Automatic Fish Species Recognition from the Cluttered Environment

Thesis Submitted by

Arnab Banerjee

Doctor of Philosophy (Engineering)

Department of Computer Science And Engineering
Faculty Council of Engineering and Technology
Jadavpur Universiry
Kolkata, India

2024

JADAVPUR UNIVERSITY
Kolkata, India

INDEX NO. **(112/19/E)**

1. **Title of the Thesis**: Automatic Fish Species Recognition from the Cluttered Environment

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

   (a) **Prof. (Dr.) Nibaran Das**
   Professor, Department of Computer Science and Engineering
   Jadavpur University, Kolkata, India

3. **List of Publications:**

   (a) **Journals:**

   i. Arnab Banerjee, Arijit Das, Samarendra Behra, Debotosh Bhattacharjee, Nagesh Talagunda Srinivasan, Mita Nasipuri, Nibaran Das,. Carp-DCAE: Deep convolutional autoencoder for carp fish classification, Computers and Electronics in Agriculture, Volume 196,2022,106810,ISSN 0168-1699, https://doi.org/10.1016/j.compag.2022.106810.

   (b) **International Conferences:**

   i. Banerjee, A., Chakraborty, R., Behra, S., Srinivasan, N.T., Bhattacharjee, D., Das, N. (2022). Deep Learning Based Identification of Three Exotic Carps. In: Das, A.K., Nayak, J., Naik, B., Vimal, S., Pelusi, D. (eds) Computational Intelligence in Pattern Recognition. CIPR 2022. Lecture Notes in Networks and Systems, vol 480. Springer, Singapore. https://doi.org/10.1007/978-981-19-3089-8_40.

   ii. A. Banerjee, D. Bhattacharjee, N. Das, S. Behra and N. T. Srinivasan, ”CARP-YOLO: A Detection Framework for Recognising and Counting Fish Species in a Cluttered Environment,” 2023 4th International Conference for Emerging Technology (INCET), Belgaum, India, 2023, pp. 1-7, doi: 10.1109/INCET57972.2023.10170475.

   iii. A. Banerjee, D. Bhattacharjee, N. T. Srinivasan, S. Behra and N. Das, ”SegFishHead: A Semantic Segmentation Approach for the identification of fish species in a Cluttered Environment,” 2023 International Conference on Computer, Electronics & Electrical Engineering & their Applications (IC2E3), Srinagar Garhwal, India, 2023, pp. 1-6, doi: 10.1109/IC2E357697.2023.10262432.

4. **List of Patents:** None

5. **List of Presentations in National/International Conferences and Workshops:**

   (a) Banerjee, A., Chakraborty, R., Behra, S., Srinivasan, N.T., Bhattacharjee, D., Das, N. (2022). Deep Learning Based Identification of Three Exotic Carps. In: Das, A.K., Nayak, J., Naik, B., Vimal, S., Pelusi, D. (eds) Computational Intelligence in Pattern Recognition. CIPR 2022. Lecture Notes in Networks and Systems, vol 480. Springer, Singapore. https://doi.org/10.1007/978-981-19-3089-8_40.

   (b) A. Banerjee, D. Bhattacharjee, N. Das, S. Behra and N. T. Srinivasan, "CARP-YOLO: A Detection Framework for Recognising and Counting Fish Species in a Cluttered Environment," 2023 4th International Conference for Emerging Technology (INCET), Belgaum, India, 2023, pp. 1-7, doi: 10.1109/INCET57972.2023.10170475.

   (c) A. Banerjee, D. Bhattacharjee, N. T. Srinivasan, S. Behra and N. Das, "SegFishHead: A Semantic Segmentation Approach for the identification of fish species in a Cluttered Environment," 2023 International Conference on Computer, Electronics & Electrical Engineering & their Applications (IC2E3), Srinagar Garhwal, India, 2023, pp. 1-6, doi: 10.1109/IC2E357697.2023.10262432.

JADAVPUR UNIVERSITY
Kolkata, India

STATEMENT OF ORIGINALITY

I, **Shri Arnab Banerjee** registered on **10th June, 2019**, do hereby declare that this thesis entitled **"Automatic Fish Species Recognition from the Cluttered Environment"** 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 8%.

Signature of the Candidate : Arnab Banerjee

Date : 30.07.2024

Certified by Supervisors :
(Signature with date, seal)

Nibaran Das 30/7/24
....
(Prof. (Dr.) Nibaran Das)

Professor
Computer Sc. & Engg. Department
Jadavpur University
Kolkata-700032

JADAVPUR UNIVERSITY
Kolkata, India

CERTIFICATE FROM THE SUPERVISOR/S

This is to certify that the thesis entitled "**Automatic Fish Species Recognition from the Cluttered Environment"** submitted by **Shri Arnab Banerjee**, who got his name registered on 10th **June, 2019** for the award of **Ph.D. (Engg.)** degree of Jadavpur University is absolutely based upon his own work under the supervision of **Prof. (Dr.) Nibaran Das** and that neither his thesis nor any part of the thesis has been submitted for any degree/diploma or any other academic award anywhere before.

Nibaran Das 30/7/24

(Prof. (Dr.) Nibaran Das)
Signature of the Supervisor and
date with Official seal

Professor
Computer Sc. & Engg. Department
Jadavpur University
Kolkata-700032

Acknowledgements

My PhD journey, which started in 10th June, 2019, has been through the many highs and lows. Many a time a lot of uncertainties and challenges came, and I tackled all of those with the help of many intellectual and kind persons. Nonetheless, I finished it in line! I want to express my deepest gratitude to all of those individuals who continuously guided me in this challenging journey and empowering me to obtain my PhD by removing all the obstacles that came along the way.

First of all, I want to thank my supervisor, Prof. (Dr.) Nibaran Das. His constant guidance and boosting have been outstanding throughout the journey. From the initial days of refinement of the research proposal to the final submission of my thesis, his insight and valuable suggestions from time to time greatly helped me to design the long path of obtaining this degree. I deeply valued the meetings we did, which helped me to keep on track as well gave me lots of encouragement.

I would like to thank the professors involved in my thesis, Prof. (Dr.) Debotosh Bhattacharjee, Prof. (Dr.) Mita Nasipuri, Prof. (Dr.) Samarendra Behera, and Prof. (Dr.) Nagesh Talagunda Srinivasan. Their valuable insights, constructive criticism, and scholarly input throughout the journey have been incredible.

I would like to thank HOD of CSE department, Jadavpur university, Prof. (Dr.) Nandini Mukherjee, for overall support throughout my PhD journey.

I would like to thank CMATER lab, Department of CSE, Jadavpur University, for the overall infrastructure and facilities.

I would like to thank my organization, Dr. B. C. Roy Polytechnic, and Dr. B. C. Roy Engineering College, Durgapur, India, for allowing me to devote my time to completing the PhD.

Thanks to all my co-authors, for being a part of the interesting works completed in this journey. Thanks to Mr. Arijit Das, Ms. Roopsia Chakraborty for your support.

Lastly, I would like to express my heartfelt gratitude to my father, Mr. Bijoy Bandyopadhyay, my mother, Mrs. Sujata Banerjee, my wife, Mrs. Bratati Ganguly, and my son, Mr. Artaniv Banerjee, for their unwavering support and encouragement throughout this journey. I am also deeply thankful to my elder brother, Mr. Gourab Banerjee, his wife, Mrs. Debasmita Banerjee, and their son, Mr. Aariketh Banerjee, for their love, guidance, and constant motivation.

Arnab Banerjee
30.07.24

. . . . . .

Dedicated to
my
beloved father Bijoy Banerjee,
beloved mother Sujata Banerjee,
beloved wife Bratati Ganguly,
and
beloved son Artaniv Banerjee

. . . . . .

“If you cannot do great things,
do small things in a great way”

--- Napoleon Hill

“Start by doing what’s necessary;
then do what’s possible;
and suddenly you are doing the impossible”

--- Francis of Assisi

* * * * * * * * *

Life Steps

Our entire life is made up of choices,
What we decide, the action we take,
the attitude we display
All represent the steps of life.

Sometimes we take two steps forward
And one-step back.
Some of us take baby steps
Some of us take giant steps

But the secret is not to let that
one step back turn into a failure.
Learn from backward steps
And keep on stepping forward in this dance Called Life!

---Catherine Pulsifer

* * * * * * * * *

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

India's fish production has seen significant growth in recent years, positioning the country as a major player in the global fisheries sector. The production growth has been supported by various government initiatives, most notably the Pradhan Mantri Matsya Sampada Yojana (PMMSY). Launched in 2020, PMMSY aims to enhance fish production, improve infrastructure, and boost fishers' income through diverse activities such as constructing ponds, developing recirculatory aquaculture systems, and establishing cold storage facilities. Additionally, the government has been developing major fishing harbours and providing financial support to states and union territories to bolster the sector further. Integrating Industry 4.0 technologies into the fisheries, often referred to as "Aquaculture 4.0," is transforming how fisheries operate, enhancing efficiency, sustainability, and productivity. There are various areas where Industry 4.0 techniques could be applied to enhance the fishery sector in India. Artificial Intelligence and Machine Learning analyse vast amounts of data to predict and optimize various aspects of fish farming, from fish identification, and feed optimization to disease detection. Internet of Things (IoT) devices and sensors are pivotal in the real-time monitoring of water quality parameters such as temperature, dissolved oxygen, pH, and salinity. Automation technologies are required in this sector to reduce the labour costs in routine tasks and increase the precision in farming. Recognition of fish species is the fundamental routine task that needs to be automated to apply the autonomous system in feeding, fish farming, cleaning to fish disease detection.

As lifestyles change and urbanization increases, human dietary patterns are evolving, with a shift towards more protein-rich foods. Fish is an easily digestible aquatic food, with low fat and high protein, a rich source of micronutrients which plays an effective role in human nutrition. Fish consumption in India has been increasing steadily over the years due to various factors including population growth, rising incomes, urbanization, and awareness of the health benefits of fish consumption. In coastal regions and areas with significant inland fisheries, fish has become an increasingly important part of the Indian diet. India has a wide coastline of over 8,162 kilometers, bringing forth manifold marine fish species. The country also has huge inland water resources, including rivers, lakes, reservoirs, ponds, and brackishwaters, which contribute to producing freshwater fish. India is the world's third largest fish producer accounting for 8.92 % of global production. An estimated 16.25 million metric tons (MMT) of fish were produced in India overall in FY 2022–2023, with 4.13 MMT coming from the marine sector and 12.12 MMT from the inland sector. Aquaculture plays a significant role in fish production in India. Major aquaculture species include freshwater fish like carp species (*Labeo rohita*, *Labeo catla*, *Cirrhinus mrigala*), exotic carps (*Hypophthalmichthys molitrix*, *Cyprinus carpio*, *Ctenopharyngodon idella*), tilapia, and freshwater prawns. Aquaculture is done in ponds, tanks, cages, and pens across various states in India, with Andhra Pradesh, and West Bengal being among the leading aquaculture states. Because of the development in transportation and logistics, different initiatives by governments, increase in per capita income, the total demand for fish, mainly freshwater fish, is steadily increasing in India. Due to the high demand, various regulations have been imposed in fisheries aiming at preventing overfishing and maintaining sustainability. Fish identification is essential in the fisheries and aquaculture industry for various purposes, such as fish counting, fish size estimation for regulatory compliance, fish sorting in stock assessment, disease identification in specific fish, selection of fish species for aquaculture, fish recognition in cluttered environments for benefitting purchasers, etc. Traditionally, fish identification has been performed by trained personnel in fish morphology. Manually identifying the fish species takes a long time. There is also a scarcity of trained personnel in the fishery industry, and it also incurs errors and ambiguities in fish identification. In this context, automatic fish species identification using machine learning techniques has become a demanding research study. It is tough for the purchasers to know the nutritional information of the various fish and to find out the appropriate fish in the market that meets the nutritional needs of an individual. So, the development of an automatic framework using machine learning to identify fish with individual nutritional needs in normal fish market conditions has become very important nowadays. The advancements in technology, particularly in machine learning and computer vision, have paved the way for automated fish identification systems.

1.2 Fish Species Recognition and Related Works

Widespread applications of Fish Species recognition (FSR) initiate studying automatic recognition of fish species in the domains of pattern recognition, computer vision, image processing and segmentation. FSR has been used in various wings of the fisheries and aquaculture industry such as fisheries management, biodiversity management, fish conservation, fish nutrition profiling etc. Meanwhile, governments are applying regulations to maintain sustainable fisheries and healthy ecosystem. FSR is the process of recognising fish species based on their visual features. Morphological features play a crucial role in the recognition of fish species. Different parts of the fish body, such as the head, trunk, fins, and tails have vital visual characteristics that differentiate different fish species. FSR recognizes and categorizes fish species based on the underlying similarity with the representative image. Studying automatic recognition of fish species using machine learning or deep learning has been a scientific choice to facilitate fishery industries in recognising fish species automatically without the need for skilled personnel in fish morphology.

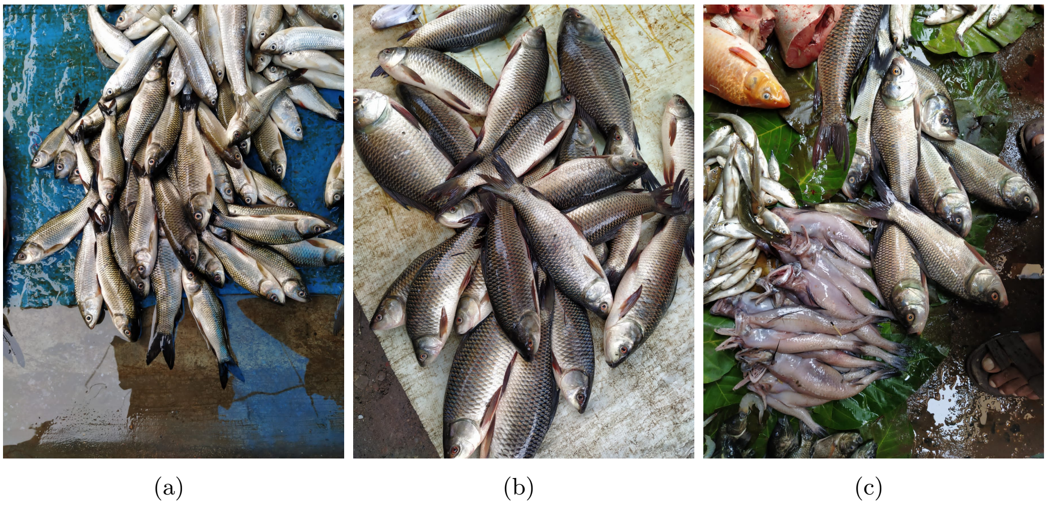
Research in automatic FSR started in 1994 by Castignolles et al. [5] to automatically classify fish species using static thresholds for segmenting fish images from the video data. Background lighting was utilized to handle contrast enhancement. Twelve geometric features were calculated and applied to the naive Bayes classifier for fish species classification. From 1999 onwards, FSR have been studied using two different approaches, machine learning, and deep learning. In 2000, Zion et al. [6] extracted several geometric features from three fish species and employed moment invariants to find optimistic results in FSR. Different landmark points of fish play an important role in FSR. In 2003, Lee et al. [7] used curvature function analysis on the fish contours for finding the critical landmark points. Mainly, the Anal fin length and adipose fin length were used to identify the fish species. Nery et al. [8] have computed four groups of features, size, shape, colour signature, and texture measurements. A total of 47 different features were extracted from these feature groups. Based on the discrimination and correlation from 47 features, 20 features were selected to achieve optimal performance in classifying six species of the Rio Grande River in Brazil. A total of 225 images of six fish species were utilized. Different shape descriptors such as line and polygon approximation, Fourier descriptors, and curvature were studied from the fish contour representation by Lee et al. [9]. Localization of the landmark points automatically is challenging and often error-prone. In many cases, these landmark points are manually computed. A fish species classification using shape and texture was proposed in 2009 by Larsen et al. [10]. A total of 108 fish images of three fish species: cod, haddock, and whiting were used. Data were present in each class with an uneven distribution. In 2010, Alsmadi et al. [11] proposed a fish recognition technique based on strong feature extraction using the size and shape of the fish species. They extracted many geometrical and distance features from some anchor points. These features were then analyzed using an Artificial Neural Network (ANN) for the goal of fish species classification. A total of 500 images from 20 fish families were used in their study. In 2012, Hu et al. [12] proposed a method based on color and texture features for fish species classification. The fish skin texture was used for the experiment (not the full body of fish species), and different statistical texture features and wavelet features were extracted from skin texture. They used multiscale Support Vector Machine(SVM) for classification purposes. A total of six common freshwater species in China were used in this work, Grass Carp (*Ctenopharyngodon idella*), Bighead Carp (*Aristichthys nobilis*), Silver Carp (*Hypophthalmichthys molitrix*), Wuchang Bream (*Megalobrama amblycephala*), Snakehead Murrel (*Channa striata*), and Red-bellied Pacu (*Colossoma brachypomum*). A total of 540 photos depicting the aforementioned fish species have been utilized. In 2013, Fouad et al. [13] proposed a system on the classification of Nile Tilapia fish using SIFT and SURF feature. A total of 96 images of Tilapia and 55 images of non-Tilapia fish were used for the binary classification. Some geometric transformations were applied on 16 images of Tilapia for the generation of the dataset. In 2014, identification of four fish species: chub, crucian, bream fish, and carp, using image processing and statistical analysis was proposed by Li and Hong [14]. The data distribution and the details of the dataset were not reported in their work. Another work using the combination of geometric features and bag of visual words (BoVW), for fish species identification was proposed in 2016 by Saitoh et al. [15]. A total of 129 fish species were used, where 20 images are present in each fish species. These images were sourced various websites, captured under diverse photography settings and environmental context. In 2016, Rossi et al. [16] presented a cloud-based mobile application that uses anchor points in fish taxonomy to recognize different fish species. The authors have collected the dataset from Turin (Italy) fish market. A total of 339 fish images were used for the seven species- *Engraulis encrasicolus* (125), *Pagellus erythrinus* (20), *Sardina pilchardu* (107), *Sparus aurata* (22), *Scomber scombrus* (18), *Merluccius merluccius* (19), and *Mullus surmuletus* (28). In 2018, Rachmatullah and Supriana [17] introduced a method for classifying low-resolution fish images by employing a architecture based on convolutional network. The authors employed data augmentation techniques, specifically rotation transformation, to evenly distribute the data throughout the different classes. Using the Fish CLEF 2015 dataset Joly et al. [18] with 15 different fish species, a convolutional network with two convolution layers, with 32 batches produced the best results out of all the convolutional networks they used in their study. In 2018, Tharwat et al. [19] introdiced a fish species idenfication system that relies on biometric data. This system utilizes the Weber Local Descriptor (WLD) and color features to achieve accurate identification. The authors used 241 fish images of four fish species: *Argyrosomus regius*, *Scomberomorus commerson*, *Sardinella maderensis* and Trachinotus. A better result was achieved using the AdaBoost classifier compared to Naive-Bayes, K-NN, and MLP algorithms. In 2019, Hussain et al. [20] porposed a modified AlexNet, specifically designed for fish species identification. This modified version of ALexNet consists of four convolutional layers and two fully connected layers. The training was conducted using the QUT fish dataset (Anantharajah et al. [21]), whereas the LifeClef 2015 Fish dataset (Joly et al. [22]) was utilized for testing and validation purposes. The experiment included a total of six fish species from both datasets: *Cirrhilabrus*, *Lethrinus*, *Thunnus*, *Epinephelus*, *Scomberoides*, and *Lutjanus*. The modified AlexNet has outperformed the original AlexNet and VGGNet. In 2019, Montalbo and Hernandez [23] proposed an optimization-based VGGNet for fish species classification. A total of 530 images from the FishBase Montalbo and Hernandez [23] collection were utilized, along with augmented images, for the experiment. In the same year, Rauf et al. [24] introduced a 32-layer convolutional neural network architecture that is based on the VGGNet network. The purpose of this network is to automatically identify fish species. A dataset named Fish-Pak, that consists of 915 images of six fish species: Grass carp (*Ctenopharyngodon idella*), Mori (*Cirrhinus mrigala*), Common carp (*Cyprinus carpio*), Rohu(*Labeo rohita*), Catla (*Catla catla*), and Silver carp (*Hypophthalmichthys molitrix*), was designed by the authors. The images are taken from three regions of the fish body: Head region, Body region, and scale. Out of the total 915 images, the authors have selected 438 images for their experiment. The numbers of images in the head region, body region, and scale region are 140, 124, and 174. A 32-layer deep CNN architecture has outperformed some popular deep networks like AlexNet, GoogleNet, ResNet50, Lenet-5, and some variations of VGGNet architecture. It is evident from the previous works on fish species classification that the datasets developed are purely region specific. In many studies different out-of-water sea fish were utilized. Very few studies have used freshwater fish species, Hu et al. [12] used some freshwater fish species of China, and Rauf et al. [24] used six carps of Asia. Though the fish species used by Rauf et al. [24] are found in India, but the dataset they developed is very small and highly imbalanced. In this thesis work, creation of a standard dataset comprising freshwater carps is the major part. Also, underwater fish recognition and classification have been popular research problems due to the widespread applications such as aquaculture and fisheries management, ecological conservation, fish tracking and fish behaviour analysis, tourist knowledge, education, and research etc. Many studies have been completed in recent years to track and recognize underwater fish species to solve various problems by Huang et al. [25], Spampinato et al. [26], Cabrera-Gámez et al. [27], Hu et al. [28], Li et al. [29], Jalal et al. [30], Jäger et al. [31], Mathur et al. [32], Mathur and Gooel [33], Zhang et al. [34], Paraschiv et al. [35], Qin et al. [36],Sun et al. [37].

1.3 Fish Species Recognition in Cluttered Environment

The basic understanding of fish species recognition is discussed in the previous section. Recognizing fish in some cluttered environments is challenging due to many inherent issues found in the fish markets. Fish markets in India are cluttered due to the placement of different fish. Generally, cluttered means not organized, untidy, and the presence of different covering objects. In the fish markets, fish are placed in an unorganized way, i.e. different type of fish are placed randomly in a congested area. Moreover, the body of one fish is occluded by other fish and in this way maximum body portion of fish is not visible. The colour changes, untidiness, and presence of other fish or non-fish objects make the fish markets inherently cluttered. Some instances of cluttered fish markets are shown in Figure 1.1.

1.4 Motivation

In recent years many studies have been carried out on FSR as discussed in the subsection 1.2. The majority of the studies were performed on sea fish recognition in some controlled environments. Very few works were done on the classification of fish species in India. Recognition of freshwater fish under some controlled environment was studied by Hu et al. [12] and Rauf et al. [24]. These two studies have been done on the fish species from China and Pakistan. The Fish-Pak dataset developed by Rauf et al. [24] consists fish species available in India have huge demand because of availability, reasonable price, and nutritional benefits. The size of the dataset is very small and imbalanced. Only 271 images



(a) (b) (c)

Figure 1.1: Some images of cluttered fish markets in India

of fish body view are available for the six fish species present in the Fish-Pak dataset. So, development of a standard dataset for freshwater fish species of India is mostly important to address. After the detailed literature survey, it is evident that no work has been done on freshwater fish species recognition under a cluttered environment. Also, it may be the first study to recognise carps in India. Fish markets in India are incredibly lively and busy during the daytime. These markets are generally located in cluttered environments among the daily bustle and are typically filled with a variety of freshwater fishes, including a diverse range of fish species such as carps, catfishes, snakeheads etc. that are both locally sourced and imported from other regions. In such markets, the environment can be cluttered and chaotic, with sights, sounds, and smells creating a sensory overload. Under these circumstances, a method for automatic FSR in some cluttered environments becomes an important research topic to help the common purchasers as well as the fisheries and aquaculture industries in a variety of contexts. In cluttered environments, fish are placed in unorganized ways, and many parts of a fish body are hidden due to the presence of other fish bodies. Also, in live fish markets, the presence of different fish species, non-fish objects, illumination changes, lighting differences, and congestion make the automatic recognition problem a very challenging job. The main research gap in FSR is the non avialbility of publicly standard dataset and corresponding research in recognizing different freshwater carps, specially in a cluttered environment. Also, in the Indian context, this research will be the first to develop a method for automatically recognizing fish species in some cluttered environment.

