “Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN,” *Computers Electron Agricul* vol. 163, pp. 104846, 2019.
79. J. Li, X. Lin, H. Che, H. Li and X. Qian, “Probability map guided bi- directional recurrent UNet for pancreas segmentation,” arXiv preprint arXiv:1903.00923, 2019.
80. Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh and J. Liang, “Unet++: A nested u-net architecture for medical image segmentation,” *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, Cham, pp 3–11, 2018.
81. Z. Zeng, W. Xie, Y. Zhang and Y. Lu, “RIC-Unet: An improved neural network based on Unet for nuclei segmentation in histology images,” *IEEE Access*, vol. 7, pp. 21420–21428, 2019.
82. Z. Luo, Y. Zhang, L. Zhou, B. Zhang, J. Luo and H. Wu, “Micro-vessel image segmentation based on the AD-UNet model,” *IEEE Access*, vol. 7, pp. 143402–143411, 2019.
83. S. Guan, A. A. Khan, S. Sikdar and P. V. Chitnis, “Fully dense UNet for 2-D sparse photoacoustic tomography artifact removal,” *IEEE J Biomed Health Inform.* vol. 24(2), pp. 568–576, 2019.
84. Y. Chen, C. Hou, Y. Tang, J. Zhuang, J. Lin, Y. He and S. Luo, “Citrus tree segmentation from UAV images based on monocular machine vision in a natural orchard environment,” *Sensors*, vol. 19(24), pp. 5558, 2019.
85. N. Häni, P. Roy and V. Isler, “A comparative study of fruit detection and counting methods for yield mapping in apple orchards,” *J Field Robot*, vol. 37(2), pp. 263–282, 2020.
86. G.V. Nardari, R. A. Romero, V. C. Guizilini, W. E. Mareco, D. M. Milori, P. R. Villas-Boas and

- I. A. D. Santos, "Crop anomaly identification with color filters and convolutional neural networks," In *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, pp. 363–369. IEEE, 2018.
87. [www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification](http://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification)
88. C. Bueger, "What is maritime security?" *Marine Policy*, vol. 53, pp. 159–164, 2015.
89. P. Iervolino, R. Guida, P. Lumsdon, J. Janoth, M. Clift *et al.*, "Ship detection in SAR imagery: A comparison study," in *Proc. IGARSS*, pp. 2050-2053, 2017.
90. S. Bruschi, S. Lehner, T. Fritz, M. Soccorsi, A. Soloviev *et al.*, "Ship surveillance with TerraSAR-X," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 3, pp. 1092-1103, 2010.
91. K. Ward, R. Tough and S. Watts, "Sea clutter: Scattering, the K Distribution and Radar Performance," *IET: Radar, Sonar and Navigation*, Michael Faraday House, Stevenage, 2013.
92. D. J. Crisp and T. Keevers, "Comparison of Ship detectors for polarimetric SAR imagery," in *Proc. IEEE OCEANS-2010*, pp. 1-8, 2010.
93. H. He, Y. Lin, F. Chen, M. Tai H and Z. Yin, "Inshore ship detection in remote sensing images via weighted pose voting," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, pp. 1–17, 2017.
94. J. F. Vesecky, K. E. Laws and J. D. Paduan, "Using HF surface wave radar and the ship Automatic Identification System (AIS) to monitor coastal vessels," in *Proc. IGARSS*, pp. 761-764, 2009.
95. H. Su, S. Wei, S. Liu, J. Liang, C. Wang *et al.*, "HQ-ISNet: High-Quality Instance Segmentation for Remote Sensing Imagery," *Remote Sensing*, vol. 12, pp. 989, 2020.
96. D. J. Crisp, "The state-of-the-art in ship detection in synthetic aperture radar imagery," *Defence Science and Technology Organization Salisbury (Australia) Info Sciences Lab.*, Salisbury, Australia, 2004.
97. M. Kang, K. Ji, X. Leng and Z. Lin, "Contextual region-based convolutional neural network with multilayer fusion for SAR ship detection," *Remote Sensing*, vol. 9, pp. 860, 2017.
98. F. Charbonneau, B. Brisco and R. Raney, "Compact polarimetry overview and applications assessment," *Canadian Journal of Remote Sensing*, vol. 36, pp. 298-315, 2010.
99. B. Zhang, X. Li, W. Perrie and O. Garcia-Pineda, "Compact polarimetric synthetic aperture radar for marine oil platform and slick detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, pp. 1407-1423, 2017.
100. C. C. Wackerman, K. S. Friedman, W. G. Pichel, O. N. P. Clemente-Col and X. Li, "Automatic detection of ships in RADARSAT-1 SAR imagery," *Canadian Journal of Remote*

*Sensing*, vol. 27, no. 5, pp. 568-577, 2001.