Around the world, the concept of a balanced diet remains crucial for maintaining a healthy life. In India, dietary patterns can vary widely based on cultural, regional, and socioeconomic factors. Along with whole grains, fruits and vegetables, legumes and pulses, fish is a great source of proteins, different vitamins, minerals, Omega-3 fatty acids etc. In Indian cuisines, particularly in the coastal areas fish is consumed daily. It is very hard for consumers to remember the various nutritional benefits of different fish species, recognize them according to the nutritional requirement of an individual. Hence automatic recognition of fish species which can meet the nutritional needs of a consumer is an essential research problem to address. In this thesis, automatic recognition of some freshwater carps using machine learning technique is addressed and a case study on selecting appropriate fish species from fish market's cluttered environment meeting individual's nutritional needs is done.

1.5 Scope and Contributions

Before diving into the different tasks on recognising fish species that have been done in this thesis, the development of four standard datasets was undertaken in the initial stages of the research. The first two datasets named **JUDVLP-WBUAFS:** Fishdb-IMC.v1 and **JUDVLP-WBUAFS**: Fishdb-EC.v1, is prepared for the recognition of three Indian major carps (IMC) and Indian exotic carps (EC) in the non-cluttered environment. These two datasets are the first dataset developed for the recognition of carp in the Indian context. Machine learning or deep learning applications need a standard dataset to learn a generalized model. In the cluttered environment, there are no datasets available for the recognition of different carps. A dataset named **JUDVLP-WBUAFS**: Fishdb-Detection.v1 is developed for recognizing six different freshwater fish in live fish markets under some cluttered conditions. Only the heads of the fish are annotated using a rectangular box and used for detection and recognition purposes. Segmentation plays a crucial role in eliminating the extra information from the images except the object of interest. A semantic segmentation framework is done in this thesis and to accomplish that a dataset named **JUDVLP-WBUAFS**: Fishdb-Segmentation.v1 is developed with the heads of the fish annotated using polygons for segmentation purposes. One of the major objectives of this thesis is to develop all the datasets in the preliminary phases such that the experiments can be done to achieve the principal objective of recognizing the fish species in non-cluttered and cluttered environment.

Apart from the dataset development, various works have been done on this research study. Briefly, the entire process of automatic recognition of freshwater fish is conducted in two stages, one is under non-cluttered environment and the other is under the cluttered environment. Using the latent representation of deep convolutional autoencoder and traditional machine learning algorithms three Indian major carps (IMC) are classified in non-cluttered environment. In an autoencoder, the latent representation refers to the compressed lower dimensional form of the input data that the autoencoder learns. This latent representation serves as a bottleneck in the network, capturing the most salient features of the input data while discarding unnecessary details. Using the latent representation learned by an autoencoder as features in machine learning tasks can be quite effective, especially in scenarios where the original data is high-dimensional or noisy. One of the primary benefits of using autoencoder latent representations is dimensionality reduction. By compressing the input data into a lower-dimensional latent space, the autoencoder captures the most salient features of the data while discarding irrelevant or noisy information. This can be particularly useful when dealing with high-dimensional data, as it reduces the complexity of the feature space and can lead to more efficient and effective learning. Transfer learning facilitates the adaptation of models from one domain to another. A model trained on a large dataset can be adapted to another related dataset with proper fine-tuning on a small dataset in the target problem. Transfer learning allows leveraging pre-trained models that have been trained on large datasets. This reduces the need for massive amounts of labelled data and computationally intensive training, especially in domains where data is scarce or expensive to acquire. Pre-trained models capture generic features from large datasets, which can generalize well across related tasks or domains. Fine-tuning these models on task-specific datasets can lead to improved performance compared to training from scratch, as the model starts with knowledge gained from previous tasks. Initializing the model with parameters learned from pre-trained models allows the model to start from a point closer to the optimal solution. This often leads to faster convergence during training, as the model requires fewer iterations to adapt to the new task or dataset. Transfer learning acts as a form of regularization, helping to prevent overfitting on small datasets. Pre-trained models have typically learned robust representations from diverse data, which helps to regularize the learning process when fine-tuning on small datasets. Different state-of-the-art deep learning models for image classification are employed with the transfer learning approach to identify the three exotic carp in the non-cluttered environment. Under the cluttered environment in the live fish markets, recognition of the fish species is a challenging problem due to the presence of many fish hiding the other fish around it, illumination changes, background noise etc. The study on recogniton of six carps under cluttered environments is done in this thesis work using object localization and identification approach, and semantic segmentation approach. Also, a mobile application is developed for the ease of consumers to identify fish according to their nutritional needs and then select the fish that best suits them. The entire research work mainly helps the fishery industry by proposing an automatic technique for fish identification, fish counting, analysis of fish catching, etc. This study will be very helpful to the purchasers in the market, as they will recognize different fish in the market without taking help from someone and they can also select the fish according to the nutritional need to maintain a balanced and healthy diet. From the brief introduction, it is evident that this thesis work is beneficial on the economic and societal side.

1.6 Thesis Organization

The thesis consists of six chapters, and a list of references overall. The Chapter 1 of the thesis briefly introduces fish production in India, its importance, consumption, and benefits regarding maintaining a healthy life. The need to develop an automatic technique for the recognition of different fish species in non-cluttered and cluttered environment is explained in this chapter. It also provides a brief overview of each work that has been done

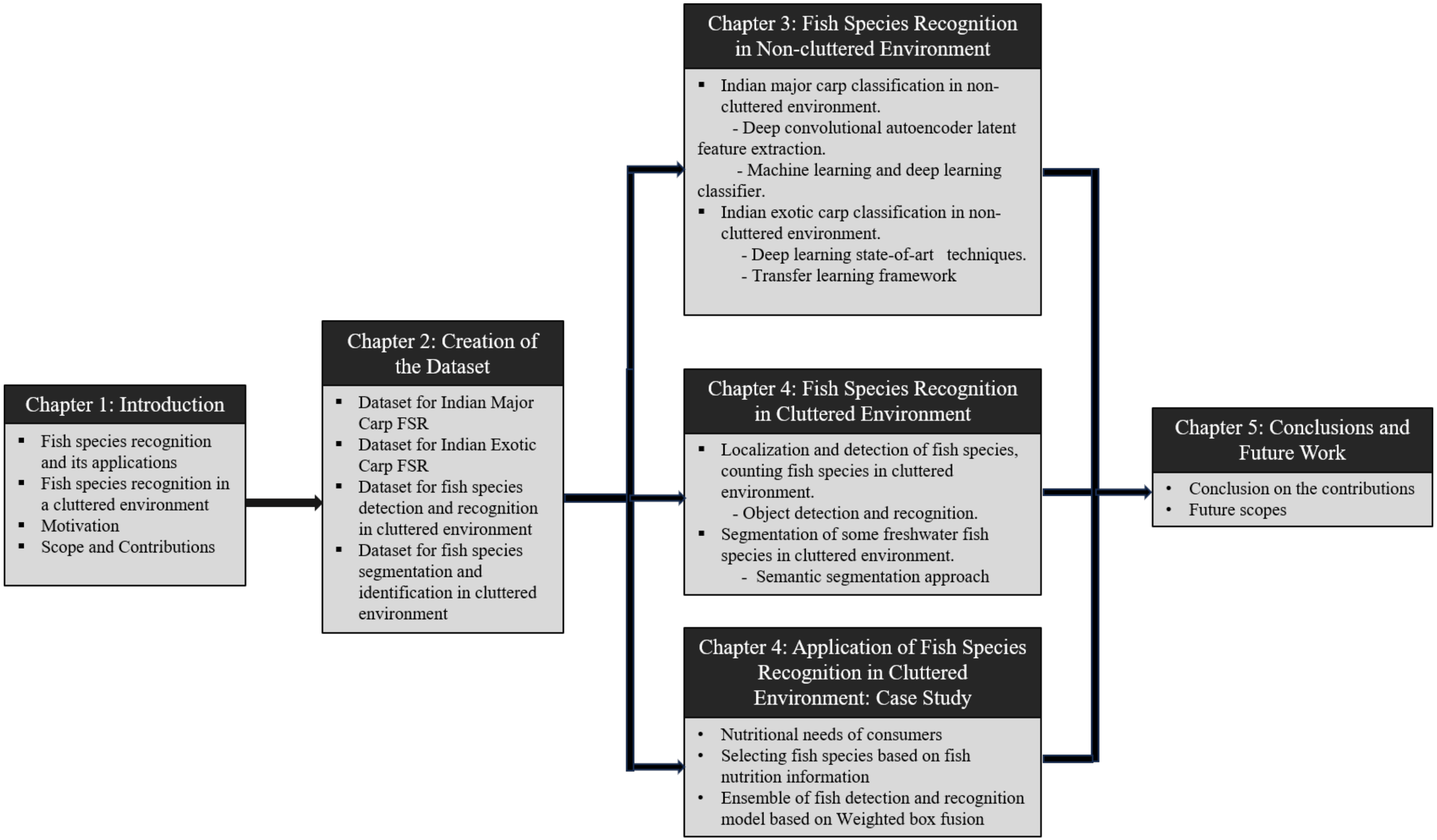

Chapter 1: Introduction
▪ Fish species recognition and its applications
▪ Fish species recognition in a cluttered environment
▪ Motivation
▪ Scope and Contributions
Chapter 2: Creation of the Dataset
▪ Dataset for Indian Major Carp FSR
▪ Dataset for Indian Exotic Carp FSR
▪ Dataset for fish species detection and recognition in cluttered environment
▪ Dataset for fish species segmentation and identification in cluttered environment
Chapter 3: Fish Species Recognition in Non-cluttered Environment
▪ Indian major carp classification in non-cluttered environment.
- Deep convolutional autoencoder latent feature extraction.
- Machine learning and deep learning classifier.
▪ Indian exotic carp classification in non-cluttered environment.
- Deep learning state-of-art techniques.
- Transfer learning framework
Chapter 4: Fish Species Recognition in Cluttered Environment
▪ Localization and detection of fish species, counting fish species in cluttered environment.
- Object detection and recognition.
▪ Segmentation of some freshwater fish species in cluttered environment.
- Semantic segmentation approach
Chapter 4: Application of Fish Species Recognition in Cluttered Environment: Case Study
• Nutritional needs of consumers
• Selecting fish species based on fish nutrition information
• Ensemble of fish detection and recognition model based on Weighted box fusion
Chapter 5: Conclusions and Future Work
• Conclusion on the contributions
• Future scopes

Figure 1.2: Diagrammatic representation of the thesis organization

in this research study. The chapter finally ends with the scope, objective, and application area of the topic undertaken in this study and the organization of the thesis.

Chapter 2 starts with the details of the available datasets for freshwater fish species recognition. The location, name of the fish species, number of data samples, data split and augmentation technique are explained in detail. Then the dataset collection process, and pre-processing, of the four benchmark datasets are presented with appropriate diagrams. It ends with a conclusion of all the datasets with their application areas.

Chapter 3 presents the automatic recognition of Indian major carps (IMC) and exotic carps (EC) using deep learning techniques. The use of latent representation of deep convolutional autoencoder is studied for extracting feature sets for recognition of IMC is explained in detail with experimental details and results. Also, the available methods for fish species recognition are explained and compared wherever applicable. Then Some state-of-the-art deep learning algorithms with transfer learning strategies are applied for recognizing three exotic carps, is presented with experiment details and results. Each of these two works is explained individually with a conclusive ending. The chapter then ends with an overall conclusion on automatic FSR in the non-cluttered environment and possible solutions to the drawbacks of these studies.

Chapter 4 consists of the need for fish identification in some cluttered environments. It presents two separate works on automatic FSR in cluttered fish market conditions by applying object localization, identification and semantic segmentation. The annotation process, details of the deep learning techniques applied, experiment strategies, and results analysis are explained for both studies are explained in detail. The chapter ends with the overall summary of automatic FSR under some cluttered environments.

Chapter 5 presents a case study on the development of a mobile application for consumers, that recognizes the different fish in live fish market's cluttered environment according to consumer's nutritional needs. A framework consisting of object localization and detection with the fish nutrition knowledge base, to help consumers to select fish that best meet their nutritional requirements, is explained with methodology, experiment design, and results. The chapter ends with a conclusion regarding the pros of such applications to the consumers and the future challenges to tackle.

Chapter 6 presents the final discussion on the outcomes of the thesis work and future scopes that could be explored in this area. The overall thesis organization is represented diagrammatically in Figure 1.2.

All the relevant references are given at the end of the thesis.

Chapter 2

Creation of the Dataset

A well-diversified dataset is essential to any machine learning or deep learning application. Chapter 1 discusses the scarcity of publicly accessible datasets on FSR, which do not align with the research in this thesis. Therefore, several datasets are developed to conduct studies on automatic FSR in both non-cluttered and cluttered environments. Two datasets named **JUDVLP-WBUAFS**: Fishdb-IMC.v1 and **JUDVLP-WBUAFS**: Fishdb-Ec.v1 are created for automatic recognition of Indian major carps and exotic carps in a non-cluttered environment. Another two datasets named **JUDVLP-WBUAFS**: Fishdb-Detection.v1 and **JUDVLP-WBUAFS**: Fishdb-Segmentation.v1 are created for automatic recognition of different carps in cluttered environments. This chapter explains the image collection procedure, pre-processing, and annotation process of the individual dataset.

2.1 Dataset for Indian Major Carp Freshwater FSR

India is home to a rich diversity of freshwater fish species, and carps are among the major contributors to the country's fisheries. Carp species are widely distributed in various water bodies, including rivers, lakes, and ponds, and they play a significant role in the aquaculture sector. Regularly farmed carp species of Indian origin are Rohu (*Labeo rohita*), Catla (*Labeo catla*), and Mrigal (*Cirrhinus mrigala*) and are collectively known as Indian Major Carps (IMC). Rohu is a widely popular indigenous carp species in India and is extensively farmed in freshwater bodies. It is popular for its tender flesh and mild flavor. It is dominantly used in many Indian households in the form of curries, fish fries, etc. Catla is another IMC species that is commonly farmed in ponds and rivers. It is recognized for its large size and distinctive taste. It is often a popular choice in Indian cuisine. Mrigal is widely distributed in rivers, lakes, and reservoirs with slow to moderate currents across the Indian subcontinent. It is an important component of the aquatic ecosystem, contributing to nutrient cycling. It holds economic significance in the aquaculture sector due to its fast This fish is valued for its growth and adaptability to a variety of environmental conditions. This fish is valued for its flesh, which is white, firm, and has a mild flavor. These three IMC fish species play a vital role in both natural ecosystems and aquaculture systems in India and contribute to the nutritional needs of the population. These species become a significant component of the country´s fisheries, due to their economic importance, adaptability, and culinary value.

Table 2.1: Different publicly available datasets for fish species recognition

Dataset name	Characteristics	Type	Location
FISH4K (F4K)	23 species, and 27,370 images	Sea fish (underwater)	Taiwan
Croatian Fish	12 species, and 794 images	Sea fish (underwater)	Croatia
LifeCLEF 2014	10 species, video data with annotation for fish instances.	Sea fish (underwater)	Taiwan
LifeCLEF 2015	15 Species, total of 93 videos, and approx 20,000 images.	Sea fish (underwater)	Taiwan
NCFM	6 species, and two other category named no fish, and other fish category. Total of 16,915 images present in the dataset.	Sea fish (underwater)	Norway
Fish-Pak	6 species, 915 images of three distinct views, fish body, scale, and head. Imbalance data distribution between three distinct views. No evidence that images are taken from same fish.	Freshwater fish (out of water)	Pakistan
Large scale fish dataset	9 species, 50 images each in the seven fish species, and 30 images each in 2 fish species. After augmentation 1000 species in each fish species.	Sea fish (out of water)	Turkey

2.1.1 Characteristics of Fish Species

The characteristics of fish species are crucial for various reasons, spanning ecological, economic, biodiversity, aquaculture management, and food security and nutrition aspects. Understanding these characteristics provides valuable insights into the biology, behavior,

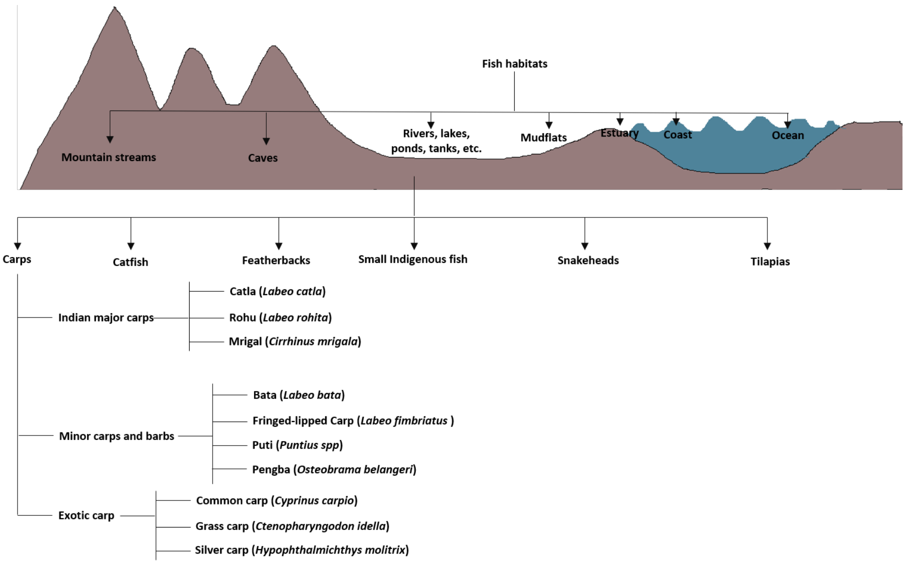


Figure 2.1: Fish habitats of Indian fish and classification of fish found in rivers, lakes, ponds, tanks etc.

and ecological roles of different fish species. In this section, some major characteristics of IMC are discussed.

- The body of Rohu is moderately elongated, with an inferior setting in the mouth. The lips are fairly thick, fringed, and have a noticeable inner fold. The dorsal fin ends either inline or somewhat anterior to the anal fin, placing it anterior to the pelvic fins. The scales are modest and cycloid. The sides and belly area of the body are silver in color, while the back is blue to brownish in hue. Overall, the weight range of the Rohu in fish markets is 500 to 2.0 kg.

- The body of Catla is short, relatively deep, has a large head, and a rounded abdomen. The mouth is wide with a lower jaw, and the lower jaw consists of movable articulation at the symphysis. Throughout the body, large cycloid scales are present with a bluntly rounded snout. The body color is grayish on the back, silvery-white below, and the fins are dusky.

- Mrigal's body is elongated, with a depressed and obtusely rounded snout. The mouth is terminal and wide, and discontinuity is there between the upper and lower lips. The scales of mrigal are large and cycloid. The caudal fin is homocercal and deeply forked. On the back part, the body color is dark gray and silvery on the sides and belly area.

2.1.2 Collection Process

In this proposed work, a total of 1500 fish images have been collected from the Garia fish market and Mukundapur fish market situated in Kolkata, India, under the supervision

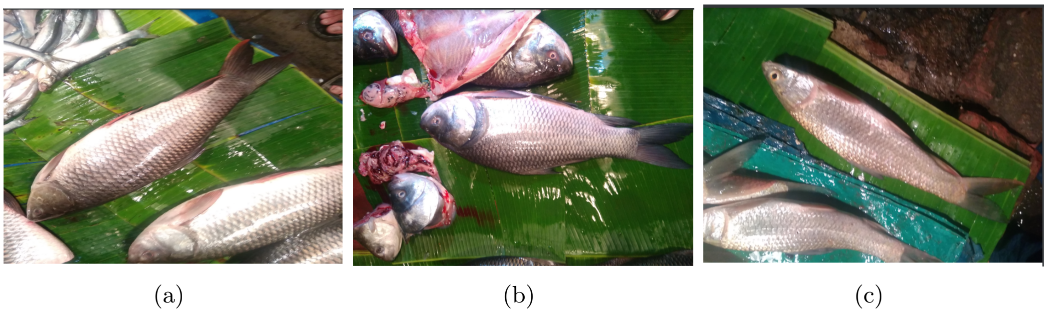


Figure 2.2: (a) Rohu (*Labeo rohita*) (b) Catla (*Labeo catla*) (c) Mrigal (*Cirrhinus mrigala*)(Banerjee et al. [1])

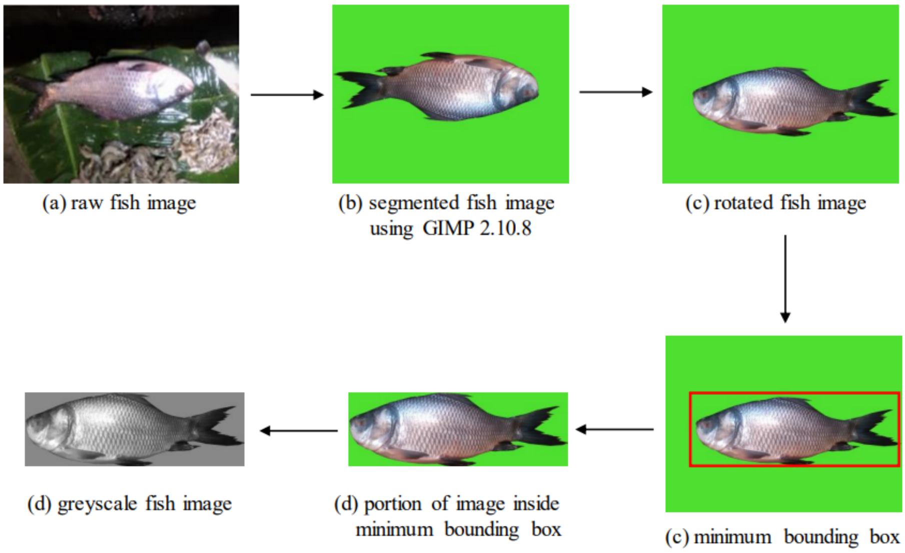

(a) raw fish image
(b) segmented fish image using GIMP 2.10.8
(c) rotated fish image
(c) minimum bounding box
(d) portion of image inside minimum bounding box
(d) greyscale fish image

Figure 2.3: Some images of **JUDVLP-WBUAFS**: Fishdb-EC.v1 dataset before and after pre-processing(Banerjee et al. [1])

of West Bengal Fisheries Science, processed and annotated at Jadavpur University, DVLP lab, under the Center for Microprocessor Application and Training and Research (CMATER). The images were used to develop the dataset named **JUDVLP-WBUAFS**: Fishdb-IMC.v1, which consists of the 1500 images categorized under three popular major carp fish species: Rohu (*Labeo rohita*), Catla (*Labeo catla*), and Mrigal (*Cirrhinus mrigala*). A normal smartphone camera (Redmi5A mobile phone, 16-megapixel camera) was used to capture the fish images from the market. Images were collected at different times (day/afternoon/night) under different illumination conditions, lightning conditions, rotation angles, and noisy conditions. The size of the fish is different in the dataset, so the samples were carefully distributed into training, testing, and validation parts so that different variations are kept in all parts. This concept is based on the stratified sampling of the datasets. A total of 500 images are present under each carp species. Some raw images from the dataset are shown in Figure 2.2.