101. Q. Fan, F. Chen, M. Cheng, S. Lou, R. Xiao *et al.*, “Ship detection using a fully convolutional network with compact polarimetric SAR images,” *Remote Sensing*, vol. 11, pp. 2171, 2019.
102. Y –L. Chang, A. Anagaw, L. Chang, Y. C. Wang, C – Y. Hsiao *et al.*, “Ship detection based on YOLOv2 for SAR imagery,” *Remote Sensing*, vol. 11, pp. 786, 2019.
103. W. Huo, Y. Huang, J. Pei, Q. Zhang, Q. Gu *et al.*, “Ship detection from ocean SAR image based on local contrast variance weighted information entropy,” *Sensors*, vol. 18, pp. 1196, 2018.
104. Y. Yao, Z. Jiang, H. Zhang, D. Zhao and B. Cai, “Ship detection in optical remote sensing images based on deep convolutional neural networks,” *Journal of Applied Remote Sensing*, vol. 11, no. 4, pp. 042611, 2017.
105. X. Yang, H. Sun, K. Fu, J. Yang, X. Sun *et al.*, “Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks,” *Remote Sensing*, vol. 10, no. 1, pp. 132, 2018.
106. T. Tran and T. Le, “Vision based boat detection for maritime surveillance,” in *Proc. ICEIC*, pp. 1-4, 2016.
107. J. L. Sanchez-Lopez, J. Pestana, S. Saripalli and P. Campoy, “An approach toward visual autonomous ship board landing of a VTOL UAV,” *Journal of Intelligent and Robotic Systems*, vol. 74, no. 1-2, pp. 113-127, 2014.
108. H. Zhao, W. Zhang, H. Sun and B. Xue, “Embedded deep learning for ship detection and recognition,” *Future Internet*, vol. 11, no. 2, pp. 53, 2019.
109. R. Wijnhoven, K. van Rens, E. G. Jaspers and P. H. de With, “Online learning for ship detection in maritime surveillance,” in *Proc. SITB2010*, pp. 73–80, 2010.
110. Y. Matsumoto, “Ship image recognition using HOG,” *The Journal of Japan Institute of Navigation*, pp. 129, 2013.
111. C. Dong, J. Liu, F. Xu and C. Liu, “Ship detection from optical remote sensing images using multi-scale analysis and Fourier HOG descriptor,” *Remote Sensing*, vol. 11, pp. 1529, 2019.
112. H. Lin, Z. Shi and Z. Zou, “Fully Convolutional Network with task partitioning for inshore ship detection in optical remote sensing images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1665-1669, 2017.
113. C. Zhu, H. Zhou, R. Wang and J. Guo, “A novel hierarchical method of ship detection from spaceborne optical image based on shape and texture features,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 9, pp. 3446-3456, 2010.

114. F. Yang, Q. Z. Xu, B. Li and Y. Ji, "Ship detection from thermal remote sensing imagery through region-based deep forest," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, pp. 449–453, 2018.
115. F. Wu, Z. Q. Zhou, B. Wang and J. L. Ma, "Inshore ship detection based on convolutional neural network in optical satellite images," *IEEE Journal of Selected Topics in Applied Earth-observation and Remote Sensing*, vol. 11, pp. 4005–4015, 2018.
116. R. F. Wang, J. Li, Y. P. Duan, H. J. Cao and Y. J. Zhao, "Study on the combined application of CFAR and deep learning in ship detection," *Journal of the Indian Society of Remote Sensing*, vol. 46, pp. 1413-1421, 2018.
117. G. Cheng, P. C. Zhou and J. W. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, pp. 7405–7415, 2016.
118. A. J. Gallego, A. Pertusa and P. Gil, "Automatic ship classification from optical aerial images with convolutional neural networks," *Remote Sensing*, vol. 10, pp. 511, 2018.
119. Q. P. Li, L. C. Mou, Q. J. Liu, Y. H. Wang and X. X. Zhu, "HSF-Net: Multiscale deep feature embedding for ship detection in optical remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, pp. 7147-7161, 2018.
120. W. C. Liu, L. Ma and H. Chen, "Arbitrary-oriented ship detection framework in optical remote-sensing images," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, pp. 937-941, 2018.
121. Z. Shao, L. Wang, Z. Wang, W. Du and W. Wu, "Saliency-aware convolution neural network for ship detection in surveillance video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, pp. 781-794, 2020.
122. S. Zhang, R. Wu, K. Xu, J. Wang and W. Sun, "R-CNN-based ship detection from high resolution remote sensing imagery," *Remote Sensing*, vol. 11, pp. 631, 2019.
123. X. Nie, M. Duan, H. Ding, B. Hu and E. K. Wong, "Attention mask R-CNN for ship detection and segmentation from remote sensing images," *IEEE Access*, vol. 8, pp. 9325-9334, 2020.
124. Y. Li, C. Peng, Y. Chen, L. Jiao, L. Zhou *et al.*, "A deep learning method for change detection in synthetic aperture radar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, pp. 5751-5763, 2019.
125. Z. Wang, T. Yang and H. Zhang, "Land contained sea area ship detection using spaceborne image," *Pattern Recognition Letters*, vol. 130, pp. 125-131, 2020.