2.1.3 Preprocessing

Collecting fish images from the local fish market presents significant challenges due to the rush in the market, noisy environment, minimum light, and presence of numerous fish species in the shop. All the variations were kept in preparing a challenging dataset for fish species identification. Figure 2.2 clearly shows the presence of some other fish species in addition to the targeted three fish species. Additionally, mutiple fish samples are present in the image (refer to 2.2(c)). In this study, the aim is to identify fish species from the fish image where only one fish was present. The 'inversion' tool of GIMP 2.10.8 software is used on the marked boundary of the targeted fish to divide the image into two regions: targeted fish and background. In the background of each image, the green color is used. Each image was rotated to place the head of the fish on the left, the dorsal fin at the top, and the body in a horizontal position to make a uniform dataset. In many images, the green background takes up most space, resulting in inaccurate prediction outcomes. Hence, an automated algorithm was applied to generate the minimum bounding box around the targeted fish. The portion inside the minimum bounding box was converted into grayscale and the final dataset was prepared. In Figure 2.3, the overall pre-processing step is pictorially presented using a single raw fish image. The fish dataset contains a total of 1500 images, comprising 500 images of each fish species: Catla (*Labeo catla*), Rohu (*Labeo rohita*), Mrigal(*Cirrhinus cirrhosis*). As there are variations in size and shape among the fish species, the final images may be resized as per the requirements of the experiment.

2.2 Dataset for Indian Exotic Carp Freshwater FSR

In Indian fisheries and aquaculture, Exotic carp (EC) typically refers to carp species that are not native to India but have been introduced for aquaculture or other purposes. These exotic carp species have been imported from other countries due to their potential for commercial production, adaptability to local conditions, and desirable traits such as fast growth rates or high market demand. The most commonly consumed exotic carps in India are Common carp (*Cyprinus carpio*), Garss carp (*Ctenopharyngodon idella*), and Silver carp(*Hypophthalmichthys molitrix*). Common carp is native to Europe and Asia, including parts of India; it is considered an exotic species in certain regions where it has been introduced for aquaculture purposes. It is highly valued for its rapid growth, adaptability to various environmental conditions, and delicious flesh, making it a popular choice for aquaculture in India. Grass carp is native to eastern Asia but has been introduced to many countries, including India, for aquatic weed control and aquaculture. The ability of this herbivorous species to control aquatic vegetation makes it valuable for maintaining water quality in ponds and lakes. India and other countries have introduced the native East Asian Silver carp for aquaculture purposes. Its rapid growth, high fecundity, and efficient conversion of phytoplankton into protein make it a valuable species in polyculture systems. These exotic carp species have contributed significantly to the expansion and diversification of aquaculture in India, providing additional options for farmers and helping

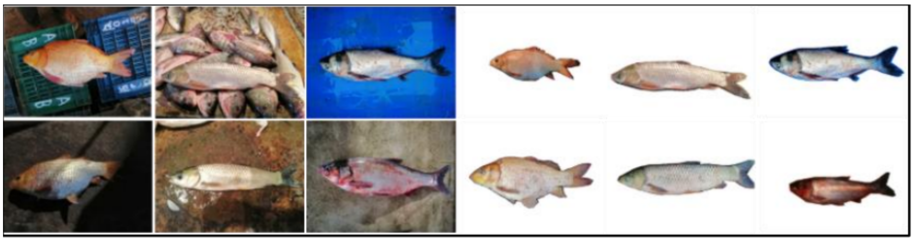


Figure 2.4: Preprocessing of process on the dataset(Banerjee et al. [2])

meet the growing demand for fish protein. However, their introduction also raises concerns about potential ecological impacts, including competition with native species and genetic pollution. Therefore, responsible management practices are essential to minimize negative consequences and ensure the sustainable use of exotic carp species in Indian fisheries and aquaculture.

2.2.1 Characteristics of Fish Species

The characteristics of fish species encompass a wide range of biological, ecological, and behavioural traits that define their role in aquatic ecosystems and their interactions with the environment. This section explains the key characteristics of some exotic carp taken into account.

- **Common carp (*Cyprinus carpio*)**: Common carp typically have a robust, elongated body with a slightly arched dorsal profile and a laterally compressed shape. It exhibits a wide range of color variations depending on environmental factors, including water quality and habitat. Wild-type individuals are typically olive-green or brownish, while selectively bred varieties may display colors such as gold, orange, or even white. They are highly adaptable and can thrive in various freshwater habitats, including rivers, lakes, ponds, and reservoirs. For centuries, people have extensively cultivated Common carp for food and recreational angling.

- **Garss carp (*Ctenopharyngodon idella*)**: Grass carp have elongated, cylindrical bodies with a slightly flattened belly and a streamlined shape. They typically have large, slightly oblique mouths with no teeth in the jaws but have pharyngeal teeth adapted for grinding plant material. Their scales are large and rough, and they lack barbels. Grass Carp typically have an olive-green to dark greenish-gray coloration on their backs, fading to a lighter shade on the sides and belly. The fins are usually dusky or dark. Grass carp are primarily found in freshwater environments, including rivers, lakes, reservoirs, ponds, and canals. Grass carp are widely used in aquaculture for vegetation management in ponds, lakes, and reservoirs. They are often stocked in water bodies to control excessive aquatic weed growth, which can impede water flow, reduce oxygen levels, and degrade habitat quality.

- **Silver carp(*Hypophthalmichthys molitrix*)**: Silver carp have a deep and laterally compressed body, giving them a streamlined shape. They have a large, upturned mouth with no teeth and a distinctive terminal position. The scales are small and silver in color, giving the fish its name. Silver carp are primarily found in freshwater environments, including rivers, lakes, reservoirs, ponds, and canals. They are widely used in aquaculture for their ability to efficiently convert phytoplankton into protein. They are often stocked in ponds, lakes, and reservoirs to enhance water quality and control algae blooms. Silver carp have been introduced to various countries for biological weed control, particularly in situations where excessive algae growth threatens water quality or impedes recreational activities.

2.2.2 Collection Process

In machine learning-based applications, creating a dataset to address a problem is a major step. Images of Common carp (*Cyprinus carpio*), Grass carp (*Ctenopharyngodon idella*), and Silver carp(*Hypophthalmichthys molitrix*) had been collected from different fish markets like Patipukur fish market (North 24-Parganas- West Bengal), Annapurna fish market (Purba Medinipur-West Bengal), and Chak Bazar (Bankura-West Bengal) using the 16-megapixel camera under different lighting conditions. During image collection, the fish were kept in a relaxed position, and the whole fish body was taken from the tip of the mouth to the end of the caudal fin. Around 1500 images of these species were gathered and sorted based on factors such as haziness, defective body parts, poor lighting, etc. After screening the images, several images were removed due to some quality issues, and a dataset of 1225 images was created for the experiment. The collection of images was under the supervision of West Bengal Fisheries Science, processed and annotated at Jadavpur University, DVLP lab, under the Center for Microprocessor Application and Training and Research (CMATER). A dataset named **JUDVLP-WBUAFS**: Fishdb-EC.v1 was created using these images.

2.2.3 Preprocessing

The collection of images was done in an unconstrained environment, which resulted in a variety of backgrounds and the presence of other fish species or unintended objects. The images have been manually segmented to divide the image into two parts: the targeted fish body and the background. The background of every image was set to white to maintain the homogeneity of the dataset. The 'inversion' tool of GIMP 2.10.8 software was utilized to segment the images into targeted fish bodies and backgrounds. White color was used in the background of every image. Rotation transformation was used to place the fish body in a horizontal direction with the head of the fish at the left and the dorsal fin at the top. Automatic cropping was done on every image to remove extra background spaces. Figure 2.4 shows some of the images from the dataset before and after applying the preprocessing.

Fish species	Number of instances	% in total data
Labeo catla	371	0.119
Labeo rohita	708	0.226
Cirrhinus mrigala	1826	0.584
Labeo bata	140	0.045
Hypophthalmichthys molitrix	52	0.016
Ctenopharyngodon idella	31	0.010

Table 2.2: Total Number of Instances per Fish Species in the **JUDVLP-WBUAFS**: Fishdb-Detection.v1 Dataset

2.3 Dataset for Fish Species Detection and Identification in Cluttered environment

2.3.1 Motivation

Recognizing fish species is tough for consumers. The appropriate knowledge of fish taxonomy and experience in identifying fish species plays a key role in FSR. In fisheries industries, the recognition of fish species and species-wise counting are important tasks that require trained personnel. Manually identifying the species and counting is a monotonous process that requires a lot of trained manpower. Most of the fish markets in India are unorganized, and during the daytime, there is a moderate crowd seen in the fish market. Due to a lack of time when purchasing fish at the market, consumers frequently struggle to identify them and may require assistance from someone with expertise in fish taxonomy. However, obtaining immediate assistance from experts is not always possible, necessitating self-sufficiency and the development of a suitable dataset for identifying fish species in the fish market's cluttered environments. As per my knowledge, no dataset was available for the recognition of freshwater fish species in some cluttered environments at the time of the study.

2.3.2 Collection Process

As no standard dataset for the specific task was available, images of six fish species, Catla (*Labeo catla*), Rohu (*Labeo rohita*), Mrigal (*Cirrhinus mrigala*), Bata (*Labeo bata*), Silver carp (*Hypophthalmichthys molitrix*), and Grass carp (*Ctenopharyngodon idella*), were collected from different fish markets in West Bengal in the daytime under unconstrained conditions. The data collection process was done under the supervision of West Bengal Fisheries Science, processed and annotated at Jadavpur University, DVLP lab, under the Center for Microprocessor Application and Training and Research (CMATER). There was a high variation in light intensity in the different areas of the market, as well as in various fish markets. Some normal phone cameras, like the Redmi Note 8 Pro, Samsung Galaxy A13, Samsung Galaxy J5, and Lenovo Note 5, were used in image collection. Generally, each image contains different fish species, some of which were considered in this study and some not. The images contain different fish species with a variety of sizes

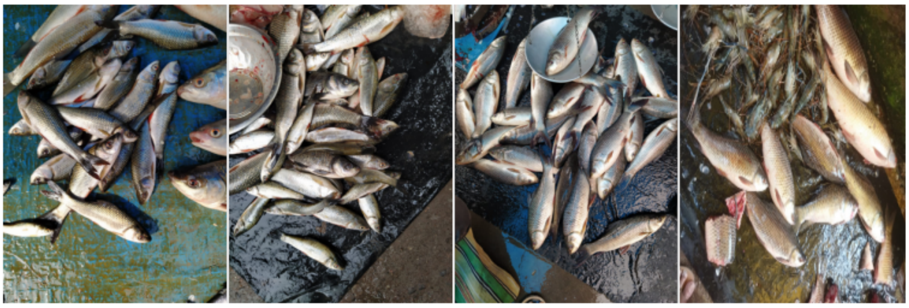


Figure 2.5: Some sample images of the **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset(Banerjee et al. [3])

(intraclass and interclass), viewing angles, light intensity, scales, etc. In live fish markets, the close proximity of various fish species often obscures the majority of their body parts. A total of 500 images were collected from the fish market. After the collection of data, all the data were analyzed, and 100 images were discarded due to reasons like extreme noise present in the image, blurriness, the presence of some external object, etc. A total of 3,128 fish specimens were present in the 400 images considered in the dataset (refer Table 2.2). Out of the six fish species, Rohu class have the most instances, and Grass carp class have the least number of instances. This dataset of 400 images with 3,128 instances was named **JUDVLP-WBUAFS**: Fishdb-Detection.v1. Some sample images of **JUDVLP-WBUAFS**: Fishdb-Detection.v1 is presented in Figure 2.5.

2.3.3 Challenge in the Dataset

Because the images were collected in cluttered environments, many of the fish species are obscured by surrounding fish species. In most cases, the truck and tail portion of the fish are not visible due to suppression by other surrounding fish species. These make the recognition problem challenging to solve. If we go through the sample images in Figure 2.5, the challenges can be understood. Additionally, variations in lighting in the fish market, variations in fish size, a variety of backgrounds, and data imbalances make the problem both challenging and interesting. To address the visibility issue, only the head part of the fish was annotated as it is a prominent visible part in all fish species and plays a crucial morphological role in fish species identification.

2.3.4 Preprocessing and Annotation Process

As shown in Figure 2.5 different fish species are clustered in a small area, and many fish species are obscured by other fish species. Therefore, only the visible head portions of fish species were annotated by a rectangular bounding box using the LabelMe tool. Several domain experts with proper knowledge of fish taxonomy verified all the annotations. The generated JSON files were then processed, and the YOLO-specific annotation format was prepared. The bounding boxes were normalized in the following format: class, X_{center}, Y_{center}, width, height. In this study, the fish species were mapped to a numeric

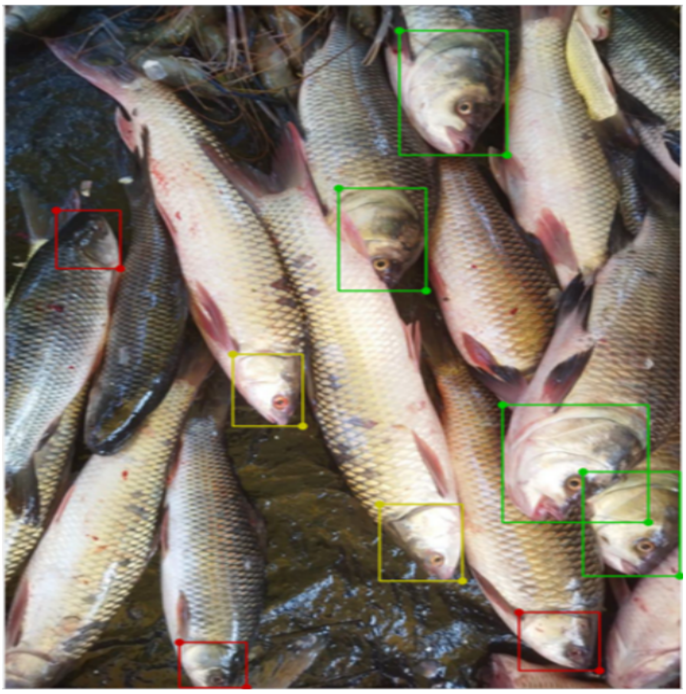


Figure 2.6: A sample image of **JUDVLP-WBUAFS**: Fishdb-Detection.v1 after the annotation process: red- Labeo rohita, green-Labeo catla, yellow-Cirrhinus mrigala (Banerjee et al. [3])

class value as: 0-*Labeo catla*, 1-*Cirrhinus mrigala*, 2-*Labeo rohita*, 3-*Labeo bata* , 4-*Hypophthalmichthys molitrix*, 5-*Ctenopharyngodon idella*. Figure 2.6 shows a sample image after the annotation is done.

2.4 Dataset for Fish Species Segmentation and Identification in Cluttered Environment

2.4.1 Motivation

Automatic identification of the different fish species in a cluttered environment helps the fishery industry, as well as the common people and other stakeholders. Semantic segmentation plays a crucial role in fine-grained understanding and improved localization. However, no such dataset was available in the literature. That was the main motivation to develop a dataset for fish recognition in cluttered environments using semantic segmentation. The segmentation process, similar to the **JUDVLP-WBUAFS**: Fishdb-Detection.v1, utilized only the fish heads.

2.4.2 Collection Process

As no standard dataset was available to work on freshwater fish species segmentation in some cluttered environments, around 800 images were collected from different live fish markets in West Bengal. In fish markets, different fish are available, including freshwater and sea fish. The collected images contained different species with a wide variety in size

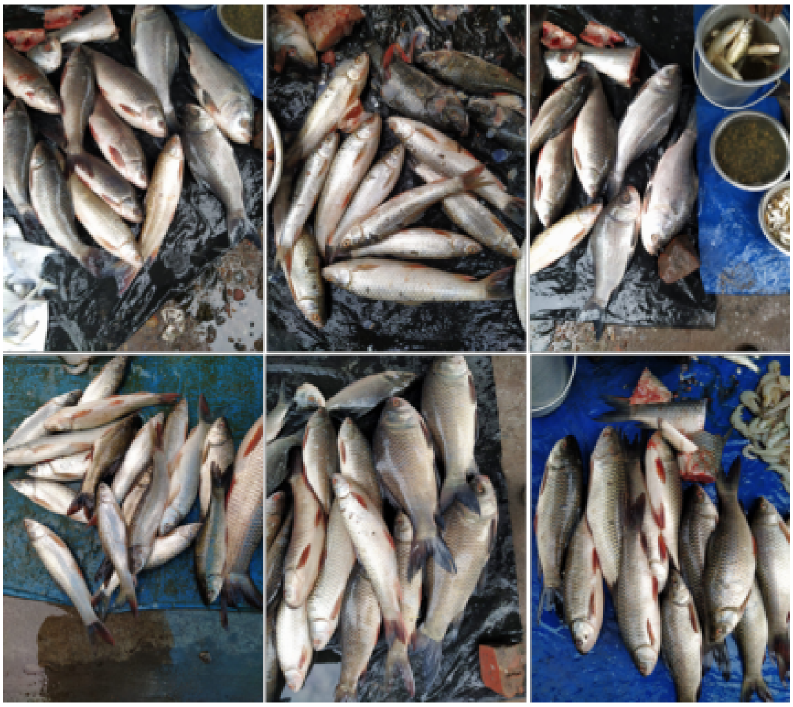


Figure 2.7: Some raw images in our proposed dataset for semantic segmentation task (Banerjee et al. [4])

(intraclass and interclass), viewing angle, color, and lighting condition. Using different Android phones, like the Redmi Note 8 Pro, Samsung Galaxy A13, Samsung Galaxy J5, and Lenovo Note 5, the images were collected in an unconstrained environment. During the day, in different lighting conditions, images were collected. The presence of other species in the image makes the dataset more challenging. Out of 800 images, 200 images were used to perform the segmentation of fish species. Five fish species named Catla(*Labeo catla*), Rohu(*Labeo rohita*), Mrigal (*Cirrhinus mrigala*), Bata(*Labeo bata*), Silver carp (*Hypophthalmichthys molitrix*), and Grass carp (*Ctenopharyngodon idella*) were considered. The data collection process was done under the supervision of West Bengal Fisheries Science, processed and annotated at Jadavpur University, DVLP lab, under the Center for Microprocessor Application and Training and Research (CMATER). The dataset is named as **JUDVLP-WBUAFS**: Fishdb-Segmentation.v1. Some raw images of the **JUDVLP-WBUAFS**: Fishdb-Segmentation.v1 are shown in Figure 2.7.

2.4.3 Challenge in the Dataset

As the images were collected in a cluttered environment, many fish species were obscured by each other. In many cases, the majority of the fish's body portion was not visible or was being covered by other fish around it. The problem becomes challenging due to the presence of different fish species in a frame, different lighting conditions, variations in the size of the fish, occlusions, etc. The semantic segmentation technique was applied to segment the fish heads that are visible in the image. According to the fish taxonomy, the head, body, and tail portion play an important role in identifying a fish species. However,

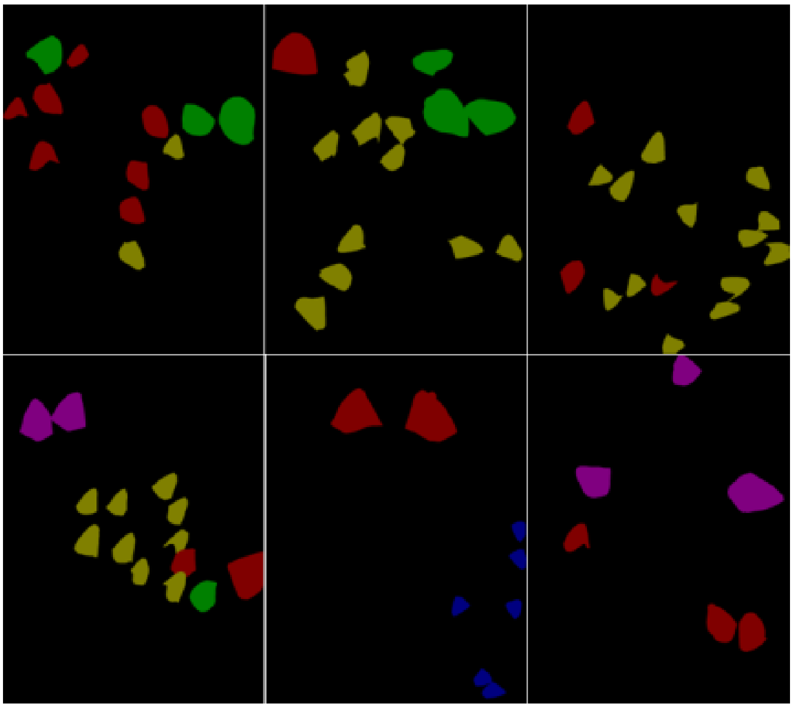


Figure 2.8: Some ground truth segmentation masks: black color represents background pixels (Banerjee et al. [4])

in most of the images collected, the tail part and the maximum portion of the fish's body were not visible. Only the head portion of the fish is often visible in a cluttered environment. The head portion of different fish species is quite different in shape and size and plays a crucial role in identifying fish species at a single glance. Therefore, only the head part of the fish was considered for the segmentation purpose.

2.4.4 Preprocessing and Annotation Process

As the images were collected in a cluttered environment, many fish species were obscured by each other. In many cases, the majority of the fish body portion was not visible or was being covered by other fish around it. According to the fish taxonomy, the head, body, and tail portion play an important role in identifying a fish species. The tail and maximum of the fish body were not visible in most of the proposed dataset samples. Therefore, in this dataset, only the head part of the fish was considered for the segmentation task. In semantic segmentation, different objects in the image were annotated using polygons in different colors. The Labelme tool was used to annotate the images. Creating ground truth data with so many fish was the time-consuming part of this study. After completing the annotation process, ground truth masks were generated. The color schemes (R, G, and B) of the ground truth masks were: Labeo catla – (0, 128, 0), Labeo rohita – (128, 0, 0), Cirrhinus mrigala – (0, 128, 128), Labeo bata - (0, 0,128), Hypophthalmichthys molitrix - (128, 0, 128). The clearly visible fish heads in the images were annotated. All other fish species and objects in the image, except for these five, were not annotated. Some of the ground truth mask data are shown in Figure 2.8. In the **JUDVLP-WBUAFS**:

Fishdb-Segmentation.v1 dataset, a total of 200 images were labelled for the experiment's purpose.

2.5 Summary

In this thesis, fish species recognition is addressed in some non-cluttered and cluttered environments using different approaches. The development of the standard dataset to conduct the study is the fundamental step in the research. In this chapter, the data collection, data pre-processing, data annotation and labelling are discussed for each dataset developed during this study. Mainly, four different datasets were developed, **JUDVLP-WBUAFS**: Fishdb-IMC.v1 for IMC recognition, **JUDVLP-WBUAFS**: Fishdb-EC.v1 for EC recognition, **JUDVLP-WBUAFS**: Fishdb-Detection.v1 for fish species recognition in cluttered environments, and **JUDVLP-WBUAFS**: Fishdb-Segmentation.v1 for fish species recognition using semantic segmentation in some cluttered environments. The dataset splitting, and the augmentation process (if any) are discussed in the specific chapter where the specific work is done. The developed datasets are a key takeaway from this research and will significantly contribute to future aquaculture research using machine learning. There is wide scope for updating these datasets by introducing more fish species and also collecting more images per fish species. Segmentation of the full visible part of the fish body can also be an area for future contribution.

Chapter 3

Fish Species Recognition in Non-cluttered Environment

THIS chapter provides an explanation of deep learning techniques for the automatic classification of various Indian carps, which can be beneficial for aquaculture monitoring and management, biodiversity research, aquatic ecosystems, education, and research. Automatically recognizing different fish greatly benefits the purchaser to identify fish without having any knowledge of fish morphology. Automatic recognition of fish species is mainly dealt with in two phases: recognition in non-cluttered, and cluttered conditions. This chapter explains the automatic recognition of Indian major carp and exotic carp in non-cluttered conditions using different machine learning techniques.