126. X. Geng, L. Shi, J. Yang, P. Li, L. Zhao et al., "Ship detection and feature visualization analysis based on lightweight CNN in VH and VV polarization images," *Remote Sensing*, vol. 13, pp. 1184, 2021.
127. C. Zhijun, C. Depeng, Z. Yishi, C. Xiaozhao, Z. Mingyang et al., "Deep learning for autonomous ship-oriented small ship detection," *Safety Science*, vol. 130, pp. 104812, 2020.
128. Y. Zhang, L. Guo, Z. Wang, Y. Yu, X. Liu et al., "Intelligent ship detection in remote sensing images based on multi-layer convolutional feature fusion," *Remote Sensing*, vol. 12, pp. 3316, 2020.
129. M. M. Stofa, M. A. Zulkifley and S. Z. M. Zaki, "A deep learning approach to ship detection using satellite imagery," in *Proc. IGRSM*, Malaysia, 2020.
130. <https://www.kaggle.com/rhammell/ships-in-satellite-imagery>.
131. K. Roy, S. S. Chaudhuri and S. Pramanik, "Deep learning based real-time Industrial framework for rotten and fresh fruit detection using semantic segmentation," *Microsystem Technologies*, vol. 27, pp. 3365-3375, 2021.
132. Y. Wu, X. Feng and G. Chen, "Plant Leaf Diseases Fine-Grained Categorization Using Convolutional Neural Networks," *IEEE Access*, vol. 10, pp. 41087-41096, 2022.
133. M. Adnan, K. Ali, G. Drushti and C. Tejal, "Plant disease detection using CNN & remedy," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 8, issue 3, 2019
134. A. Abade, P. A. Ferreira and F. D. B Vidal, "Plant diseases recognition on images using convolutional neural networks: A systematic review," *Computers and Electronics in Agriculture*, vol.185, pp.106125, 2021.
135. M. H. Saleem, S. Khanchi, J. Potgieter and K. M. Arif, "Image-Based Plant Disease Identification by Deep Learning Meta-Architectures," *Plants*, vol. 9, 2020.
136. N. Ullah, J. A. Khan, L. A. Alharbi, A. Raza, W. Khan and I. Ahmad, "An Efficient Approach for Crops Pests Recognition and Classification Based on Novel DeepPestNet Deep Learning Model," *IEEE Access*, vol. 10, pp. 73019-73032, 2022.
137. S. Ahmed, M. B. Hasan, T. Ahmed, M. R. K. Sony and M. H. Kabir, "Less is More: Lighter and Faster Deep Neural Architecture for Tomato Leaf Disease Classification," *IEEE Access*, vol. 10, pp. 68868-68884, 2022.
138. E. Elfatimi, R. Eryigit and L. Elfatimi, "Beans Leaf Diseases Classification Using MobileNet Models," *IEEE Access*, vol. 10, pp. 9471-9482, 2022.
139. L. Aversano, M. L. Bernardi, M. Cimitile, M. Iammarino and S. Rondinella, "Tomato diseases Classification Based on VGG and Transfer Learning," *IEEE International Workshop on*

*Metrology for Agriculture and Forestry (MetroAgriFor 2020)*, pp. 129-133, Trento, Italy, November 2020.

140. S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational Intelligence and Neuroscience*, 2016.
141. E. Özbilge, M. K. Ulukök, Ö. Toygar and E. Ozbilge, "Tomato Disease Recognition Using a Compact Convolutional Neural Network," *IEEE Access*, vol. 10, pp. 77213-77224, 2022.
142. H. Ajra, M. K. Nahar, L. Sarkar and M.S. Islam, "Disease Detection of Plant Leaf using Image Processing and CNN with Preventive Measures," *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE-2020)*, pp.1-6, Bangladesh, December 2020.
143. M. Chohan, A. Khan, R. Chohan, S. Hassan and M. Mahar, "Plant disease detection using deep learning," *International Journal of Recent Technology and Engineering*, vol.9, no. 1, pp.909-914,2020.
144. A. Rao and S. B. Kulkarni, "A hybrid approach for plant leaf disease detection and classification using digital image processing methods," *International Journal of Electrical Engineering & Education*, October 2020.
145. M. K. Singh, S. Chetia, and M. Singh, "Detection and classification of plant leaf diseases in image processing using MATLAB," *International Journal of Life Sciences Research*, vol.5, no. 4, pp.120-124, 2017.
146. A. Patel, and B. Joshi, "A survey on the plant leaf disease detection techniques", *International Journal of Advanced Research in Computer and Communication Engineering*, vol.6, no. 1, pp. 229-231, 2017.
147. J. Gui, L. Hao, Q. Zhang and X.Bao, "A new method for soybean leaf disease detection based on modified salient regions," *International Journal of Multimedia and Ubiquitous Engineering*, vol.10, no. 6, pp.45-52, 2015.
148. S. Pavithra, A. Priyadharshini, V. Praveena and T. Monika, "Paddy Leaf Disease Detection Using SVM Classifier," *International Journal of Communication and Computer Technologies*, vol.3, no. 1, pp.16-20, 2015.
149. Y. M. Oo and N. C. Htun, "Plant leaf disease detection and classification using image processing," *International Journal of Research and Engineering*, vol.5, no. 9, pp.516-523, 2018.
150. D. Ashourloo, H. Aghighi, A. A. Matkan, M. R. Mobasheri and A. M. Rad, "An investigation into machine learning regression techniques for the leaf rust disease detection using hyperspectral measurement," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 9, pp.4344-4351, 2016.

151. H. Nazki, S. Yoon, A. Fuentes, and D. S. Park, "Unsupervised image translation using adversarial networks for improved plant disease recognition," *Computers and Electronics in Agriculture*, vol.168, pp.105117, 2020.
152. Q. Zeng, X. Ma, B. Cheng, E. Zhou, and W. Pang, "GANs-based data augmentation for citrus disease severity detection using deep learning," *IEEE Access*, vol. 8, pp.172882-172891, 2020.
153. Y. Zhao, Z. Chen, X. Gao, W. Song, Q. Xiong, J. Hu, and Z. Zhang, "Plant Disease Detection using Generated Leaves Based on DoubleGAN," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021.
154. L. Li, S. Zhang, and B. Wang, "Plant disease detection and classification by deep learning - A review," *IEEE Access*, vol. 9, pp.56683-56698, 2021.
155. M. S. Mahmoudi, K. Boukhalfa, and A. Moussaoui, "Deep interpretable architecture for plant diseases classification," In *2019 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, IEEE, pp. 111-116,2019.
156. J. G. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," *Biosystems Engineering*, vol.172, pp.84-91, 2018.
157. PlantVillage Dataset (<https://www.kaggle.com/abdallahalidev/plantvillage-dataset>).
158. F. Harrou, M. N. Nounou, H. N. Nounou and M. Madakyaru, "Statistical fault detection using PCA-based GLR hypothesis testing", *Journal of loss prevention in the process industries*, vol.26, no. 1, pp.129-139, 2013.
159. A. M. Dawud, K. Yurtkan, and H. Oztoprak, "Application of deep learning in neuroradiology: brain haemorrhage classification using transfer learning," *Computational Intelligence and Neuroscience*, 2019.
160. F. Mohameth, C. Bingcai and K. A. Sada, "Plant disease detection with deep learning and feature extraction using plant village", *Journal of Computer and Communications*, vol.8, no. 6, pp.10-22, 2020.
161. A. Rehman, S. Naz, M. I. Razzak, F. Akram and M. Imran, "A deep learning-based framework for automatic brain tumors classification using transfer learning," *Circuits, Systems, and Signal Processing*, vol.39, no. 2, pp.757-775, 2020.
162. Y. H. Bhosale and K. Sridhar Patnaik, "IoT Deployable Lightweight Deep Learning Application for COVID-19 Detection with Lung Diseases Using Raspberry Pi," *2022 International Conference on IoT and Blockchain Technology (ICI BT)*, Ranchi, India, pp. 1-6, 2022.
163. A. Tripathy, A. Agrawal and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Systems with Applications*, vol.57, pp.117-126, 2016.