3.1 Introduction

Identifying fish species is one of the main tasks in many areas, like managing aquatic ecosystems, fisheries management, aquaculture monitoring, biodiversity research, aquarium management, and many more. Accurately identifying the fish species requires a thorough understanding of fish taxonomy and morphological features. To perform this task in the fishery industry, trained personnel are required. Due to its monotonous nature and high error rate, this process consumes a significant amount of time. The availability of trained personnel in fish taxonomy is a key issue in the fishery industry, prompting the need to investigate whether machine learning can automatically identify different fish species. The fish species are generally found in two types of environments, non-cluttered environments and cluttered environments. In the non-cluttered environment, fish do not overlap with each other, and each body part of the fish is visible in the image. In the cluttered environment, overlapped fish images typically appears. This chapter explains the automatic recognition of different carps using machine learning techniques. In India, carp are classified into two categories, Indian Major Carp (IMC), and Exotic Carp (EC). An automatic technique for recognizing these two types of carp is developed and tested on unseen samples in real-time cases.

3.2 Indian Major Carp Classification in Non-cluttered Environment

The fishery industry heavily depends on automated fish species identification for its socio-economic prosperity. Identifying major carp fish species based on their physical characteristics can be challenging due to their similar shape and size. The proposed autoencoder network models have been applied to a fish dataset containing 1500 images of three popular big carp species of India to recognise them automatically. The latent representation of autoencoder models has been used as feature. Once the training phase is finished, the decoder is eliminated, and fish species are classified using several classifiers. The Simple autoencoder (SAE), the Deep autoencoder (DAE), and the Deep convolutional autoencoder (DCAE) were applied with varying epochs and learning rates. With a learning rate of 0.0001, a promising accuracy rate of 97.33% was achieved in 250 epochs. The autoencoder models' performance was compared with Hu moments, Haralick texture, Weber local descriptor, HOG descriptor, InceptionV3, InceptionResNet, MobileNet, VGG16 and VGG19. By 52%, 43.55%, 13.77%, 6.67%, 22.22%, 15.11%, 6.66%, 4.89%, and 9.78% the deep convolutional autoencoder beat Hu moments, Haralick texture, Weber local descriptor, HOG, InceptionV3, InceptionResNet, MobileNet, VGG16 and VGG19. Several well-known machine learning algorithms, including Logistic Regression, Naive Bayes, K-Nearest Neighbor, Support Vector Machine, and Random Forest, were employed to assess the efficacy of the latent representation as a feature vector. The latent representation of the deep convolutional autoencoder, based on Support Vector Machine, exhibited superior performance compared to all other methods. It demonstrates the efficacy of the proposed technique in recognizing these carp species in non-cluttered environments.

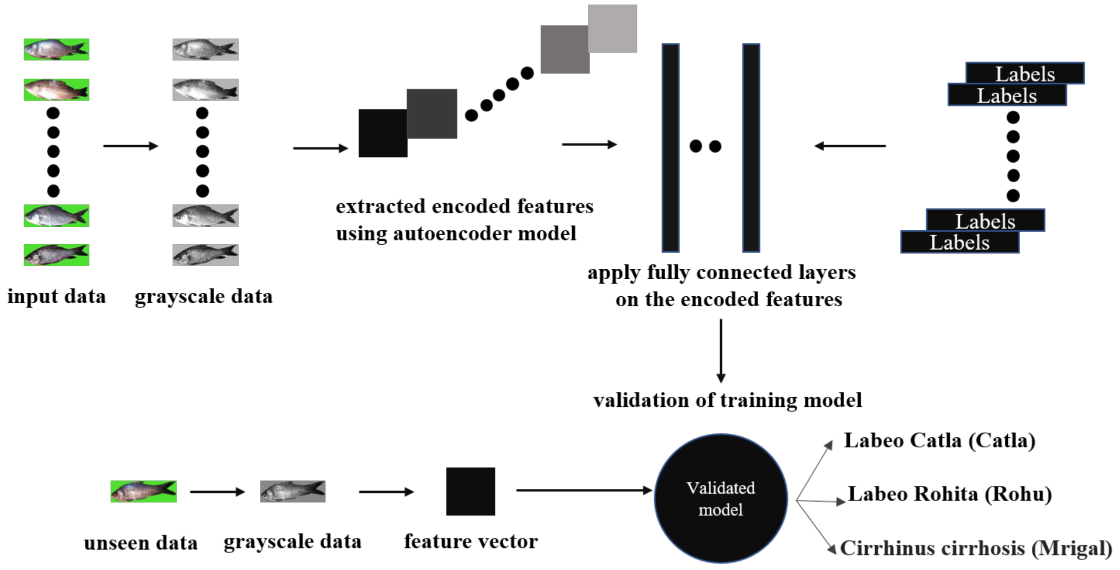


Figure 3.1: Workflow of the proposed approach for automatic fish species recognition (Banerjee et al. [1])

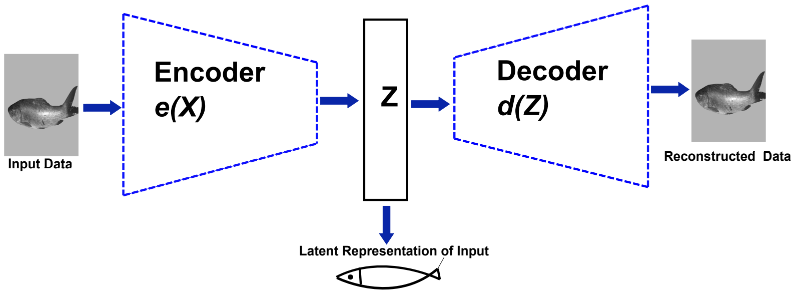
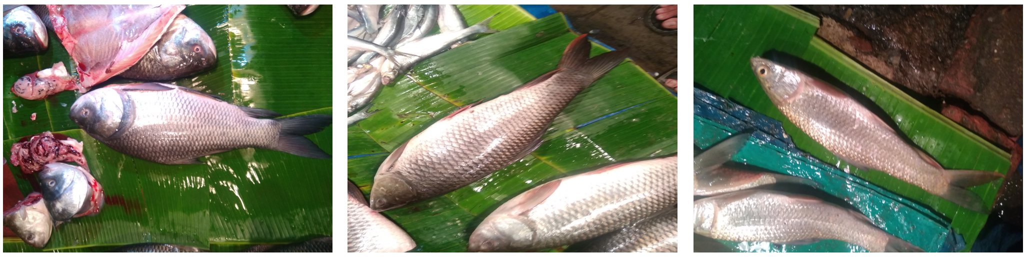


Figure 3.2: Autoencoder based feature extraction from a data sample (Banerjee et al. [1])

Around the world, a total of 34,700 fish species have already been recognized Froese and Pauly. [38] at present, while many more fish species are yet to be identified and recognized. These fish species have a wide range of habitats. India is bestowed with a total of 940 freshwater fish species distributed in rivers, ponds, and lakes. It is also the second-largest in fish-producing, as well as in aquaculture. Indian fisheries and aquaculture play an important role in food and nutrition supply, maintaining livelihoods and strengthening the economy. Identification of species is the most fundamental step not only for all biological research works but also for industries, marketing, and trading, and from the consumer's point of view. Fish show plenty of variations in their anatomical, morphological, molecular, and genetic characteristics, making them difficult to identify. The proficient in the fishery field who possess fish taxonomy knowledge mainly use classical methods or advanced taxonomic tools to identify the species. However, for those who have limited knowledge about the fishery, particularly consumers, students, and the public, it is imperative to follow identification by the fish's visual differences in-between different kinds of fish available. Due to the scarcity of such personnel, the manual process is time-consuming and costly. Therefore, finding a cost-effective and rapid method is the need of the hour. The automated image recognition system is one of the most promising and growing techniques to use artificial intelligence to identify fish, which has quick response time, high accuracy, and adaptability among all sections of people. A fish dataset was developed containing 1500 images of freshwater fish taken from several markets in West Bengal. The images of the fish were collected at various times and under various lighting conditions. Three Indian big carps are adopted for the study: i.e., *Labeo catla* (Catla), *Labeo rohita* (Rohu), and *Cirrhinus mrigala* (Mrigal). Due to their rapid growth rate, these species were chosen as the mainstay of freshwater aquaculture in India. They also have higher acceptability to consumers and are the most widely consumed species across India, and Bangladesh.

Identification of fish species has become an important research choice for its diversified applications in the fishery industry. In 2003, a fish species classification and migration monitoring system was proposed by Lee et al. [39]. The contour of the fish species was used to get the shape of the species. In their study, the images of the species were taken from

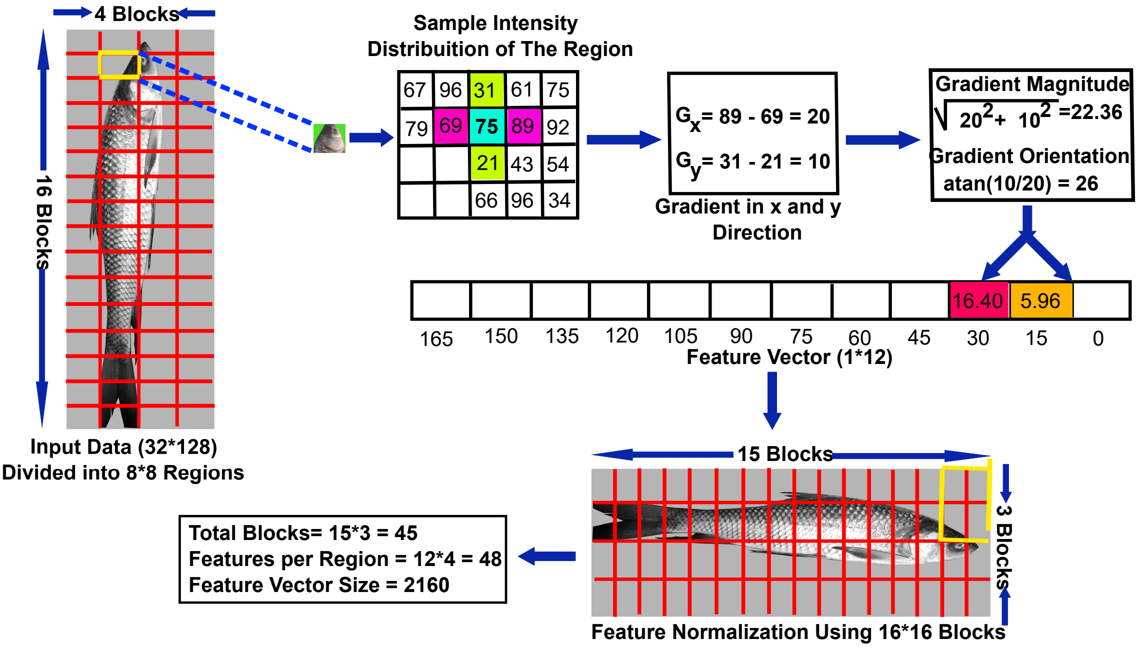


(a) Labeo Catla (Catla) (b) Labeo Rohita (Rohu) (c) Cirrhinus Cirrhosus (Mrigal)

Figure 3.3: Some samples of **JUDVLP-WBUAFS**: Fishdb-IMC.v1 Dataset (Banerjee et al. [1])

videotape and then digitized. A fish species classification using shape and texture was proposed in 2009 by Larsen et al. [10]. A total of 108 fish images of three fish species: cod, haddock, and whiting were used. There was an unequal distribution of data present per class. Another fish recognition system based on robust feature extraction from the size and shape of fish species was proposed in 2010 by Alsmadi et al. [11]. Several geometrical and distance features were extracted from the pre-calculated anchor points, and an Artificial Neural Network (ANN) was used in this study for classification purposes. A total of 500 images from 20 fish families were used in their study. In 2012, a method based on color and texture features was proposed by Hu et al. [12] for fish species classification. The fish skin texture was used for the experiment (not the full body of fish species), and different statistical texture features and wavelet features were extracted from that part. A multiscale Support Vector Machine(SVM) was used for classification purposes. A total of six common freshwater species in China were used in their work. The used species were Grass Carp *(Ctenopharyngodon idellus)*, Silver Carp *(Hypophthalmichthys molitrix)*, Bighead Carp *(Aristichthys nobilis)*, Snakehead Murrel *(Channa striata)*, Wuchang Bream *(Megalobrama amblycephala)*, and Red-bellied Pacu *(Colossoma brachypomum)*. They have used 540 images of the above fish species. Work on the classification of Nile Tilapia fish based on the features extracted from SIFT and SURF was proposed in 2013 by Fouad et al. [13]. A total of 96 images of Tilapia fish and 55 images of non-Tilapia fish were used for the binary classification. Some geometric transformations were applied to 16 distinct Tilapia fish for the generation of the dataset. In 2014, the identification of four fish species: chub, crucian, bream fish, and carp, using image processing and statistical analysis was proposed by Li and Hong [14]. The data distribution and the details of the dataset were not reported in their work. Another work using the combination of geometric features and bag of visual words (BoVW), for fish species identification was proposed in 2016 by Saitoh et al. [15]. A total of 129 fish species were used (20 images per fish species), and the images were collected from the web under different photographic conditions and environments. A cloud-based mobile app for fish species recognition based on fish anatomy anchor points was proposed by Rossi et al. [16] in 2016. The authors have collected the dataset from Turin (Italy) fish market. A total of 339 fish images were used for the seven species- *Engraulis encrasicolus* (125), *Sardina pilchardu* (107), *Pagellus erythrinus* (20), *Scomber scombrus* (18), *Sparus aurata* (22), *Merluccius merluccius* (19), *Mullus surmuletus* (28). In 2018, Rachmatullah and Supriana [17] proposed a convolutional network-based low-resolution fish image classification. The authors have used data augmentation techniques using rotation transformation to properly distribute the data among the classes. Among the used convolutional network schemes, two convolutional network layer-based approaches with 32 batches achieved the best result on the Fish CLEF 2015 dataset(Joly et al. [18]) with 15 different fish species. In 2018, Tharwat et al. [19] proposed a biometric-based fish species identification scheme using Weber Local Descriptor (WLD) and color features. The authors used 241 fish images of four fish species: *Argyrosomus regius*, *Sardinella maderensis*, *Scomberomorus commerson*, and *Trachinotus*. A better result was achieved using the AdaBoost classifier compared to Naive-Bayes, K-NN, and MLP algorithms. A modified version of the deep learning network, AlexNet, with four convolutional layers and two fully connected layers, was proposed for fish species classification in 2019 by Hussain et al. [20]. The training was done on the QUT fish dataset (Anantharajah et al. [21]), and for the testing and validation, the LifeClef 2015 Fish dataset (Joly et al. [22]) was used. A total of six fish species from both datasets were considered for the experiment: Cirrhilabrus, Lethrinus, Thunnus, Epinephelus, Scomberoides, Lutjanus. The modified AlexNet has outperformed the original AlexNet and VGGNet. In 2019, Montalbo and Hernandez [23] proposed an optimization-based VGGNet for fish species classification. Total 530 images of the FishBase (Montalbo and Hernandez [23]) dataset were used in addition to augmentation-based images for the experiment. In the same year, a 32-layer CNN architecture based on the VGGNet network was proposed by Rauf et al. [24] to identify fish species. The authors have developed a dataset termed as Fish-Pak, which consists of 915 images of six fish species- *Ctenopharyngodon idella* (Grass carp), *Cyprinus carpio* (Common carp), *Cirrhinus mrigala* (Mori), *Labeo rohita* (Rohu), *Hypophthalmichthys molitrix* (Silver carp), *Catla catla* (Thala). These images were taken from three regions of the fish body: head region, body region, and scale. Out of the total 915 images, the authors have selected 438 images for their experiment. The numbers of images in the head region, body region, and scale region are 140, 124, and 174. A 32-layer deep CNN architecture has outperformed some popular deep networks like AlexNet, GoogleNet, ResNet50, Lenet-5, and some variations of VGGNet architecture. In automatic FSR, the preparation of a standard dataset comprising of the fish species is the foremost phase. It is apparent that collected fish species are purely geographically region-specific, as stated in the previous paragraph about the recent study in automatic fish identification techniques. The datasets in many studies are either completely skewed or extremely small. In this work, this issue is addressed by creating a fish dataset with 1500 photos of three common Indian carp species: *Labeo catla* (Catla), *Labeo rohita* (Rohu), and *Cirrhinus mrigala* (Mrigal). Different autoencoder models were employed in this study to identify these fish species using the latent representation as the feature set. The flattened and fully connected softmax layers have been used for the recognition of these species. Simple Autoencoder (SAE), Deep Autoencoder (DAE), and Deep Convolutional Autoencoder (DCAE) were used with different numbers of epochs and learning rates. The result of the autoencoder models is compared with some conventional feature extraction techniques: Hu Moments Mercimek et al. [40], Haralick texture Haralick et al. [41], Weber local descriptor(WLD) Chen et al. [42] and Histogram of oriented gradients (HOG) Dalal and Triggs [43] as well as some state-of-the-art deep learning techniques in image recognition field such as InceptionV3Szegedy et al. [44], InceptionResNetSzegedy et al. [45], MobileNetSandler et al. [46], VGG16Simonyan and Zisserman [47] and VGG19Simonyan and Zisserman [47]. The latent representations of different autoencoder models are also evaluated with other popular classifiers: Logistic Regression (LR), Naive Bayes (NB), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Random Forest (RF) classifier. Results using these classifiers are compared with the results of other feature extraction techniques.

The main contributions of this study are as follows-



67	96	31	61	75
79	69	75	89	92
		21	43	54
		66	96	34

Figure 3.4: HOG feature extraction from a data sample (Banerjee et al. [1])

- The latent representations of three autoencoder models- SAE, DAE, and DCAE are extracted and applied for the identification of fish species using a fully connected layer as the last layer of the model. A standard dataset containing 1500 images of three popular major carp fish species of India (500 images of each species) is used.

- To establish the effectiveness of the latent representation of the autoencoder models as a feature, results are compared with some popular machine learning algorithms like Logistic Regression, Naive-Bayes, K-Nearest Neighbor, Support Vector Machine, Random Forest and some other feature extractor.

- Experiments are done incrementally on the collected data to measure the efficiency of the proposed approach with the increase of the amount of data.

- The results of the proposed approach are compared with the result using some traditional image descriptors: Hu moments, Haralick texture, WLD, and HOG as well as some popular deep neural networks like InceptionV3, InceptionResNet, MobileNet, VGG16 and VGG19.

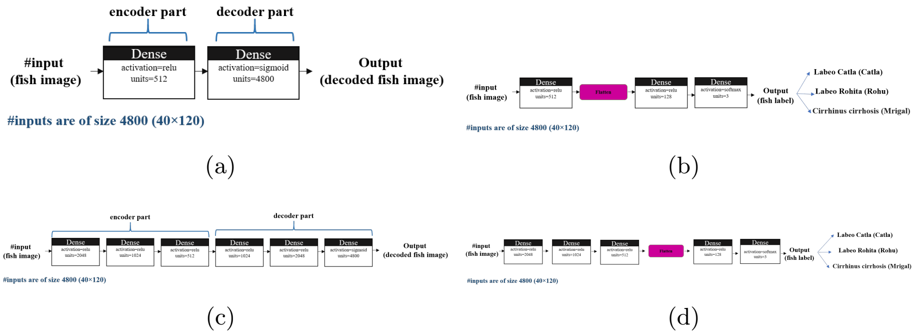
3.2.1 Dataset Preparation

In this proposed work, a total of 1500 fish images were collected from the Garia fish market and Mukundapur fish market situated in Kolkata, India. The prepared dataset contains the images of three major carp fish species of India- *Labeo catla* (Catla), *Labeo rohita* (Rohu), *Cirrhinus mrigala* (Mrigal). The name in the brackets is the local name of these fish species. A normal smartphone camera (Redmi 5A mobile phone, 16-megapixel camera) was used to capture the fish images from the market. Images were collected at different times (day/afternoon/night) under different illumination conditions, lightning conditions, rotation angles, and noisy conditions. A total of 500 images are present under each carp species. Some raw images from the dataset are shown in figure 3.3. When the images were collected from the fish market, other fish species were also present, creating a overlapping situation. As our aim in this work was to identify fish in non-cluttered environments, images with the presence of only a sinle fish were developed in the dataset. The 'inversion' tool of GIMP 2.10.8 software was used on the marked boundary of the targeted fish to divide the image into two regions: the targeted fish and the background. The green color was used in the background of each image. Each image was rotated to place the head of the fish on the left, its dorsal fin at the top, and its body in a horizontal position to make the dataset uniform. Details of the dataset collection and pre-processing is explained in section 2.1 of chapter 2.

3.2.2 Methodology

During the feature extraction phase, significant quantifiable measures were identified and collected together to make a robust feature vector. The feature vector is used to differentiate the three popular carp- Catla, Mrigal, and Rohu (local name). The developed fish dataset becomes challenging because of the variation in size (large, small, medium) of fish samples, different illumination conditions in the samples, the orientation of fish, the location of fish species in the image, etc. The samples are taken at different times and from different fish markets in West Bengal. Certain changes in the lighting condition, nature of the fish shop, and fish body parts suppressed by another fish species and, the presence of other fish species in the market act as catalysts in intra-class variation. These three fish species have similarity in shape and size, which makes the inter-class variation and the problem of recognizing the species, becomes a challenging task.

In this work, different effective autoencoder models were used to generate effective feature vectors from the latent representation of the models. The feature vectors were then passed to flattening and fully connected layers for classification. Also, the performance of fish recognition using Hu moments, Haralick texture, WLD and HOG was compared with the proposed autoencoder model. The feature generation and classification approach are pictorially represented in figure 3.1.



(a) (b)

(c) (d)

Figure 3.5: The structure of SAE, DAE and design of classifier using SAE and DAE for Indian major carp identification (a) Simple Autoencoder Network (SAE) for fish species identification, (b) Simple Autoencoder (SAE) based classifier for fish species identification (c) Deep Autoencoder network (DAE) for fish species identification (d) Deep Autoencoder (DAE) based classifier for fish species identification (Banerjee et al. [1])

Autoencoder

The autoencoder encodes the input data into a latent representation and then reconstructs it again with the minimum reconstruction loss. Basically, an autoencoder is an unsupervised way to represent data. It consists of two stages- encoder and decoder. If X is a data sample, the encoder part generates the reduced or latent representation Y. Then the decoder part regenerates the data sample, say $X^{'}$. The reconstruction error ϵ is involved in mapping $X->X^{'}$. The objective of the autoencoder is to reconstruct data with a minimum ϵ. The encoder is a function e that converts X into a latent representation Y. It is expressed as

$$Y = e(X) = S_e(WX + b_X) \tag{3.1}$$

here S_e is the activation function that may be linear or non-linear. The encoder part has two parameters: the weight matrix W and the bias vector $b_X \in R^n$. The decoder function d converts the latent representation Y back to a reconstruction $X^{'}$.

$$X^{'} = d(Y) = S_d(WY + b_Y) \tag{3.2}$$

here S_d is the activation function. During the training process of an autoencoder, the objective is to find parameters $\theta = (W, b_X, b_Y)$ that minimize the representation loss on the given data sample X. It is expressed as-

$$O = min_{\theta} L(X, X^{'}) = min_{\theta} L(X, d(e(X))) \tag{3.3}$$

In the linear activation function, the reconstruction loss L_{linear} is represented by-

$$L_{linear}(\theta) = \sum_{i=1}^{n} \left\| (x_i - x_i^{'})^2 \right\| = \sum_{i=1}^{n} \left\| (x_i - d(e(x_i)))^2 \right\| \tag{3.4}$$

If sigmoid activation is used, then the reconstruction loss $L_{sigmoid}$ is expressed as-

$$L_{sigmoid}(\theta) = -\sum_{i=1}^{n} [x_i log(y_i) + (1 - x_i) log(1 - y_i)] \tag{3.5}$$

where $x_i \in X$, $x_i^{'} \in X^{'}$ and $y_i \in Y$.
The fundamental operation of the autoencoder is presented in figure 3.2. In this work, an autoencoder has been used in a supervised way to identify the above three carp species. Initially, the autoencoder is trained with the encoder and decoder components for certain epochs, using both training and validation data. Next, the decoder portion is removed from the network, and fully connected network layers are employed. The entire network with a fully connected layer, after making the layers of the encoder part non-trainable, is retrained using the true labels of the data. The reduced latent representation will function as a robust feature map. Here, Simple Autoencoder (SAE), Deep Autoencoder (DAE) and Deep Convolutional Autoencoder (DCAE) were used. The entire structure of the models is described in subsection 3.2.4.