164. V. V. Srinidhi, A. Sahay and K. Deeba, "Plant pathology disease detection in apple leaves using deep convolutional neural networks: Apple leaves disease detection using efficientnet and densenet", In *2021 5<sup>th</sup> International Conference on Computing Methodologies and Communication (ICCMC-2021)*, pp. 1119-1127, IEEE, 2021.
165. E. Harte, "Plant disease detection using CNN," *ResearchGate*, 2020.
166. Y. Toda and F. Okura, "How Convolutional Neural Networks Diagnose Plant Disease", *Plant Phenomics*, 2019.
167. P. Jiang, Y. Chen, B. Liu, D. He and C. Liang, "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks," *IEEE Access*, vol.7, pp.59069-59080, 2019.
168. M. Souaidi, and M. El Ansari, "A New Automated Polyp Detection Network MP-FSSD in WCE and Colonoscopy Images Based Fusion Single Shot Multibox Detector and Transfer Learning," *IEEE Access*, vol. 10, 2022, pp. 47124-47140.
169. J. Meng, Z. Luo, Z. Chen, J. Zhou, Z. Chen, B. Lu, M. Zhang, Y. Wang, C. Yuan, X. Shen, Q. Huang, Z. Zhang, Z. Ye, Q. Cao, Z. Zhou, Y. Xu, R. Mao, M. Chen, C. Sun, Z. Li, ST Feng, X. Meng, B. Huang and X. Li, "Intestinal fibrosis classification in patients with Crohn's disease using CT enterography-based deep learning: comparisons with radiomics and radiologists," *European Radiology*. 2022, pp. 1-14.
170. E. Redondo-Cerezo, A. D. Sánchez-Capilla, P. De La Torre-Rubio and J. De Teresa, "Wireless capsule endoscopy: perspectives beyond gastrointestinal bleeding," *World Journal of Gastroenterology: WJG*, vol. 20 (42), pp. 15664, 2014.
171. G. Pan and L. Wang, "Swallowable wireless capsule endoscopy: Progress and technical challenges," *Gastroenterology research and practice*, 2012.
172. S. Tanwar, S. Vijayalakshmi, M. Sabharwal, M. Kaur A. A AlZubi and H. N. Lee, "Detection and Classification of Colorectal Polyp Using Deep Learning," *BioMed Research International*, 2022.
173. Y. Masmoudi, M. Ramzan, S. A Khan and M. Habib, "Optimal feature extraction and ulcer classification from WCE image data using deep learning," *Soft Computing*, pp. 1-14, 2022.
174. P. Muruganatham and S. M. Balakrishnan, "Attention aware deep learning model for wireless capsule endoscopy lesion classification and localization," *Journal of Medical and Biological Engineering*, vol. 42(2), pp. 157- 168, 2022.
175. D. Marin-Santos, J. A. Contreras-Fernandez, I. Perez-Borrero, H. Pallares-Manrique and M. E. Gegundez-Arias, "Automatic detection of crohn disease in wireless capsule endoscopic images using a deep convolutional neural network," *Applied Intelligence*, pp.1-15, 2022.

176. E. Klang, A. Grinman, S. Soffer, R. Margalit Yehuda, O. Barzilay, M. M. Amitai, E. Konen, S. Ben-Horin, R. Eliakim and U. Kopylov “Automated detection of Crohn’s disease intestinal strictures on capsule endoscopy images using deep neural networks,” *Journal of Crohn’s and Colitis*, vol. 15(5), pp.749-756, 2022.
177. A. de Maissin, R. Vallée, M. Flamant, M. Fondain-Bossiere, C. Le Berre, A. Coutrot, N. Normand, H. Mouchère, S. Coudol, C. Trang and A. Bourreille, “Multi-expert annotation of Crohn’s disease images of the small bowel for automatic detection using a convolutional recurrent attention neural network,” *Endoscopy International Open*, vol. 9(07), pp. E1136-E1144, 2021.
178. Z. Falin, L. Haihua, and P. Ning, “Gastrointestinal Polyps and Tumors Detection Based on Multi-scale Feature-fusion with WCE Sequences,” 2022, arXiv preprint arXiv: 2204.01012.
179. A. Ellahyani, I. E. Jaafari, S. Charfi and M. E. Ansari, “Fine-tuned deep neural networks for polyp detection in colonoscopy images,” *Personal and Ubiquitous Computing*, pp. 1-13, 2022.
180. B. S. Lewis, “Small intestinal bleeding,” *Gastroenterology Clinics of North America*, vol. 29(1), pp.67-95, 2000.
181. G. R. Zuckerman, C. Prakash, M. P. Askin and B. S. Lewis, “AGA technical review on the evaluation and management of occult and obscure gastrointestinal bleeding,” *Gastroenterology*, vol. 118(1), pp. 201-221, 2000.
182. M. K. Goenka, S. Majumder and U. Goenka, “Capsule endoscopy: Present status and future expectation,” *Journal of Gastroenterology: WJG*, vol. 20(29), pp. 10024, 2014.
183. U. C. Ghoshal and S. Amornyotin, “Capsule endoscopy: A new era of gastrointestinal endoscopy,” *InTech.*, pp. 75-88, 2013.
184. A. Moglia, A. Menciasci, P. Dario and A. Cuschieri, “Capsule endoscopy: progress update and challenges ahead,” *Nature Reviews Gastroenterology Hepatology*, vol. 6(6), pp. 353, 2009.
185. B. Li and M. Q. H. Meng, “Computer-aided detection of bleeding regions for capsule endoscopy images,” *IEEE Transactions on Biomedical Engineering*, vol. 56(4), pp.1032-1039, 2009.
186. A. K. Kundu, S. A. Fattah and M. N. Rizve, “An automatic bleeding frame and region detection scheme for wireless capsule endoscopy videos based on interplane intensity variation profile in normalized RGB color space,” *Journal of healthcare Engineering*, 2018.
187. K. Pogorelov, S. Suman, F. AzmadiHussin, A. Saeed Malik, O. Ostroukhova, M. Riegler, P. Halvorsen, S. Hooi Ho and K. L. Goh, “Bleeding detection in wireless capsule endoscopy videos – Color versus texture features,” *Journal of applied clinical medical physics*, vol. 20(8), pp.141-154, 2021.

188. J. M. Herrerias, A. Caunedo, M. Rodriguez-Tellez, F. Pellicer and J. M. Herrerias Jr, "Capsule endoscopy in patients with suspected Crohn's disease and negative endoscopy," *Endoscopy*, vol. 35(07), pp. 564-568, 2003.
189. W. A. Voderholzer, J. Beinhoelzl, P. Rogalla, S. Murrer, G. Schachschal, H. Lochs and M. A. Ortner, "Small bowel involvement in Crohn's disease: a prospective comparison of wireless capsule endoscopy and computed tomography enteroclysis," *Gut.*, vol. 54(3), pp.369-373, 2005.
190. M. Tukey, D. Pleskow, P. Legnani, A. S. Cheifetz and A. C. Moss, "The utility of capsule endoscopy in patients with suspected Crohn's disease," *American Journal of Gastroenterology*, vol. 104(11), pp.2734-2739, 2009.
191. T. Aoki, A. Yamada, K. Aoyama, H. Saito, A. Tsuboi, A. Nakada, R. Niikura, M. Fujishiro, S. Oka, S. Ishihara, T. Matsuda, S. Tanaka, K. Koike and T. Tada, "Automatic detection of erosions and ulcerations in wireless capsule endoscopy images based on a deep convolutional neural network," *Gastrointestinal endoscopy*, vol. 89(2), pp.357-363, 2019.
192. S. Fan, L. Xu, Y. Fan, K. Wei and L. Li, "Computer-aided detection of small intestinal ulcer and erosion in wireless capsule endoscopy images," *Physics in Medicine Biology*, vol. 63(16), pp.165001, 2018.
193. H. Alaskar, A. Hussain, N. Al-Aseem, P. Liatsis, and D. Al-Jumeily, "Application of convolutional neural networks for automated ulcer detection in wireless capsule endoscopy images," *Sensors*, vol.19(6), pp.1265, 2019.
194. B. Li and M. Q. H. Meng, "Texture analysis for ulcer detection in capsule endoscopy images," *Image and Vision computing*, vol. 27(9), pp.1336- 1342, 2009.
195. L. Fuccio, A. Mussetto, L. Laterza, L. H. Eusebi and F. Bazzoli, "Diagnosis and management of gastric antral vascular ectasia," *World journal of gastrointestinal endoscopy*, vol. 5(1), pp.6, 2013.
196. I. Yusoff, F. Brennan, D. Ormonde and B. Laurence, "Argon plasma coagulation for treatment of watermelon stomach," *Endoscopy*, vol. 34(05), pp.407-410, 2002.
197. M. Abbari, R. Cherry, J. O. Lough, D. S. Daly, D. G. Kinnear and C. A. Goresky, "Argon plasma coagulation for treatment of watermelon stomach," *Gastroentology*, vol. 87(5), pp. 1165-1170, 1984.
198. S. Q. Yang, R. Huang, L. H. Zhang, J. G. Hu and L. Yang, "Tapeworm infection identified on capsule endoscopy," *Journal of interventional gastroenterology*, vol. 2(1), pp.19, 2012.
199. N. Hosoe, H. Imaeda, S. Okamoto, R. Bessho, R. Saito, Y. Ida, S. Kobayashi, T. Kanai, T. Hibi and H. Ogata, "A case of beef tapeworm (*Taeniasaginata*) infection observed by using video