Hu Moments

In pattern analysis, the main concern is to recognize the objects irrespective of their size, orientation, and location in the image. Hu moment consists of seven measures calculated using central moments(Mercimek et al. [40]). Among the seven measures, the first six measures are invariant to scale, rotation, reflection, and translation. But the sign of the seventh measure changes in the case of reflection of an object. The $moment(i, j)$ of an image $f(x, y)$ of size $R \times C$ is defined as-

$$M_{ij} = \sum_{x=0}^{R-1} \sum_{y=0}^{C-1} x^i y^j I(x, y) \tag{3.6}$$

here i is the order of x and j is the order of y. The centroid$(\overline{x}, \overline{y})$ is calculated using-

$$\overline{x} = \frac{M_{10}}{M_{00}} \overline{y} = \frac{M_{01}}{M_{00}} \tag{3.7}$$

The central moment μ_{ij} is calculated by subtracting the centroid from x and y-

$$\mu_{ij} = \sum_{x=0}^{R-1} \sum_{y=0}^{C-1} (x - \overline{x})^i (y - \overline{y})^j I(x, y) \tag{3.8}$$

Central moment is invariant to translation transformation. When calculating the seven different measures of Hu moment, the normalized central moment was used, which is also scale invariant. The normalized central moment η_{ij} is calculated as-

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\frac{(i+j)}{2}+1}} \tag{3.9}$$

The Hu moments were calculated for each training and validation sample from the developed fish dataset. A validation dataset was used to find an appropriate model, and this model was used at the time of the testing phase. The feature vector size here is 1×7 if a single sample is being considered.

Haralick Texture

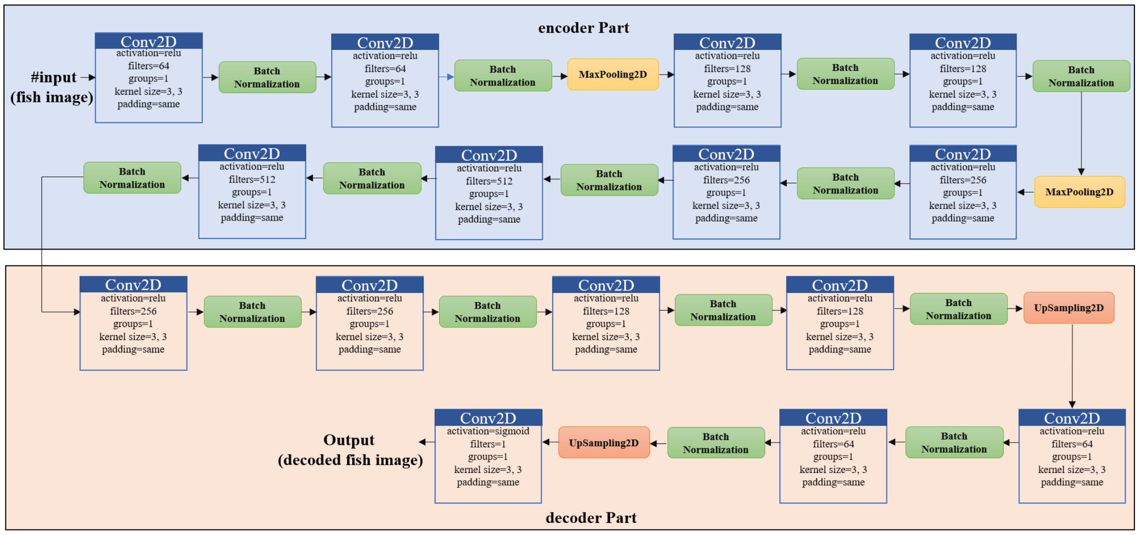
Texture is the repetitive pattern in an object in realization. When identifying different fish species, the texture of the fish body plays an important role. Here, Haralick texture feature is used to find the quantifiable texture feature. Mainly, the texture of the tail and head portion of the above carp fishes is distinguishable. But the tail of fish are mostly same in the action. The Haralick textures are derived from the Gray Level Co-occurence Matrix(GLCM) (Haralick et al. [41]). A total of fourteen (14) measures are present in the Haralick texture feature. In this work, only the first thirteen features were considered to form the texture feature vector. The feature vector size here is 1×13 if a single sample is being considered.

Histogram of Oriented Gradients (HOG)

HOG descriptor deals with the shape of the object in the image. It finds the edge direction from the detected edges in images (Dalal and Triggs [43]). The images are divided into several blocks, and then the gradient and orientation are calculated from each of those blocks. All the gradient magnitudes are grouped according to the orientation bin of the specific pixel. In this work, a total of 12 orientation bins are taken with an orientation range of 30°. Any intermediate bin is placed in both nearest orientation bins, with more weightage in the closest bin. Features of all the blocks are collected together to form the HOG feature vector of the sample. The images of the developed fish dataset are resized into 40×120 and divided into 8×8 regions. The HOG feature is extracted from all the 8×8 regions and then normalized using 16×16 blocks. The feature vector size here is 1×2160 if a single sample is being considered. The overall process is described in figure 3.4.

Weber Local descriptor (WLD)

The WLD (Chen et al. [42]) is based on the law proposed by Ernest Weber, draws a linear relationship between incremental threshold and background intensity. This relationship can be described by Weber's law, i.e.,$\delta I/I = k$, where $\delta I/I$ is called Weber's fraction (δI represents incremental threshold and I represents background intensity). WLD is formed using differential excitation (ξ) and orientation (θ).



#inputs are of size 40×120

Figure 3.6: Deep Convolutional Autoencoder (DCAE) structure for fish species identification Banerjee et al. [1]

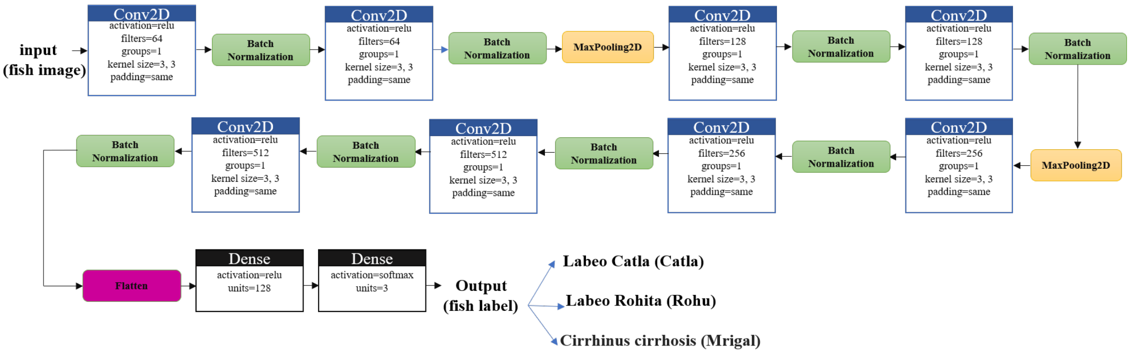


Figure 3.7: Deep Convolutional Autoencoder (DCAE) based classifier for fish species identification (Banerjee et al. [1])

a) Differential Excitation: The micro-variations within an image act as a major feature in identifying objects. It can be computed using the intensity difference between the neighbouring pixels. It can be computed as follows,

$$\Delta I = \sum_{i=0}^{p-1} \Delta I(x_i) = \sum_{i=0}^{p-1} I(x_i) - I(x_c), \tag{3.10}$$

where i^{th} neighbours of x_c is represented by $x_i (i = 0, 1, ...p - 1)$ and in a region, total of neighbours is p. $I(x_i)$ presents the intensity of the neighboured pixels, and $I(x_c)$ presents the intensity of the current pixel. The differential excitation is expressed as,

$$\xi(x_c) = arctan(\frac{\Delta I}{I}) = arctan(\sum_{i=0}^{p-1}(\frac{I(x_i) - I(x_c)}{I(x_i)})). \tag{3.11}$$

In the case of $\xi(x_c)$ is positive, the centre pixel is darker with respect to the neighbour pixels and in the case of $\xi(x_c)$ is negative, the current pixel is lighter with respect to the neighbour pixels.

b) Orientation: The directional pattern of the pixels is determined by this component. It can be expressed as:

$$\theta(x_c) = arctan(\frac{dI_h}{dI_v}), \tag{3.12}$$

where $dI_h = I(x_7) - I(x_3)$ and $dI_v = I(x_5) - I(x_1)$ is calculated from the intensities of the eight neighbor pixels of x_c. The mapping of $f : \theta \rightarrow \theta'$ can be expressed as, $\theta' = arctan2(dI_h, dI_v) + \pi$, and

$$f(x) = \begin{cases} \theta, & dI_h > 0 and dd I_v > 0 \\ \pi - \theta, & dI_h > 0 and dd I_v < 0 \\ \theta - \pi, & dI_h < 0 and dd I_v < 0 \\ -\theta, & dI_h < 0 and dd I_v > 0, \end{cases} \tag{3.13}$$

where θ varies from -90°. The quantization is done using,

$$\phi_t = f_q(\theta') = \frac{2t}{T}\pi, where\, t = mod(\left\lfloor \frac{\theta'}{\frac{2\pi}{T}} + \frac{1}{2} \right\rfloor, T). \tag{3.14}$$

The WLD histogram is formed using the above two components: differential excitation and orientation. This histogram can be used in identifying objects. A two-dimensional histogram is formed by $WLD(\xi_j, \phi_t)$, where (j from 0 to N-1) and (t varies from 0 to T-1). Here, N denotes the image dimension and T denotes the total number of dominant orientations. In this study, differential excitation is grouped into different numbers of bins: 2, 4, and 6. A total of 8 dominant orientations are used. The length of the WLD feature vector is computed as $N * T * S * m * n$, where S denotes scale (single/multiscale), m and n denote the size of the partition. If the image is divided into partitions, the WLD histogram is calculated for each partition and then concatenated to form the feature vector. Both single-scale and multiscale WLD are applied with single and multi partitioning.

3.2.3 Classifier

The autoencoder generates lower-dimensional features from the input data, and these valuable features are used in the classification phase. The latent features of the autoencoder were flattened using a flattening layer and passed to the fully connected neural network. In this work, two dense layers (fully connected layers) make the classification of the input data using the latent features of the autoencoder. As there are three fish species to be classified, in the final dense layer a softmax activation function is used. Also, the latent representation of the best model of the three autoencoder networks was extracted, and Logistic Regression (LR), Naive Bayes (NB), K-Nearest neighbour (KNN), Support Vector Machine (SVM), and Random Forest (RF) classifiers were used to compare the performance of the major carp species identification with Hu moments, Haralick Texture, WLD, and HOG.

Algorithm 1: Training the autoencoder model for the identification of three popular major carp fish species

```
Result: Trained model($Model_{train}$)
Input: working dataset($W_D$) ← extract fish dataset category wise.
1. The raw images in the working dataset are segmented using GIMP 2.10.8
   software, preprocessed, and then a region inside the minimum bounding box
   around the object is taken as the region of interest and saved into $W_D$. $W_D$ now
   contains 1500 images of three major carp fish species of size $40 \times 120$.
2. Divide the $W_D$ into training($W_{TR}$), validation($W_{VL}$) and testing($W_{TE}$)
   dataset using $70 : 15 : 15$ ratio.
3. Train the autoencoder model with the encoder($e(X)$) and decoder ($d(Z)$)
   using the $W_{TR}$ and $W_{VL}$ for different numbers of epochs and learning rate.
   Here, X represents input data and Z represents the latent representation.
4. Delete the decoder part of the autoencoder network and add fully connected
   layers for the classification task.
5. Make all layers of the encoder part non-trainable, and then train the fully
   connected layers only.
6. save the trained model as $Model_{train}$.
```

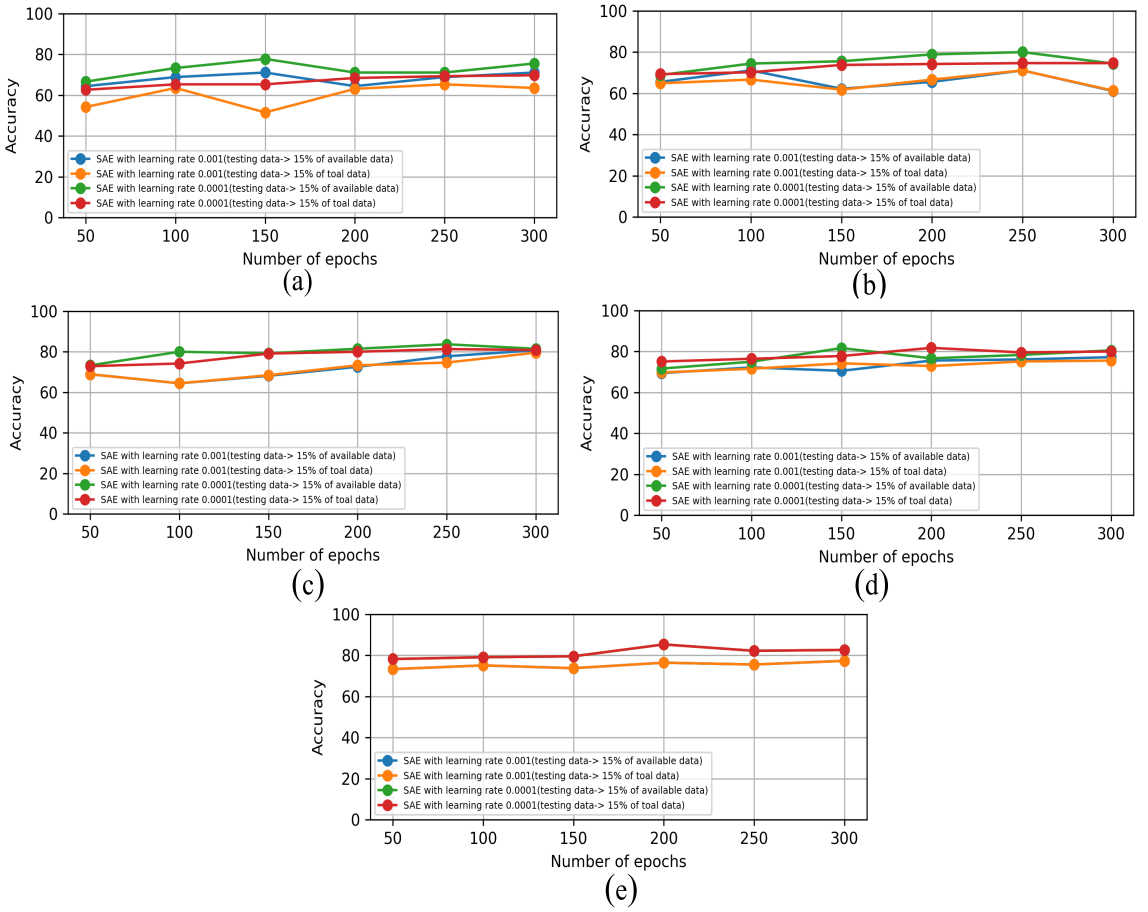
Algorithm 2: Testing the autoencoder model for the identification of three popular major carp fish species

```
Result: Class prediction of the unseen fish species($Prediction_{unseen}$)
Input: Test dataset($W_{TE}$), Trained model($Model_{train}$)
1. With $Model_{train}$, test the model using $W_{TE}$ .
2. The hot encoded result is processed and generates the class label of the unseen
   fish species.
```

3.2.4 Proposed Approach

Latent Representation Extraction of Autoencoder Models:

In this work, the latent representation of the autoencoder was used as the feature set for the classification of the fish species. Here, three different autoencoder networks were applied: SAE, DAE, and DCAE with experimental parameters and network structures. As more time is needed to train the deep convolutional autoencoder with large-size images, the images of the fish dataset were resized into smaller dimensions (for this proposed work, the dimensions of the images are 40 × 120). Generally, autoencoder is an unsupervised (or self-supervised) learning method. The stages of any autoencoder are input, encoder, bottleneck (latent), decoder, and output. The main task of the autoencoder is to generate the output from the compressed latent representation of the input with the minimum reconstruction error. The main steps of the proposed work are given in algorithm 1 and algorithm 2.



(a) (b) (c) (d) (e)

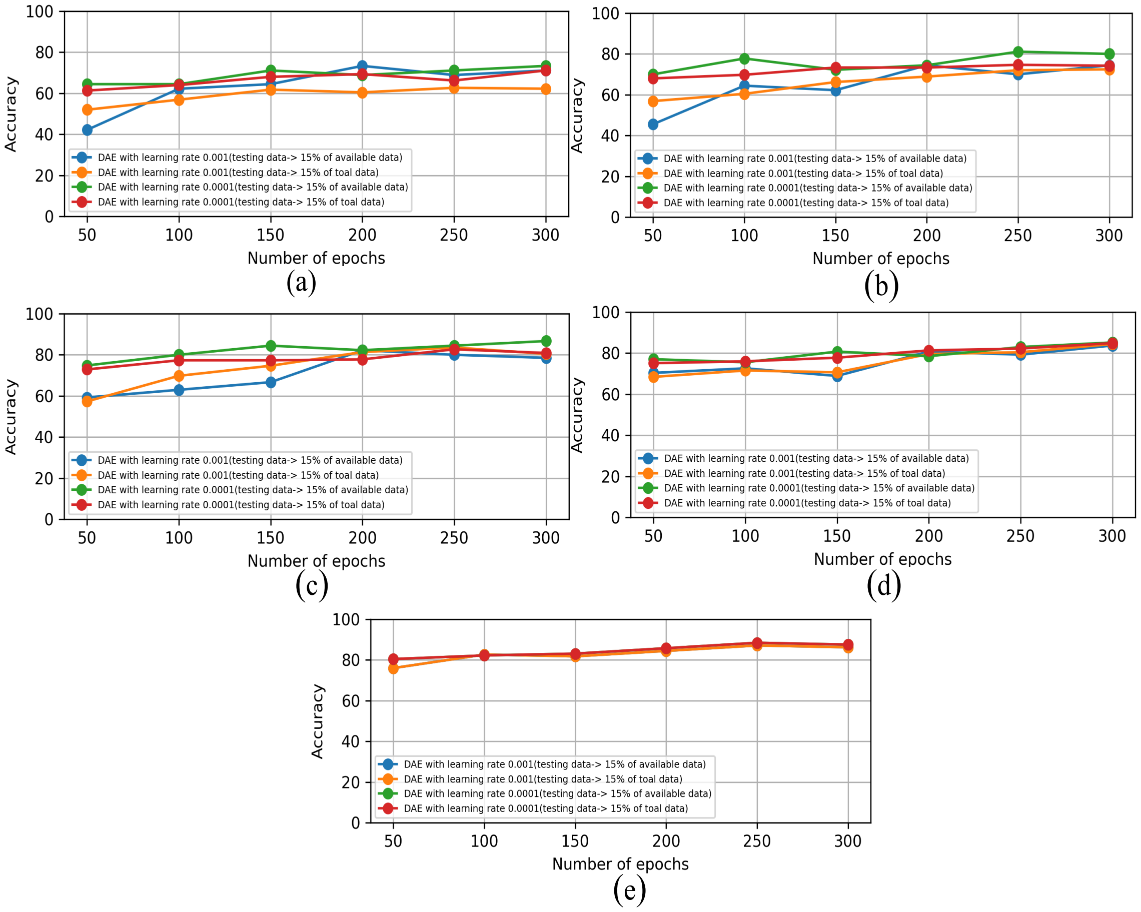
Figure 3.8: Number of epochs vs Accuracy graph for Simple Autoencoder Network (SAE)- (a) using 100 data per fish species (b) using 200 data per fish species (c) using 300 data per fish species (d) using 400 data per fish species (e) using 500 data per fish species

(a) Simple Autoencoder (SAE): In a simple autoencoder (SAE) network, the input layer provides input data from the given image, the encoder layer compresses the data, and the decoder layer decompresses the data. The output layer of the autoencoder provides the image with nearly the same as the input data. As per the SAE requirement, the image data (40×120) was resized to make a vector of size 4800 and fitted into the encoder part. In the encoder part, the encoding dimension of 512 was used, and the ReLU activation function was used. The decoder part has the exact dimension of 4800 as the image vector, and the sigmoid activation function is used. In figure 3.5(a), the structure of the simple autoencoder is depicted. Following the training phase, the decoder portion was removed, and some dense, flattened layers were employed to classify fish species. In this work, the latent representation of the data from the encoder part was flattened at first and then passed to the first dense layer of dimension 128. The ReLU activation function was used in the first dense layer. The output from the dense layer was then passed to another dense layer with a dimension of 3, just because there are three fish species to be identified by the system. A softmax activation function was used to deal with the multi-class classification problem. The identification part is pictorially represented in figure 3.5(b).

(b) Deep Autoencoder (DAE): A deep autoencoder network (DAE) consists of multiple hidden layers between the input and output layers of the network. In the encoder and decoder parts, the total number of layers is the same. The number of hidden layers is used as one of the functional parameters in the design of the autoencoder structure. In this work, three layers are used in the encoder and decoder parts. Like the simple autoencoder, the input image data vector has a dimension of 4800, passed to the encoder section. In the encoder part, a total of 3 layers were present, with dimensions 2048, 1024, and 512. The ReLU activation was used in all the layers. Then, the latent data was again reconstructed back to the original image using three dense layers with dimensions 1024, 2048, and 4800. The ReLU activation function was used in the first two layers, and the sigmoid activation function was used in the last layer. In figure 3.5(c), the structure of the DAE is depicted.

After completing this network's training, the decoder part was omitted, and the latent data vector was flattened. Then, the flattened data was passed to a dense layer with a dimension of 128 and a ReLU activation function. Next, a dropout layer was used to drop out 20% nodes. The output from this layer was passed to the last dense layer with dimension 3. In the last layer, a softmax activation function was used. In figure 3.5(d)., the identification based on DAE is represented.

(c) Deep Convolutional Autoencoder (DCAE): In the Deep Convolutional Autoencoder network (DCAE), convolutional layers are used in the encoder part of the network, and the deconvolution layers are used in the decoder part of the network. Each convolutional layer contains several filters, 3×3 kernels, and a ReLU activation function. The max pooling layer was applied in the encoder part to carry forward the most important information from the window. A total of eight layers were present in the encoder part with 64, 128, 256, and 512 filters (3.6). There were 512 variables in the latent representation of DCAE. The decoder part of the network does the deconvolution process and produces the reconstructed output. After each layer, batch normalization was applied with the previous layer's output to standardize the input to the next layer using the regularization factor and hence increase the training speed. In the training phase, more informative features are extracted from the input data. Like the previous two networks: SAE and DAE, the decoder part was omitted after the initial training process was over. After that, the latent representation was flattened and passed to dense layers, just like the previous two networks. A dropout layer was used here to drop out 15% of nodes. The final output gives the proper class label of the specific input data. In figure 3.7, the identification scheme based on DCAE is represented.

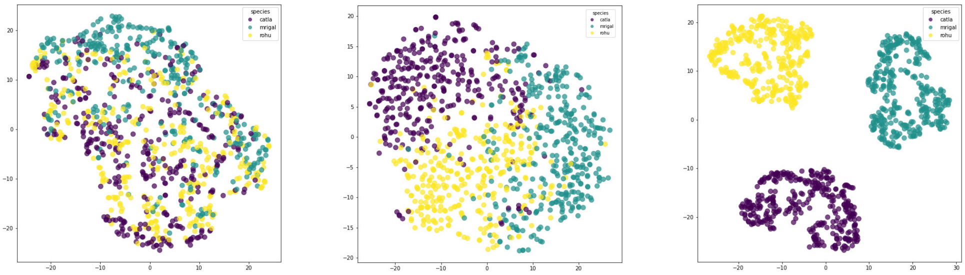


(a) (b) (c) (d) (e)

Figure 3.9: Number of epochs vs. Accuracy graph for Deep Autoencoder Network (DAE)- (a) using 100 data per fish species (b) using 200 data per fish species (c) using 300 data per fish species (d) using 400 data per fish species using 500 data per fish species (Banerjee et al. [1])

3.2.5 Experiment Protocol and Result Analysis:

The above three autoencoder networks wre applied on the **JUDVLP-WBUAFS**: Fishdb-IMC.v1 dataset comprising a total of 1500 images, where 500 images are present in each fish species: *Labeo catla* (Catla), *Labeo rohita* (Rohu), and *Cirrhinus mrigala* (Mrigal). As per the need of the experiment, the dataset was divided into training, validation, and testing datasets in a 70 : 15 : 15 ratio. Broadly, two types of experiments have been done using an autoencoder-based approach. Due to the heavy rush in the local fish market, more time was spent collecting the images and pre-process them. Based on



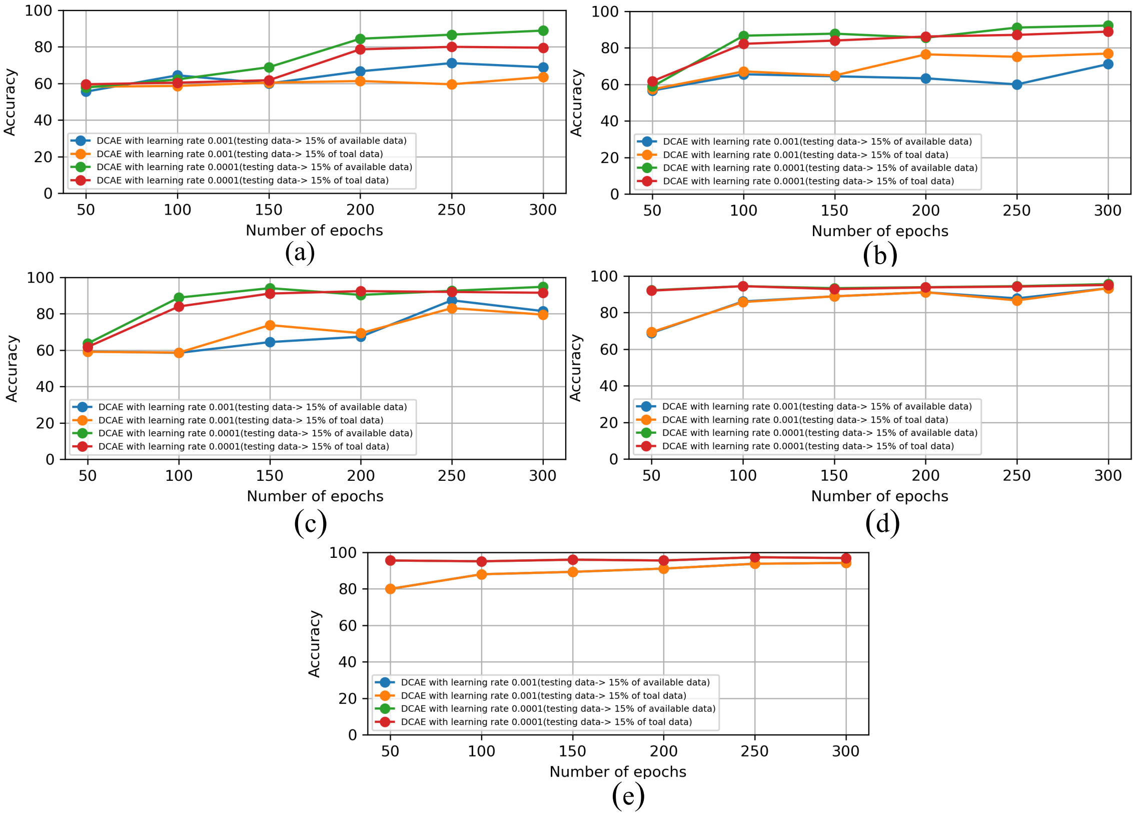
(a) (b) (c)

Figure 3.10: TSNE data distribution of fish dataset-(a) Latent features of SAE using training dataset. (b) Latent features of DAE using training dataset. (c) Latent features of DCAE using training dataset (Banerjee et al. [1])

this experience, the experiments are done using two approaches. In the first approach (termed as protocol-1), after collecting 100 images of each of the three fish species, the experiment was done, and different metrics were recorded. So, the same experiment was done five times, using 100, 200,300, 400, and 500 data samples per fish species. For example, with 100 data per fish species, training, validation, and testing data are 70, 15, and 15, respectively.

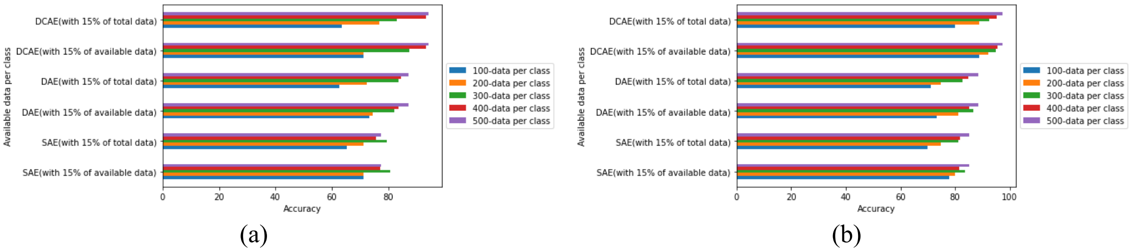
In the second approach (termed protocol-2), the experiment was done on the entire dataset of 500 images per fish species. To compare the performance of the two approaches, the same experimental configurations like protocol-1 were applied but with a different number of testing data. In protocol-2, 15% of the total collected data was used as the testing data, i.e., 225 fish images (75 images from each fish species). There are many functional parameters present in the autoencoder that may be tuned to get a detailed analysis of the network performance on the specific dataset, and also to get the optimal parameter settings that lead to better performance. In this work, epochs and learning rates were used as the functional parameters. The $50-300$ epoch range was used to maintain a gap of 50 and a learning rate of 0.01 and 0.0001 for the experiment. The Adam optimizer and Mean Squared Error (MSE) loss function was used in all the experiments.

SAE was applied on the fish dataset for $50, 100, 150, 200, 250,$ and 300 epochs. As per the experiment protocol stated above, two types of experiments were done. When protocol-1 was applied with 100 data per fish species, the best accuracy of 77.78% was achieved using the SAE with a learning rate of 0.0001 after running for 150 epochs. Using protocol-2, SAE with a learning rate of 0.0001 achieved 65.33% accuracy after running for 150 epochs. Using protocol-1, with the 200 data per fish species, accuracy using the SAE with a learning rate of 0.0001 is 80% after running for 250 epochs. But using protocol-2, SAE achieved 74.66% accuracy with a learning rate of 0.0001 after running for 200 epochs. Accuracy of 83.7% (using protocol-1) and 81.33% (using protocol-2) was achieved with 300 data per fish species using the SAE. The performance was achieved using a learning



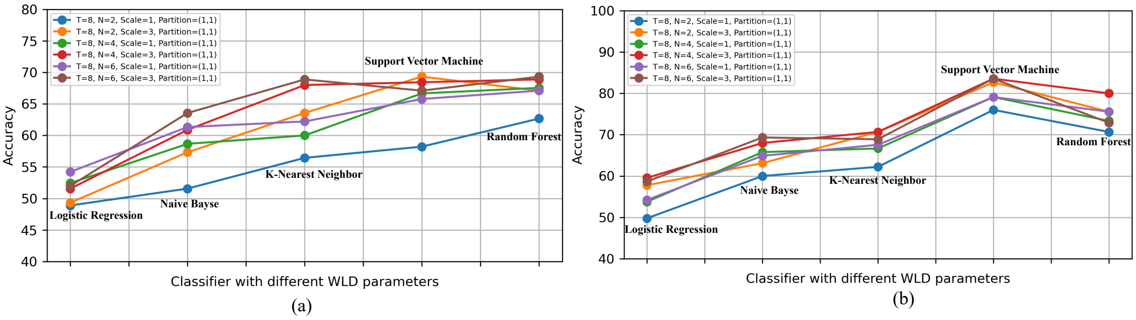
(a) (b) (c) (d) (e)

Figure 3.11: The number of epochs vs. Accuracy graph for Deep Convolutional Autoencoder Network (DCAE)- (a) using 100 data per fish species (b) using 200 data per fish species (c) using 300 data per fish species (d) using 400 data per fish species using 500 data per fish species (Banerjee et al. [1])



(a) (b)

Figure 3.12: Variation of performance of three autoencoder network with the increase of the amount of data using (a) learning rate 0.001 and (b) learning rate 0.0001.



(a) (b)

Figure 3.13: (a) Performance of single-scale WLD with different parameter settings (b) Performance of multi-scale WLD with different parameter settings (Banerjee et al. [1])

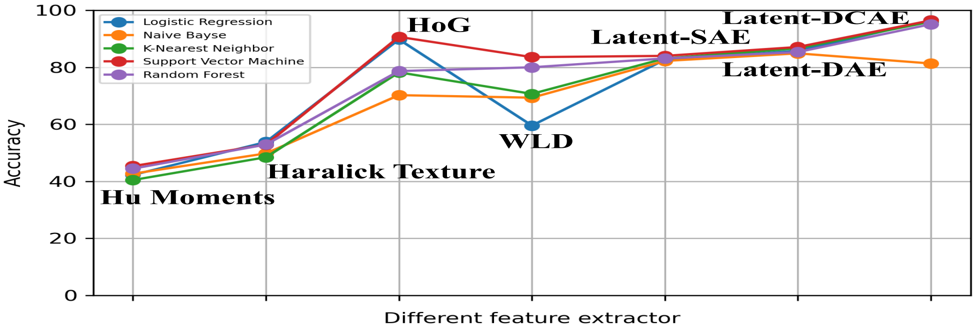


Figure 3.14: Performance comparison of Hu-moments, Haralick Texture, WLD, HOG and best performing SAE, DAE, DCAE model using different machine learning algorithms (Banerjee et al. [1])

rate 0.0001 after running for 250 epochs. The SAE achieved 81.66% (using protocol-1) and 81.77% (using protocol-2) accuracy using 400 data per fish species after running for 150 and 200 epochs. Again, the best performance was seen for the learning rate of 0.0001. It is observed that the accuracy of the network using protocol-2 was becoming better with the increase of data per fish species. The best accuracy of 85.33% was seen using the SAE with 500 data per fish species after running for 200 epochs. Figure 3.8 presents the epochs vs. accuracy graph for the SAE using 100, 200, 300, 400, 500 data.

DAE(refer figure 3.5(c).and figure 3.5(d).) was applied to the **JUDVLP-WBUAFS**: Fishdb-IMC.v1 dataset and run for $50, 100, 150, 200, 250, and 300$ epochs. Using 100 data per fish species, by following protocol-1, DAE with learning rates 0.001 and 0.0001 achieved the best accuracy of 73.33% after running for 200 epochs and 300 epochs. Using the protocol-2, DAE gained 62.66% and 71.11% accuracy using learning rates 0.001 and 0.0001. When 200 data per fish species was used, with protocol-1, DAE with learning rates 0.001 and 0.0001 achieved 74.44% and 81.11% accuracy after running the network for 200 and 250 epochs. With the same amount of data, following protocol-2, DAE with learning rates 0.001 and 0.0001 achieved 72.44% and 74.66% accuracy after running the network for 300 and 250 epochs. The DAE network with a learning rate of 0.001 and 0.0001 achieved 82.22% and 86.67% accuracy using 300 data per fish species using protocol-1. Just like the previous two cases, following protocol-2, less accuracy was achieved compared to protocol-1. Accuracy of 83.55% and 82.66% was achieved using DAE with a learning rate of 0.001 and 0.0001 by following protocol-2. In the subsequent two cases, a more stable result was achieved using the same network. Using 400 data per fish species, following protocol-1 83.7% and 85.18% accuracy was achieved using DAE with a learning rate of 0.001 and 0.0001 after running the network for 300 epochs. Following protocol-2, DAE with a learning rate 0.001 and 0.0001 achieved 84.44% and 84.88% accuracy after running the network for 300 epochs. Analysing the performance of the DAE, it is clear that a more stable result is achieved when the number of data is increased. The best accuracy of 88.44% was achieved using a DAE with a learning rate of 0.0001 after running the network

for 250 epochs. Figure 3.9 depicts the epochs vs. accuracy graph for the DAE network using $100, 200, 300, 400, and 500$ data per fish species.
DCAE (refer figure 3.6. and figure 3.7.) was applied to the fish dataset for $50, 100, 150, 200, 250,$ and 300 epochs. Using 100 data per fish species, following protocol-1, DCAE with learning rate 0.001 and 0.0001 achieved 71.11% and 88.89% accuracy after running the network for 250 and 300 epochs. Following protocol-2, the DCAE achieved 63.55% and 80% accuracy after running the network for 300 and 250 epochs with the same settings. When 200 data per fish species was used, following the protocol-1, DCAE with learning rates 0.001 and 0.0001 achieved 71.11% and 92.22% accuracy after running the network for 300 epochs. Whereas, with the same settings following protocol-2, DCAE achieved 76.88% and 86.88% accuracy after running the network for 300 epochs. The result using 300 and 400 data per fish species was improved by a reasonable amount. Accuracy of 87.4% and 94.81% was achieved with protocol-1, using DCAE with a learning rate of 0.001 and 0.0001 after running the network for 250 and 300 epochs. Using the same settings, 83.11%, and 92.44% accuracy was achieved by following protocol-2, using the DCAE, after running for 250 and 200 epochs. As the number of data increases, the results are improved and get stable for both protocol-1 and protocol-2. With the 400 data per fish species, following protocol-1, 93.33% and 95.56% accuracy was achieved using DCAE with learning rate 0.001 and 0.0001, after running the network for 300 epochs. When protocol-2 is followed with the same settings, the DCAE achieved 93.33% and 95.11% accuracy after running the network for 300 epochs. The best result using the DCAE was achieved when 500 data per fish species was used. The best accuracy of 97.33% was achieved, with 500 data per fish species, using the DCAE with a learning rate of 0.0001 after running the network for 250 epochs. The proposed DCAE outperformed the SAE and DAE by 12% and 8.89%. Figure 3.11 depicts the epochs vs. accuracy graph for the DCAE using $100, 200, 300, 400, and 500$ data per fish species. Analyzing the results of all the networks, it becomes clear that with the increase of data per fish species, the accuracy of the network gets better over epochs. Significantly, the performance of DCAE was improved reasonably with the increase in the number of data. Also, the performance of the DAE was better (3.11%) than the performance of the SAE, and the DCAE performed 12% and 8.89% better than the SAE and DAE. In figure 3.12, the performance variation of the three autoencoder networks with the amount of data is presented.

Some other popular feature extractors, such as Hu moments, Haralick texture features, WLD, and HOG were used to identify major carp fish species on the same dataset. To compare the performance of the different techniques, some popular machine learning classification algorithms, LR, NB, KNN, SVM, and RF classifiers, were used in the study. Also, some popular deep neural networks such as InceptionV3, InceptionResNet, MobileNet, VGG16, and VGG19 were used and the results of these networks were compared with the autoencoder networks. Five-moment measures were calculated in Hu moments, and a feature of size 1×7 was obtained for each data sample. Using multiclass

Method(Classifier)	Recall			Precision			F1-Score			Accuracy
	Catla	Mrigal	Rohu	Catla	Mrigal	Rohu	Catla	Mrigal	Rohu	
Hu Moment(SVM)	0.455	0.485	0.425	0.467	0.440	0.453	0.461	0.462	0.439	45.33
Haralick Texture(LR)	0.544	0.597	0.488	0.573	0.493	0.547	0.558	0.540	0.516	53.78
WLD(SVM)	0.851	0.853	0.803	0.840	0.853	0.813	0.846	0.853	0.808	83.56
HOG(SVM)	0.908	0.957	0.861	0.920	0.893	0.907	0.914	0.924	0.883	90.66
InceptionV3	0.813	0.813	0.680	0.735	0.762	0.823	0.772	0.787	0.745	76.89
InceptionResNetV2	0.800	0.907	0.760	0.845	0.829	0.792	0.822	0.866	0.776	82.22
MobileNet	0.973	0.934	0.813	0.880	0.922	0.938	0.924	0.934	0.871	91.11
VGG16	0.933	0.960	0.893	0.946	0.935	905	0.940	0.947	0.899	92.89
VGG19	0.907	0.947	0.800	0.872	0.922	0.857	0.889	0.934	0.828	88.44
SAE(SVM)	0.827	0.851	0.842	0.827	.840	.853	.827	.846	.848	84
DAE(SVM)	.903	.887	.829	.867	.840	.907	.884	.863	.866	87.11
DCAE(SVM)	.961	.986	.947	.973	.960	.960	.967	.973	.954	96.44
SAE(FCN)	.849	.865	.846	.827	.853	.880	.838	.859	.863	85.33
DAE(FCN)	.917	.901	.841	.880	.853	.920	.898	.877	.879	88.44
DCAE(FCN)	.961	1	.960	.973	.987	.960	.967	.993	.960	97.33

Table 3.1: Statistical Measures- Recall, Precision, F1-Score, and Accuracy of the top performing models of different methods (Banerjee et al. [1])

SVM as a classifier, the best accuracy of 45.33% is achieved using Hu moment. As the three fish species are very similar, the accuracy using Hu moment is not good. In this study, the first thirteen Haralick texture features are considered, and a feature of size 1 × 13 is obtained for each data sample. The best accuracy of 53.78% was achieved using the Logistic Regression(LR) classifier.

Though the performance using the Harlick texture is better than using the Hu moments, as there is moderate variation in the body texture of these species, the accuracy

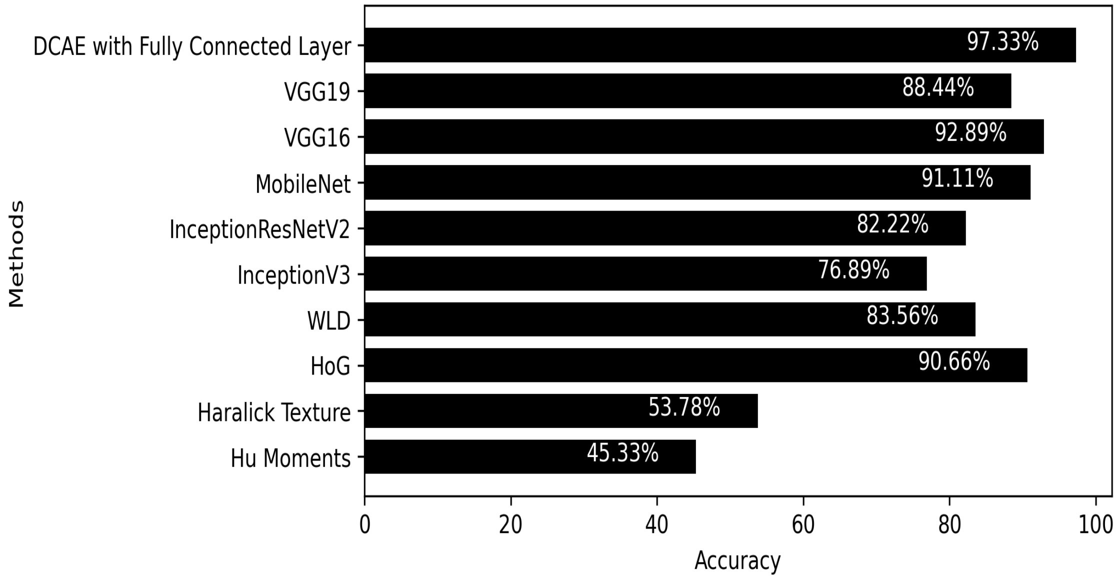
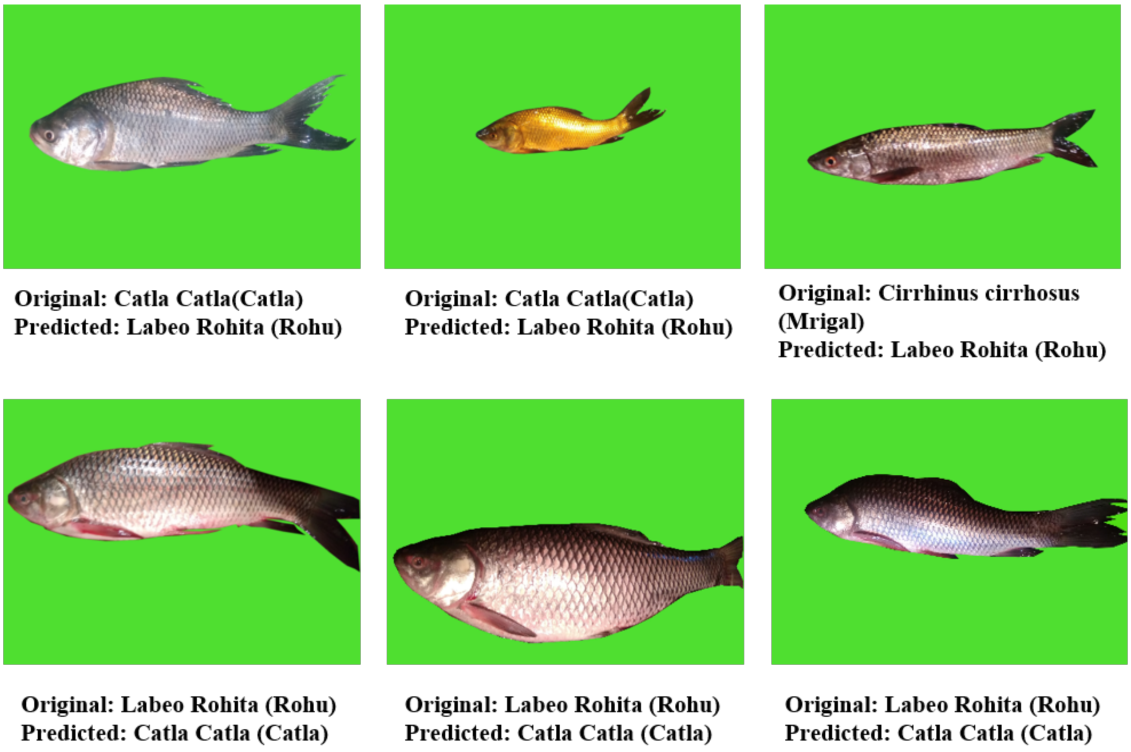


Figure 3.15: Comparative analysis of some conventional methods, some popular deep neural networks, and best performing deep convolutional autoencoder model (Banerjee et al. [1])



Original: Catla Catla(Catla)
Predicted: Labeo Rohita (Rohu)

Original: Catla Catla(Catla)
Predicted: Labeo Rohita (Rohu)

Original: Cirrhinus cirrhosus (Mrigal)
Predicted: Labeo Rohita (Rohu)

Original: Labeo Rohita (Rohu)
Predicted: Catla Catla (Catla)

Original: Labeo Rohita (Rohu)
Predicted: Catla Catla (Catla)

Original: Labeo Rohita (Rohu)
Predicted: Catla Catla (Catla)

Figure 3.16: The misclassified fish images in deep convolutional autoencoder network (Banerjee et al. [1])

is not promising due to the similarity in the scales of the fish species. WLD is applied with single and multi-partitioning using N=2, 4, and 6 and T=8. Both single and multi-scale WLD are applied. In figure 3.13 the performance of WLD with different parameter settings using single-scale and multi-scale is shown. When the HOG descriptor is applied

Method	Dataset Used	Classifier	Accuracy	Remarks
Lee et al. [39]	22 images of nine fish species Bonneville Cisco (BC), Brook trout (Bk), Brown trout (Bn), Chinook salmon (Ci), Coho salmon (CO), Cutthroat trout (Ct), Kamloops trout (Ks), Steelhead trout (Sd) and Yellowstone Cutthroat trout (Ye).	l2 norm was used to find the distance between features	Not reported	Very few images are present per fish species. The accuracy was not reported.
Larsen et al. [10]	Total 108 fish images: 20 cod (torsk), 58 haddock (kuller), and 30 whiting (hviling) caught in Kattegat. Standardized color CCD camera was used to capture images in white light illumination.	LDA	76%	Non-uniform dataset of 108 fish images are used. The constraint in dataset collection is that the images are taken in white light illumination.
Alsmadi et al. [11]	500 images from 20 fish families	Neural Network	86%	Different geometrical and distance features were extracted from the pre-calculated anchor points. The used dataset is very small.
Hu et al. [12]	540 images of six fish species: Grass carp (Ctenopharyngodon idellus), Silver carp (Hypophthalmichthys molitrix), bighead carp (Aristichthys nobilis), snakehead murrel (Channa striata), Wuchang bream (Megalobrama amblycephala), and red-bellied pacu (Colossoma brachypomum).	one-to-one based SVM	97.77%	Statistical and Wavelet features are extracted only from the fish skin texture (not the full body of fish). The dataset size is small here.