- capsule endoscopy and radiography (with videos),” *Gastrointestinal endoscopy*, vol. 74(3), pp. 690-691, 2011.
200. K. Barnett, P. Emdar, A. S. Day and W. S. Selby, “Tapeworm infestation: a cause of iron deficiency anemia shown by capsule endoscopy,” *Gastrointestinal endoscopy*, vol. 66(3), pp. 625-627, 2007.
201. D. Mukhtorov, M. Rakhmonova, S. Muksimova and Y. I. Cho, “Endoscopic Image Classification Based on Explainable Deep Learning,” *Sensors*, vol. 23(6), pp. 3176, 2023.
202. I. Iqbal, K. Walayat, M. U. Kakar and J. Ma “Automated identification of human gastrointestinal tract abnormalities based on deep convolutional neural network with endoscopic images,” *Intelligent Systems with Applications*, vol. 16, pp. 200149, 2022.
203. M. Obayya, F. N Al-Wesabi, M. Maashi, A. Mohamed, M. A. Hamza, S. Drar, I. Yaseen, and M. I. Alsaïd “Modified Salp Swarm Algorithm with Deep Learning Based Gastrointestinal Tract Disease Classification on Endoscopic Images,” *IEEE Access*, vol. 11, pp. 25959-25967, 2023.
204. Z. Ghaleb Al-Mekhlafi, E. Mohammed Senan, J. Sulaiman Alshudukhi and B. Abdulkarem Mohammed, “Hybrid Techniques for Diagnosing Endoscopy Images for Early Detection of Gastrointestinal Disease Based on Fusion Features,” *International Journal of Intelligent Systems*, 2023.
205. V. Sharmila and S. Geetha, “Detection and Classification of GI-Tract Anomalies from Endoscopic Images Using Deep Learning,” In 2022 *IEEE 19<sup>th</sup> India Council International Conference (INDICON)*, pp.1-6, Nov. 2023.
206. A. Ahmed, “Classification of gastrointestinal images based on transfer learning and denoising convolutional neural networks,” In Proc. of *International Conference on Data Science and Applications: ICDSA 2021*, Springer Singapore, pp. 631-639, 2022.
207. S. Bandopadhyay, I. J Ray, S. Mondal, S. Manna, S. Mitra, K. Roy, S. Banerjee and S. S. Chaudhuri, “Drowsy Driving Detection Based on Deep Neural Network for Accident Avoidance,” *Proc. of International Conference on Computational Intelligence, Data Science and Cloud Computing. Algorithms for Intelligent Systems*. Springer, Singapore.
208. K. Roy, S. S. Chaudhuri, J. Frnda, S. Bandopadhyay, I. J Ray, S. Banerjee and J. Nedoma, “Detection of Tomato Leaf Diseases for Agro-Based Industries Using Novel PCA DeepNet,” *IEEE Access*, vol. 11, pp.14983-15001, 2023, doi: 10.1109/ACCESS.2023.3244499.
209. S. Mohapatra, G. K. Patil, M. Mishra and T. Swarnkar, “Gastrointestinal abnormality detection and classification using empirical wavelet transform and deep convolutional neural network from endoscopic images,” *Ain Shams Engineering Journal*, vol. 14(4), pp.101942, 2023.

210. G. Satyanarayana, P.A. Naidu, V. S. Desanamukula and B. C. Rao, "A mass correlation based deep learning approach using deep Convolutional neural network to classify the brain tumor," *Biomedical Signal Processing and Control*, vol. 81, pp. 104395, 2023.
211. Q. Pan, Y. Bao and H. Li, "Transfer learning-based data anomaly detection for structural health monitoring," *Structural Health Monitoring*, vol. 66(3), pp.14759217221142174, 2023.
212. S. Matta, M. Lamard, P. H. Conze, A. Le Guilcher, V. Ricquebourg, A. A. Benoyoussef, P. Massin, J. B. Rottier, B. Cochener and G. Quellec, "Meta learning for anomaly detection in fundus photographs," *In Meta-Learning with Medical Imaging and Health Informatics Applications*, Academic Press, pp. 301-329, 2023.
213. W. Fan, W. Shangguan and N. Bouguila, "Continuous image anomaly detection based on contrastive lifelong learning," *Applied Intelligence*, pp.1-5, 2023.
214. I. Ahmed, M. Ahmad, A. Chehri and G. Jeon, "A Smart-Anomaly- Detection System for Industrial Machines Based on Feature Autoencoder and Deep Learning," *Micromachines*, vol. 14(1), pp. 154, 2023.
215. Z. Wang, Z. Wang, C. Zeng, Y. Yu, and X. Wan, "High-quality image compressed sensing and reconstruction with multi-scale dilated convolutional neural network," *Circuits, Systems, and Signal Processing*, vol. 42(3), pp.1593-1616, 2023.
216. C. Yin, S. Zhang, J. Wang and N. N. Xiong, "Anomaly Detection Based on Convolutional Recurrent Autoencoder for IoT Time Series," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 1, pp. 112-122, Jan. 2022.
217. M. N. Onal, G. E. Guraksin and R. Duman, "Convolutional neural network- based diabetes diagnostic system via iridology technique," *Multimedia Tools and Applications*, vol. 81(1), pp.173-194, 2023.
218. N. Lopac, F. Hržic, I. P. Vuksanovic´ and J. Lerga, "Detection of Non-Stationary GW Signals in High Noise from Cohen's Class of Time-Frequency Representations Using Deep Learning," *IEEE Access*, vol. 10, pp. 2408-2428, 2022, doi: 10.1109/ACCESS.2021.3139850.
219. Y. Zhang, J. Yi, A. Chen and L. Cheng, "Cardiac arrhythmia classification by time-frequency features inputted to the designed convolutional neural networks," *Biomedical Signal Processing and Control*, vol.79, pp.104224, 2023.
220. Y. Jin, Z. Li, C. Qin, J. Liu, Y. Liu, L. Zhao and C. Liu, "A novel attentional deep neural network-based assessment method for ECG quality," *Biomedical Signal Processing and Control*, vol. 79, pp.104064, 2023.