Table 3.2: Summary of the previous studies of automatic fish recognition with the dataset, classifier, and performance (Banerjee et al. [1])

with the normalization using 16 × 16 blocks, the best accuracy of 90.66% accuracy was achieved using the multiclass SVM. The experiment protocol and settings are described in 3.2.2. Among these techniques, the best performance was achieved using HOG and the next best performance was achieved using WLD. The performance of the HOG descriptor

Table 3.2: Summary of the previous studies of automatic fish recognition with the dataset, classifier, and performance (Banerjee et al. [1])

Method	Dataset Used	Classifier	Accuracy	Remarks
Fouad et al. [13]	96 images of Tilapia fish and 55 images of non-Tilapia fish	SVM	94%	A binary classification of Tilapia and non-Tilapia fish with very few images.
Saitoh et al. [15]	20 samples in each of 129 fish species in Japanese rivers and seas.	Random Forest	96.30%	Combination of geometric features and bag of visual words (BoVW) was used. The dataset was collected from Web. Very few samples per class.
Rossi et al. [16]	339 fish images were used of seven species- Engraulis encrasicolus (125), Sardina pilchardu (107), Pagellus erythrinus (20), Scomber scombrus (18), Sparus aurata (22), Merluccius merluccius (19), Mullus surmuletus (28).	ANN	Classification schemes based on cross-validation and in-field validation achieved encouraging results.	The exact accuracy is not reported. The number of images in the dataset is not adequate.
Rachmatullah and Supriana [17]	Fish CLEF 2015 datasetJoly et al. [18]- 15 fish species. Highly unbalanced dataset, data augmentation is applied to increase the number of images.	CNN	99.7%	Due to highly unbalanced dataset, rotation(-10 to +10) based data augmentation was used on, where number of images are less than 500. As the rotation angle is very small, augmented images have very little difference.

is promising if the challenges of the fish dataset are considered. Also, to draw a comparison with these methods and to establish the effectiveness of the autoencoder-based technique, the latent representation of the best-performing models of SAE, DAE, and DCAE is used. The latent representation acts as the feature vector and LR, NB, KNN, SVM, and

Table 3.2: Summary of the previous studies of automatic fish recognition with the dataset, classifier, and performance (Banerjee et al. [1])

Method	Dataset Used	Classifier	Accuracy	Remarks
Tharwat et al. [19]	241 images of four different fish species: Argyrosomus regius, Sardinella maderensis, Scomberomorus commerson, and Trachinotus. K-fold cross-validation was used.	ADABOOST	96.40%	Weber Local Descriptor and color momnets were used as feature. The dataset size is small here. The ADABOOST classifier performance is better than Naive Bayesian, k-Nearest Neighbor, and Multilayer Perceptron.
Hussain et al. [20]	QUT fish dataset for training, Fish CLEF 2015 dataset. Six fish species were taken for the experiment- Cirrhilabrus, Lethrinus, Thunnus, Epinephelus, Scomberoides, Lutjanus. Total 1334 images (training-866, validation-201, testing-267).	Modified AlexNet	90.48%	AlexNet model was applied on the QUT fish dataset at the time of training. The model is is validated and tested on FISH CLEF 2015 dataset. The species wise number of images are few and in some fish species, data augmentation was applied.
Montalbo and Hernandez [23]	530 images of three fish species- Amphiprion clarkii (AC), Chaetodon baronessa(ChB), Ctenochaetus binotatus(CtB). Training-455, Test-75	Optimization based VGGNet	99%	Optimization based VGGNet was applied on the 530 images of the three fish species. Data augmentation was applied to produce more images. The VGGNet model is applied here for only 10 epochs. Number of images after data augmentation is not reported here.

Table 3.2: Summary of the previous studies of automatic fish recognition with the dataset, classifier, and performance (Banerjee et al. [1])

Method	Dataset Used	Classifier	Accuracy	Remarks
Rauf et al. [24]	438 images of six fish species- Ctenopharyngodon idella (Grass carp), Cyprinus carpio (Common carp), Cirrhinus mrigala (Mori), Labeo rohita (Rohu), Hypophthalmichthys molitrix (Silver carp), Catla catla (Thala). Fish head region-140 images, fish body region-124 images, and fish scale region-174 images.	32-layer CNN architecture based VGGNet	Fish body-96.33% Fish head region- 92% Fish scale region-87.33%	Out of toal 438 images of six fish species, 348 images were used in training and 90 images were used in testing. A 32-layer CNN architecture based on VGGNet was applied on fish head region, body region and scale region. Paricularly, each of these region is having very few number of images. Again,seeing species wise, number of images per region (head, body, and scale) are very few.
Proposed approach	1500 images of three carp species from India: Labeo rohita (Rohu), and Labelo catla (Catla), Cirrhinus mrigala (Mrigal). In each fish species, 500 images are present.	Deep convolutional autoencoder(DCAE) based identification	97.33%	Various autoencoder models are used on the fish dataset with 1500 images consisting of three carp species. The images are taken in different environment and time. A wide variety is present in scale, rotation and intensity distribuition.

RF classifier algorithms were used. The latent representation of the best performing DCAE model achieved the best accuracy of 96.44% using multi-class SVM. In figure 3.14, comparisons of performance of different features using different machine learning algorithms are presented. In this study, the best-performing DCAE network achieved 97.33% accuracy with a fully connected layer, and a dropout of 0.15 and outperformed the Hu moments, Haralick texture, WLD and HOG descriptors by 52%, 43.55%,13.77% and 6.67%. Overall performance comparison of DCAE and other methods are shown in figure 3.15.

Traditional handcrafted feature extraction methods, such as Hu Moments, Haralick Textures, WLD, and HOG, often fail to capture the intricacies of complex, high-dimensional data. These methods rely on predefined heuristics, making them less adaptable to the subtle patterns necessary for fine-grained classification tasks.In contrast, deep convolutional autoencoders (DCAEs) offer a more robust solution. Unlike standard convolutional neural networks (CNNs) such as MobileNet, InceptionV3, or VGG16/19, which are not explicitly optimized for compact feature representation, DCAEs are inherently designed for efficient feature compression. By encoding input data into a compact latent space, autoencoders emphasize retaining only the most discriminative features, leading to improved class separability.

In this work, different statistical measures such as recall, precision, F1 Score, and accuracy are used in comparing performance of different methods. Precision measures how much the positive predictions are correct and recall measures how much the actual positives are predicted correctly. The precision for the species Catla, Mrigal, and Rohu are .973, .987, and .960 respectively. It means that out of the times Catla, Mrigal, and Rohu species are predicted, DCAE with FCN as classifer is 97.3%,98.7%, and 96% accurate. And out of all the times Catla, Mrigal, and Rohu should have been predicted, DCAE with FCN as classifier correctly predicted the species by 96.10%, 100%, and 96%. Also, the F1 score for Catla, Mrigal, and Rohu is 0.967, 0.993, and 0.96. The precision, recall, and F1 score for DCAE with FCN as a classifier is better than the other methods applied in the study. Here, no species are wrongly classified as Mrigal, which means for the Mrigal category, there are no false negatives. Three samples of Rohu species are wrongly identified as Catla, two samples of Catla species are identified as Rohu, and one sample of Mrigal species is classified as Rohu.

Comparisons with other studies: Different studies for fish species identification are described in section 3.2, with the methods and datasets. The research in this field has been done in different countries, and the datasets are from that specific region. As per my knowledge, only one datasetRauf et al. [24] of freshwater fish species (found in India) is publicly available. In 2019, Rauf et al. [24] developed Fish-Pak dataset, that consists of 438 images of six fish species from Asia. They have used 140 images of the fish head region, 124 images of fish body shape, and 174 images of the fish scale region. Due to the small dataset size and class imbalance, the comparison does not make any sense. Table 3.2 summarizes all previous studies (as per authors' knowledge) in automatic fish recognition and proposed technique. As the images are taken at different times of the day and in different lighting and illumination conditions, the dataset created becomes challenging. The results of latent representation of the autoencoder as a feature studied in this work is promising considering the dataset challenges.

3.2.6 Conclusion

This study proposed a method of using the latent representation of different autoencoders as a feature, for the recognition of three widely distributed carp species in India. Due to the lack of a standard fish dataset, a fish dataset containing 1500 images of three major carp species was created: *Labeo catla* (Catla), *Labeo rohita (Rohu)*, and *Cirrhinus mrigala* (Mrigal). The experiment methodology is established incrementally in the current work to achieve parity with the data collection procedure mentioned previously due to multi-phase data gathering.

The study demonstrates that as the amount of data increases, the performance of the various autoencoder networks improves. Experiments are conducted using two different learning rates: 0.001 and 0.0001, and majorly, the performance of the model with a learning rate of 0.0001 is better compared with a learning rate of 0.001. As demonstrated in the study, among the autoencoder models used, DCAE has outperformed the other autoencoder models and conventional feature extraction techniques in terms of accuracy. Despite being the best-performing autoencoder network, the DCAE misclassifies some fish images in the dataset during testing (refer figure 3.16). There could be numerous explanations for these misclassifications; one possibility is that the image already contains distortions. Other fish species may have merged with the target fish species, resulting in segmentation abnormalities. On the other hand, the high degree of resemblance among fish species may be another factor contributing to the misclassification. In light of these obstacles, the technique designed will be beneficial in the social and economic realms. Additionally, the authors believe that additional data on these carp fish species is necessary and that including additional fish species will make the problem more challenging and useful for the fishery industry and common people. In future, different deep learning networks could be designed to achieve better performance and model generalization.

3.3 Indian Exotic Carp Classification in Non-cluttered Environment

A system for automatic identification of three exotic carps, namely the Common carp (*Cyprinus carpio*), Silver carp (*Hypophthalmichthys molitrix*), and Grass carp (*Ctenopharyngodon idella*), has been developed using deep learning. The process involves four steps, namely, data collection, image preprocessing, image segmentation, and species identification. The images of the fish species were captured at different fish markets in West Bengal, under varying lighting situation. The proposed **JUDVLP-WBUAFS**:

Fishdb-EC.v1 dataset comprises 1225 images, which are divided into a 3:1:1 ratio for training, testing, and validation purposes. Following preprocessing and segmentation, several well-known pre-trained deep learning architectures were utilized to recognize these species using transfer learning. More precisely, VGG16, VGG19, InceptionV3, MobileNetV2, and InceptionResNetV2 were used for different epochs, with some task-specific fully connected layers. These networks were utilized with a learning rate of 0.00001. These architectures attained maximum accuracies of 99.18%, 98.36%, 98.36%, 99.18%, and 98.36% correspondingly. The findings demonstrate that the MobileNetV2 and VGG16 networks achieved a maximum accuracy of 99.18% and surpassed the performance of other deep learning networks.

A total of 34,800 fish species have been identified around the world (Froese and Pauly. [38]). Approximately 58% of them are marine, 41% are freshwater dwellers, and 1% are migrants. In the Indian subcontinent, there are about 2600 species, 1000 of which inhabit inland water and the rest in the seas. Local fishers, traders, and residents who live near the fishing areas may identify many species based on their expertise alone, but the identification may not be always appropriate. Non-professionals find proper enumeration difficult due to similarities in morphological characteristics, particularly within the generic level. With the developments in automatic identification techniques in recent years, various new ways have emerged to overcome such difficulties. India's freshwater aquaculture is carp-centric with Indian major carps viz., *Labeo catla*, *Labeo rohita* and *Cirrhinus mrigala* contributing lion's share. This carp-centric trend's lion share is contributed by the three Indian Major Carp (IMC) species. In this context, exotic carps or Chinese carps namely *Cyprinus carpio* (Common carp), *Hypophthalmichthys molitrix* (Silver carp), and *Ctenopharyngodon idella* (Grass carp) are the next most important category. Thus, diversification ensures greater production with proper utilization of the niches of the aquatic ecosystem. It is also worth noting that these three are among the top five major species produced in aquaculture around the world (Froese and Pauly [48]). Many people who do not have a taxonomic background, such as fishery inspectors, customs officers, data collectors, dealers and consumers, deal with complex situations and face challenges in fish species recognition. The appropriate identification of fish species necessitates the development of an easy-to-use identification tool. As a result, image recognition-based automated systems for fish species recognition are in high demand, and they are useful for the common people who do not possess knowledge of fish taxonomy. Automatic detection of different exotic carp is studied. A total of 1500 images of exotic carps were collected, and after the screening phase, a dataset consisting of 1225 images was developed. The main contribution here is the creation of such an unconstrained dataset and the application of some popular deep learning algorithms to effectively identify the three different exotic carps in a non-cluttered environment.

3.3.1 Related Works

Automatically recognizing fish species has become a subject of great interest due to its wide range of applications in the fishery industry. This challenge has been tackled in two ways during the last decade: standard machine learning and deep learning. The analysis of morphological traits is used to identify fish species. The skin texture, wavelet feature, color features, and other statistical features were employed as feature vectors, and these features were then used in various machine learning algorithms for fish species recognition. Hu. et. al. Hu et al. [12] suggested a method based on fish skin texture, wavelet characteristics, and various statistical features in 2012. A total of 540 images of six different fish species were used, with SVM as the classifier. In 2016, Rossi et al. Rossi et al. [16] proposed different geometric features based on morphological attributes from the user-selected anchor points. For the seven fish species, a total of 339 fish images were obtained from the Turin (Italy) fish market. This study depended entirely on the anchor points selected by the user because the morphological dimensions were prepared based on the anchor points. A combination of fish skin texture features using Weber Local Descriptor (WLD), and color features were used as the feature for the classification of 241 fish images of four fish species by Tharwat et al. [19].

Various deep learning algorithms, such as modified AlexNet, VGGNet, and others, have recently been used to identify fish species. In 2018, Rachmatullah and Supriana [17] proposed a convolutional neural network for the classification of 15 fish species. Because the dataset was highly imbalanced, rotation data augmentation was applied. A variant of AlexNet, consists of four convolutional layers and two fully connected layers, was proposed for the classification of six fish species in 2019 by Hussain et al. [20]. The QUT fish dataset by Anantharajah et al. [21] was utilized for training, and the LifeClef 2015 Fish dataset by Joly et al. [22] was used for testing and validation. The modified AlexNet has outperformed the AlexNet and VGGNet. Montalbo and Hernandez [23] proposed an optimized VGGNet and applied to 530 images collected from the FishBase dataset, and Rauf et al. designed and tested a 32-layer CNN architecture based on VGGNet on 438 images from the Fish-Pak dataset Rauf et al. [24]. The proposed CNN architecture was tested individually on the fish body, fish head, and the fish scale.

3.3.2 Dataset Preparation

Images of three exotic carps, Common carp (*Cyprinus carpio*), Silver carp (*Hypophthalmichthys molitrix*), and Grass carp (*Ctenopharyngodon idella*) were collected using 16-megapixel camera from different fish markets like Patipukur fish market (North 24-Parganas- West Bengal), Annapurna fish market (Purba Medinipur-West Bengal), and Chak Bazar (Bankura-West Bengal). A total of 1500 images were collected, but due to quality issues several images were removed, and a dataset consisting of 1225 images was developed. The images were segmented into two parts: the fish body part, and the background part, which is set to white color to make a homogeneous dataset. The dataset is divided into training, validation, and testing datasets with 735, 245, and 245 images. The dataset is named as **JUDVLP-WBUAFS**: Fishdb-EC.v1. The details of the image collection and data preparation are explained in the section 2.2 of chapter 2.

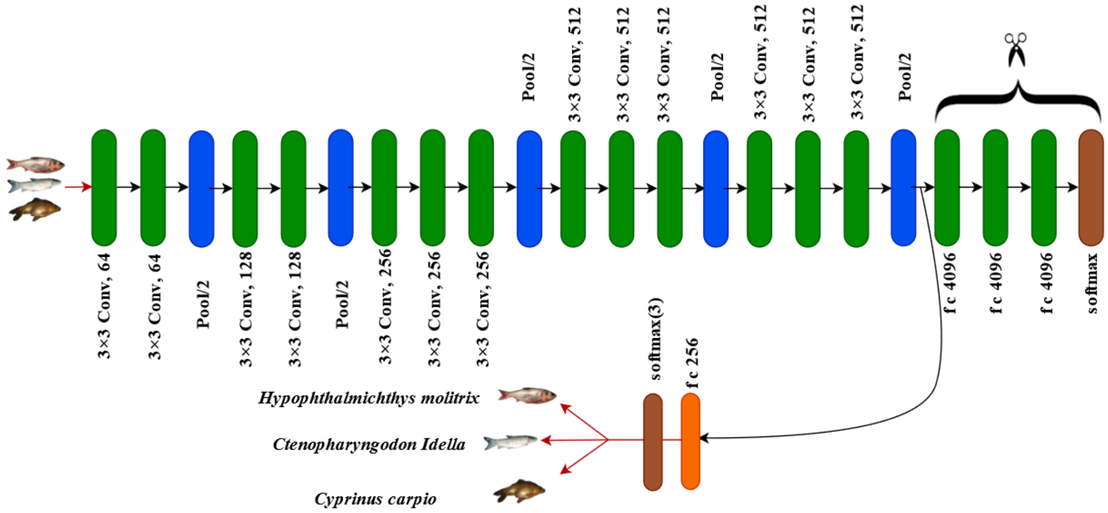


Figure 3.17: Architecture of VGG16 with transfer learning approach (Banerjee et al. [2])

3.3.3 Methodology

Some popular deep learning techniques were used in this work to automatically recognize the three major exotic carp. Transfer learning became a popular choice because deep learning techniques are computationally expensive, and it is also effective for moderately small size datasets. It is the process of applying a pre-trained network on a large dataset to a new task. In this study, the transfer learning mechanism is used with some pre-trained networks like VGG16, VGG19, In-ceptinoV3, MobileNetV2, and InceptionResNetV2. This section explains the architectural details of these deep networks.

VGG16 (Li et al. [49])

It has 16 convolution layers with a 3×3 kernel size (Figure 3.17). The number of feature mappings or convolutions rises as the network's depth increases. There are 138 million parameters in the network. The input dimension of the architecture is used as (224×224). In the pre-processing stage, the RGB image's mean value is subtracted from each pixel. In the convolution layers of VGG19, there are 64 feature maps in the 1^{st}, and 2^{nd} layer, 128 feature maps in the 3^{rd}, and 4^{th} layer, 256 feature maps in the 5^{th}, 6^{th}, and 7^{th} layer, and 512 feature maps in the 8^{th} -13^{th} layer. Three fully connected layers with 4096 units are used and a softmax output layer of 1000 units is used at the end to classify 1000 different objects of ImageNet.

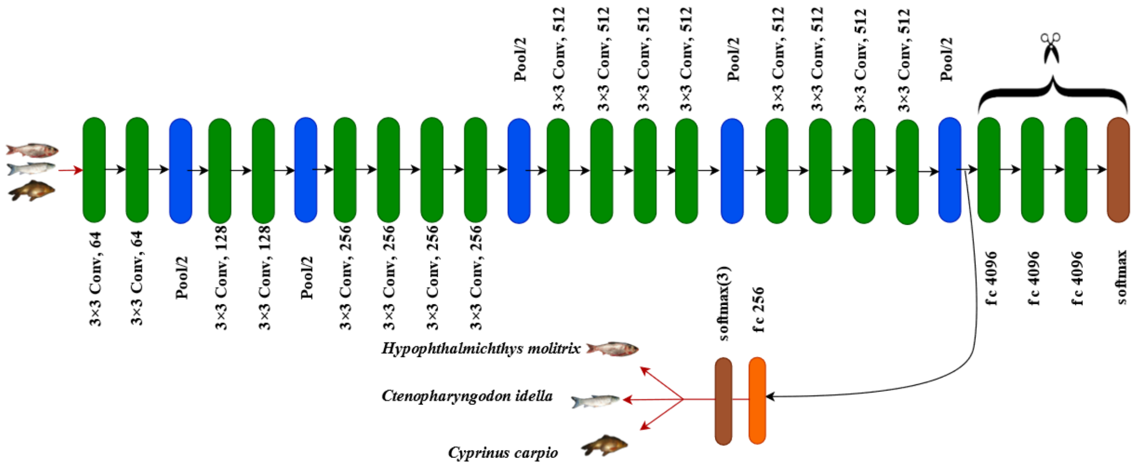


Figure 3.18: Architecture of VGG19 with transfer learning approach (Banerjee et al. [2])

VGG19 (Li et al. [49])

Multiple ways exist for doing transfer learning, wherein the same model is utilized to extract feature representations from new images. As the proposed dataset is not large, a pre-trained deep neural model is used as a feature extractor. The VGG-19 model is known to attain high accuracies for large datasets such as ImageNet (Figure 3.18). The VGG-19 model consists of around 143 million parameters, which are trained using the ImageNet dataset, which has 1.2 million images of various objects belonging to 1,000 distinct categories. A total of 19 trainable layers are present in VGG19 including convolutional, max pooling, and dropout layers. In this study, pre-trained convolution base was used with a customized classification part including densely connected layer and softmax layer.

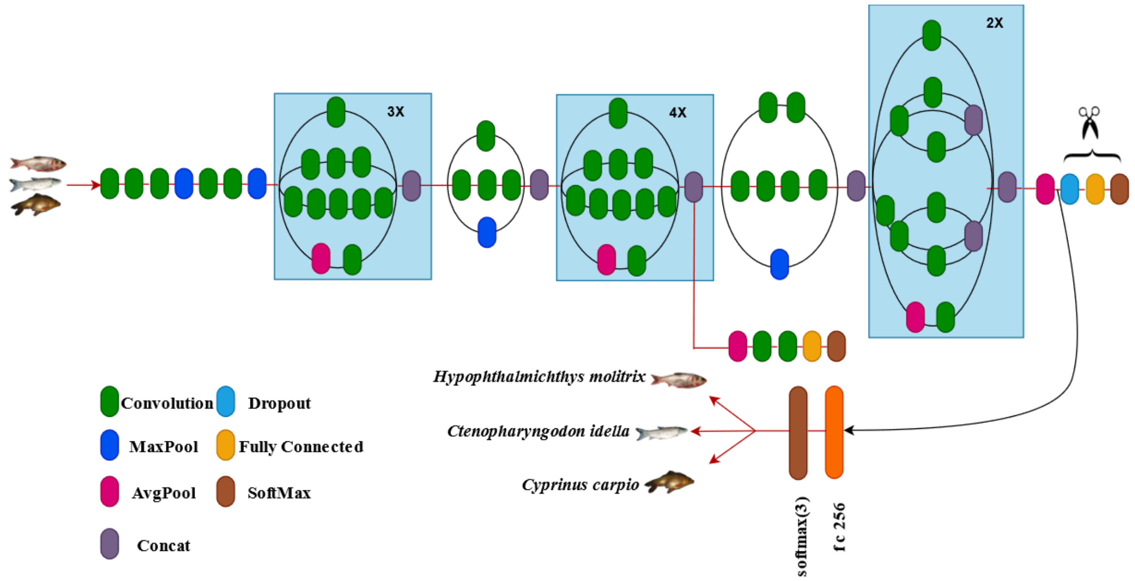


Figure 3.19: Architecture of InceptionV3 with transfer learning approach (Banerjee et al. [2])

InceptionV3 (Li et al. [49])

The GoogleNet architecture was introduced as GoogleNet (InceptionV1), later refined as InceptionV2, and recently as InceptionV3 (Figure 3.19). The Inception modules are treated as convolutional feature extractors, capable of learning enriched representations with fewer parameters. In general, the convolutional layer attempts to learn filters, with 2 spatial dimensions (width and height) and a channel dimension of the image. In this case, a single convolution kernel is used to look at cross-channel and spatial correlations at the same time.