221. T. Rahim, M. A. Usman and S. Y. Shin, "A survey on contemporary computer-aided tumor, polyp, and ulcer detection methods in wireless capsule endoscopy imaging," *Computerized Medical Imaging and Graphics*, vol.101767, 2020.
222. P. Szczypin'ski, A. Klepaczko, M. Pazurek and P. Daniel, "Texture and color based image segmentation and pathology detection in capsule endoscopy videos," *Computer methods and programs in biomedicine*, vol. 113(1), pp.396-411, 2014.
223. E. Tuba, M. Tuba and R. Jovanovic, "An algorithm for automated segmentation for bleeding detection in endoscopic images," In *2017 International Joint Conference on Neural Networks (IJCNN)*, IEEE., pp. 4579-4586, May. 2017.
224. K. Pogorelov, M. Riegler, S. L. Eskeland, T.de Lange, D. Johansen, C. Griwodz, P. T. Schmidt and P. Halvorsen, "Efficient disease detection in gastrointestinal videos—global features versus neural networks," *Multimedia Tools and Applications*, vol. 76(21), pp. 22493-22525, 2017.
225. G. Bao, L. Mi, Y. Geng, M. Zhou and K. Pahlavan, "A video-based speed estimation technique for localizing the wireless capsule endoscope inside gastrointestinal tract," In *36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, pp. 5615-5618, Aug. 2014.
226. A. Novozámský, J. Flusser, I. Tachecí, L.Sulík, J. Bureš, and O. Krejcar, "An algorithm for automated segmentation for bleeding detection in endoscopic images," *Journal of biomedical optics*, vol. 21(12), pp.126007, 2016.
227. E. David, R. Boia, A. Malaescu and M. Carnu, "Automatic colon polyp detection in endoscopic capsule images," In *International Symposium on Signals, Circuits and Systems ISSCS2013*, IEEE, July 2013.
228. D. Chen, M. Q. H. Meng, H. Wang, C. Hu and Z. Liu, "A novel strategy to label abnormalities for wireless capsule endoscopy frames sequence," In *2011 IEEE International Conference on Information and Automation*, pp. 379-383, IEEE, Jun. 2011.
229. V. S. Prasath, I. N. Figueiredo and P. N. Figueiredo, "Colonic mucosa detection in wireless capsule endoscopic images and videos," In *Congress on Numerical Methods in Engineering (CMNE 2011)*, Coimbra, Portugal, Jun. 2011.
230. S. Bejakovic, R. Kumar, T. Dassopoulos, G. Mullin and G. Hager, "Analysis of Crohn's disease lesions in capsule endoscopy images," In *2009 IEEE International Conference on Robotics and Automation*, pp. 2793-2798, IEEE, May, 2009.
231. P. C. Khun, Z. Zhuo, L. Z. Yang, L. Liyuan and L. Jiang, "Feature selection and classification for wireless capsule endoscopic frames," In *2009 International Conference on Biomedical and Pharmaceutical Engineering*, pp. 1-6, IEEE, Dec. 2009.

232. E. Gal, A. Geller, G. Fraser, Z. Levi and Y. Niv, "Assessment and validation of the new capsule endoscopy Crohn's disease activity index (CECDAI)," *Digestive diseases and sciences*, vol. 53(7), pp.1933-1937, 2008.
233. K. S. Chuang, H. L. Tzeng, S. Chen, J. Wu and T. J. Chen, "Fuzzy c-means clustering with spatial information for image segmentation", *Computerized Medical Imaging and Graphics*, 30(1), 9-15, 2006.
234. S. K. Choy, K. Yuen and C. Yu, "Fuzzy bit-plane-dependence image segmentation," *Signal Processing*, vol. 154, pp.30-44, 2019.
235. S. Chakraborty, A. Raman, S. Sen, K. Mali, S. Chatterjee and H. Hachimi, "Contrast optimization using elitist metaheuristic optimization and gradient approximation for biomedical image enhancement," In 2019 *Amity International Conference on Artificial Intelligence (AICAI)*, pp. 712-717, IEEE, Feb. 2019.
236. S. Chakraborty, K. Mali, S. Chatterjee, S. Banerjee, A. Sah, S. Pathak, S. Nath and D. Roy, "Bio-medical image enhancement using hybrid metaheuristic coupled soft computing tools," In 2017 *IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 231-236, IEEE, Oct.2017.
237. J. Afonso, M. Mascarenhas, T. Ribeiro, H. Cardoso, P. Andrade, J. Ferreira, M. Parente, R. N. Jorge and G. Macedo, "S558 Artificial Intelligence and Capsule Endoscopy: Automatic Detection of Gastric Vascular Lesions Using a Convolutional Neural Network," *Official journal of the American College of Gastroenterology| ACG*, vol. 116, pp. S255, 2021.
238. S. Mandal, M. Adhikari, S. Banerjee and S. S. Chaudhuri, "Novel Adaptive Statistical Method-CNN Synergism Based Two-Step WCE Image Segmentation," In 2019 *International Conference on Intelligent Computing and Control Systems (ICCS)*, IEE, pp. 1024-1029, May, 2019.
239. <https://www.kaggle.com/datasets/meetnagadia/kvasir-dataset>

Kyranelis Ray  
9/9/2023