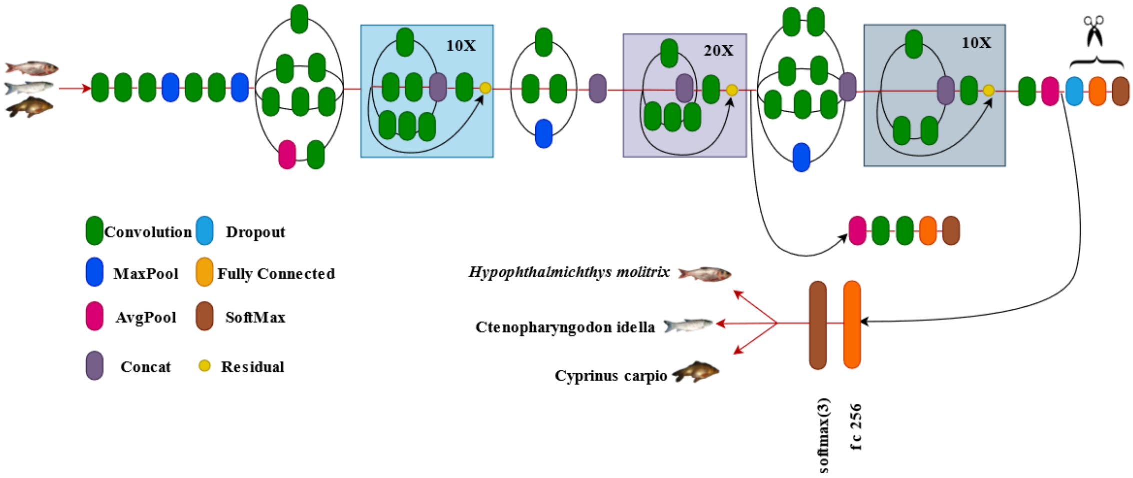


Figure 3.20: Architecture of Inception ResNetV2 with transfer learning approach (Banerjee et al. [2])

Inception ResNetV2 (Li et al. [49])

Residual Networks (ResNet) are deep convolutional networks in which the primary idea is to skip blocks of convolutional layers by forming residual blocks utilizing shortcut connections (Fig. 3.20). The training efficiency of these stacked residual blocks is considerably improved, and the degradation problem in deep networks is largely resolved. InceptionResNetV2 is a convolutional neural network trained on over a million images from the ImageNet database. The network has a total of 164 layers and is trained to classify images into 1000 object categories in ImageNet, such as the keyboard, mouse, pencil, and many animals. A 299 × 299 image is utilized as an input, and the result is class probability estimation. Here, jointly the Inception structure and the Residual connection are used. The core architecture of the InceptionResnetV2 architecture is shown in Figure 3.20.

MobileNet (Li et al. [49])

The MobileNet architecture is specifically designed for mobile and embedded vision applications. It utilizes a simplified design that constructs lightweight deep neural networks by employing depth-wise separable convolutions (Figure 3.21). Two simple global hyperparameters are presented that efficiently trade off latency and accuracy.

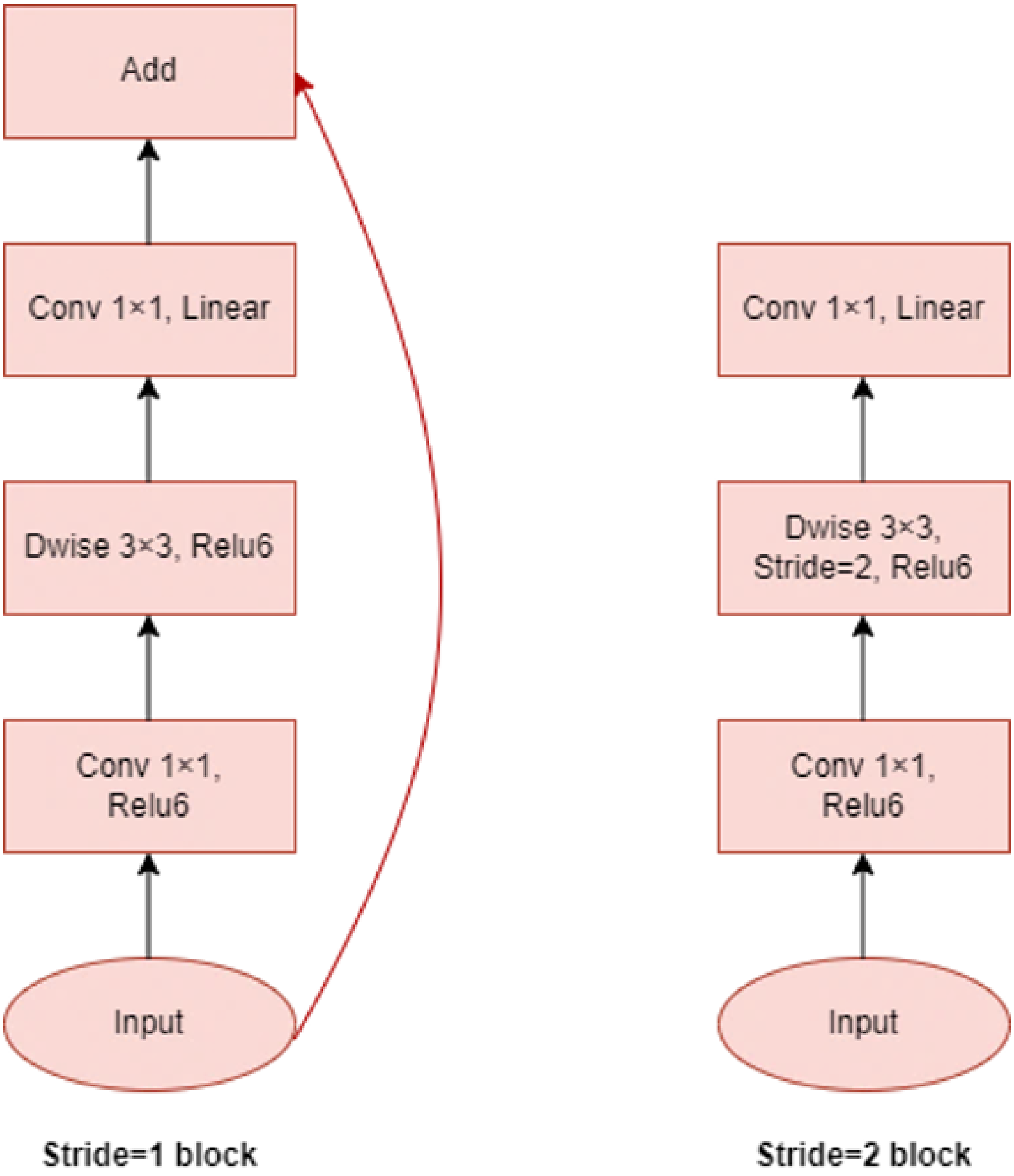
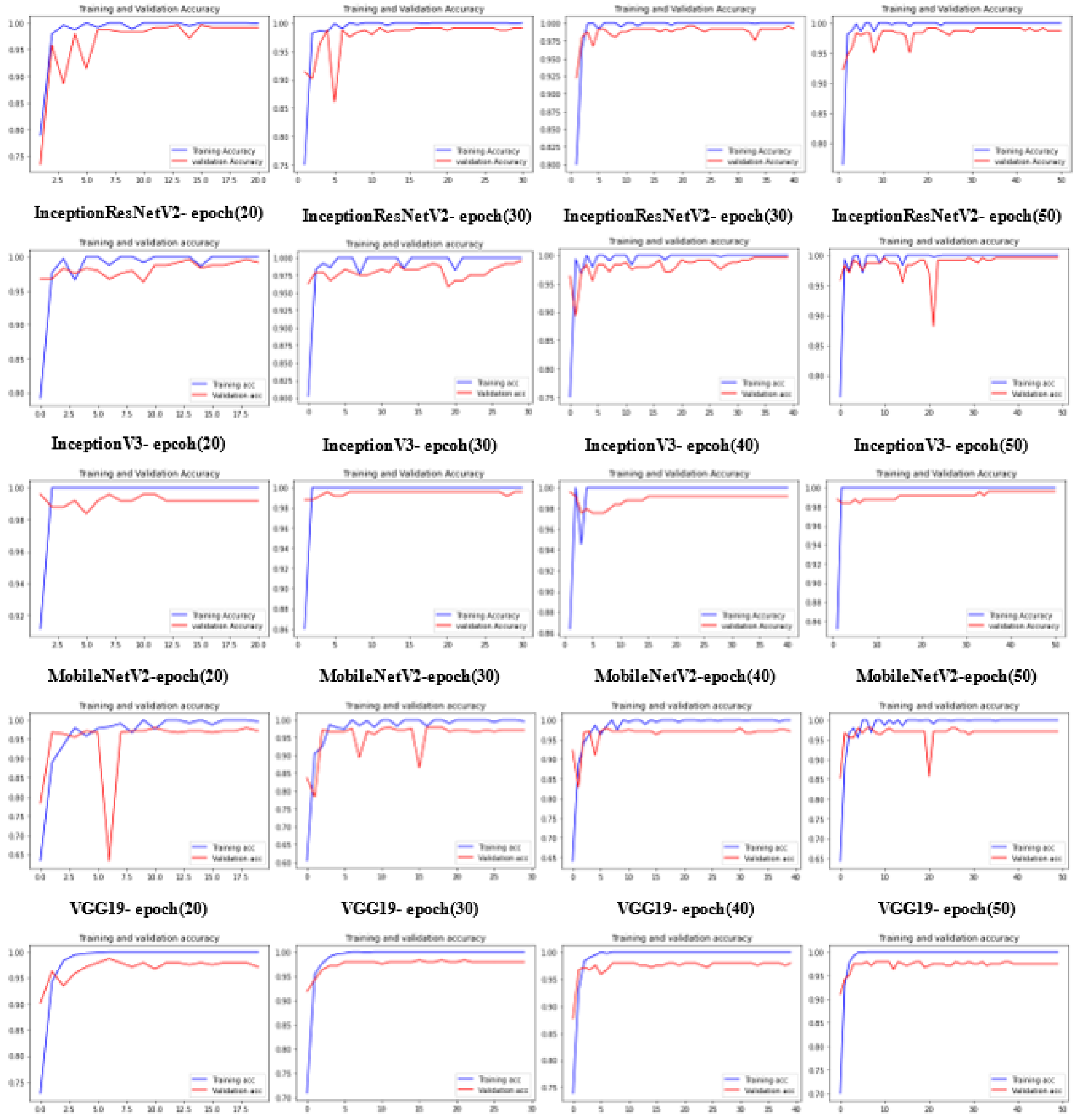


Figure 3.21: Architecture of MobileNet with transfer learning approach (Banerjee et al. [2])

Depth-wise separable filters are the foundation of MobileNet. The network's structure is another factor that contributes to higher performance.

3.3.4 Experiment Protocols and Results

VGG16, VGG19, InceptionV3, InceptionResNetV2, and MobileNetV2 deep neural networks were used with the transfer learning paradigm to identify three exotic carp present in the **JUDVLP-WBUAFS**: Fishdb-EC.v1 dataset. These networks have already been trained on the ImageNet challenge and have demonstrated their effectiveness. The fully connected layer and softmax layer of the above pre-trained networks are removed



InceptionResNetV2- epoch(20) InceptionResNetV2- epoch(30) InceptionResNetV2- epoch(30) InceptionResNetV2- epoch(50)

InceptionV3- epcoh(20) InceptionV3- epcoh(30) InceptionV3- epcoh(40) InceptionV3- epcoh(50)

MobileNetV2-epoch(20) MobileNetV2-epoch(30) MobileNetV2-epoch(40) MobileNetV2-epoch(50)

VGG19- epoch(20) VGG19- epoch(30) VGG19- epoch(40) VGG19- epoch(50)

VGG16-epoch(20) VGG16-epoch(30) VGG16-epoch(40) VGG16-epoch(50)

Figure 3.22: Accuracy graph of used pre-trained deep learning models for different epochs (Banerjee et al. [2])

and attached a fully connected layer with 256 nodes and a softmax layer with three nodes instead of thousand classes. At the time of training, the weights of the pre-trained layers are frozen, and the weights of the fully connected layer are tuned to the target problem. Transfer learning is an excellent technique applied here, because the dataset does not contain enough samples to train a deep learning network from scratch. In addition, the transfer learning technique reduces the time needed in training of neural network models. All of these pre-trained networks were applied on the **JUDVLP-WBUAFS**: Fishdb-EC.v1 dataset in Google CoLab using Keras environment for different epochs (20, 30, 40, and 50) with a learning rate of 1e-4. The categorical cross-entropy was used as the loss function. In VGG16, and VGG19, the input image is of shape 256×256×3, and in InceptionV3, InceptionResNetV2, and MobileNetV2 the input image is of shape

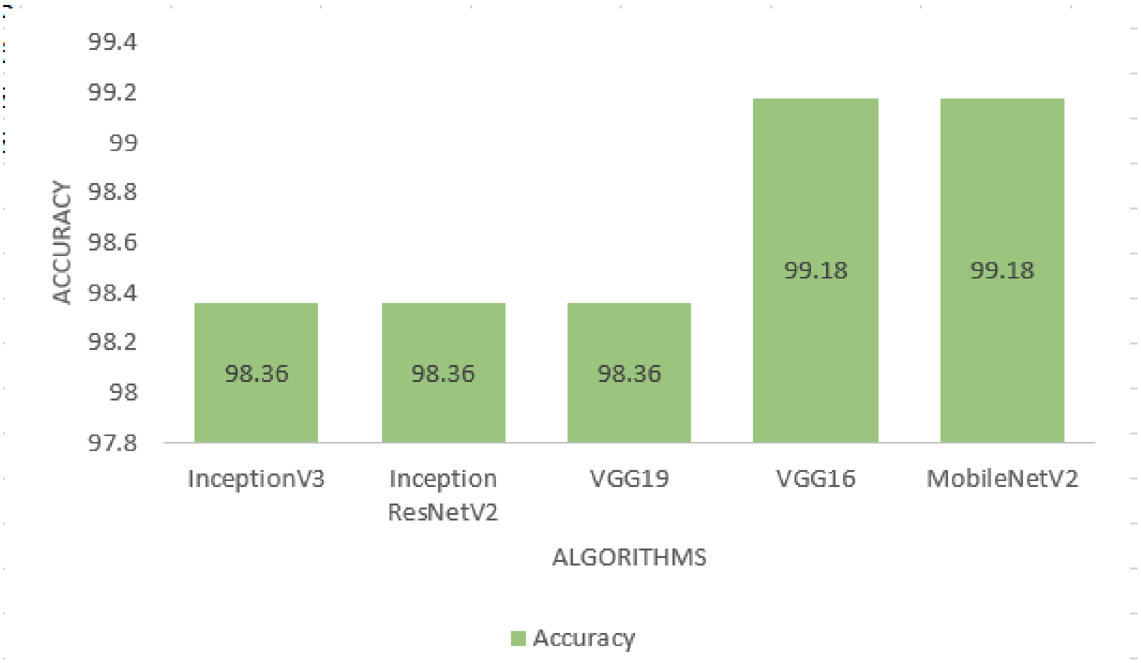


Figure 3.23: Accuracy of best performing pre-trained deep neural models on the identification of three major exotic carps (Banerjee et al. [2])

224×224×3. There are 735, 245, and 245 photos in the training, validation, and testing datasets, respectively. Running for 40 epochs, VGG16 obtained 99.18% accuracy and 98.36% accuracy was obtained when run for 20, 30, and 50 epochs. Accuracy of 98.36% was achieved using VGG19 after running for 20 epochs, and 97.95% accuracy was achieved when run for 30, 40, and 50 epochs. InceptionV3 obtained 98.77% accuracy by running 50 epochs, and 98.36% accuracy when run for 20, 30, and 40 epochs. InceptionResnetV2 obtained 98.36% accuracy when run for 30 and 40 epochs and 97.95% accuracy when for 20 and 50 epochs. The lightweight MobileNetV2, achieved 99.18% accuracy when run for 20 epochs, 98.77% accuracy after running for 30 and 40 epochs, and 98.36% accuracy for 50 epochs. The maximum accuracy of 99.18% was achieved using MobileNetv2 and VGG16 networks when run for 20 and 40 epochs respectively (refer Figure 3.22 and Figure 3.23). In Table 3.3, precision, recall, and F1-score of all pre-trained best performing deep learning models used here are presented. MobilNetV2 and VGG16 successfully recognize all *Cyprinus carpio* species. MobileNetv2 incorrectly recognizes one *Ctenopharyngodon idella* species as *Cyprinus carpio* and one *Hypophthalmichthys molitrix* as *Cyprinus carpio*. VGG16 correctly recognizes all *Hypophthalmichthys molitrix* species. A total of two *Ctenopharyngodon idella* species are wrongly recognized by VGG16 (one *Ctenopharyngodon idella* as *Cyprinus carpio* and one *Ctenopharyngodon idella* as *Hypophthalmichthys molitrix*). The individual performances of the other networks are available in the confusion matrix given in Figure 3.24.

Table 3.3: Precision, Recall and F1-score of best performing Inception ResnetV2, MobileNet, In-ceptionV3, VGG19 and VGG16 pre-trained models on exotic carp fish dataset Banerjee et al. [2]

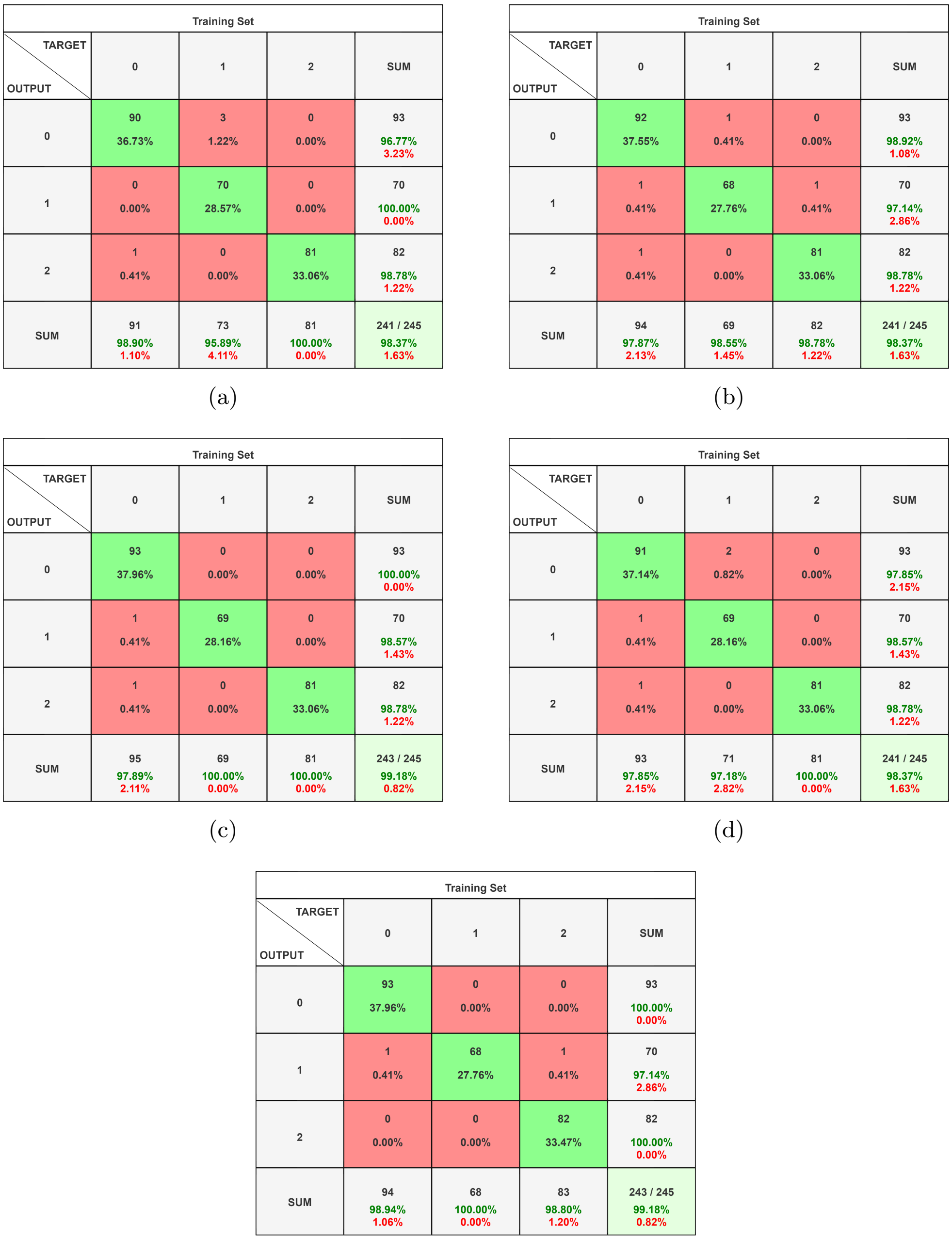
Algorithm	Fish Species	Precision	Recall	F1-score
Inception ResNetV2	*Cyprinus carpio*	0.99	0.97	0.98
	Ctenopharyngodon idella	0.96	1.0	0.98
	Hypophthalmichthys molitrix	1.0	0.99	0.99
MobileNetV2	*Cyprinus carpio*	0.98	1.0	0.99
	Ctenopharyngodon idella	1.0	0.99	0.99
	Hypophthalmichthys molitrix	1.0	0.99	0.99
InceptionV3	*Cyprinus carpio*	0.98	0.99	0.98
	Ctenopharyngodon idella	0.99	0.97	0.98
	Hypophthalmichthys molitrix	0.99	0.99	0.99
VGG19	*Cyprinus carpio*	0.98	0.98	0.98
	Ctenopharyngodon idella	0.97	0.99	0.98
	Hypophthalmichthys molitrix	1.0	0.99	0.99
VGG16	*Cyprinus carpio*	0.99	1.0	0.99
	Ctenopharyngodon idella	1.0	0.97	0.99
	Hypophthalmichthys molitrix	0.99	1.0	0.99

3.4 Conclusion

Due to a regional shortage of fish taxonomists, the developed system will assist the common people in identifying the appropriate exotic carp at the fish market. This study specifically investigates an automatic method for recognizing three important exotic carp. Some popular pre-trained deep neural networks- VGG16, VGG19, InceptionV3, Inception ResnetV2, and MobileNetV2, are applied here. MobileNetV2 and VGG16 have obtained the best accuracy of 99.18% on the test dataset, comprising a total of 245 images of three exotic carp fish species. Despite the high performance of MobileNetV2, and VGG16, some misclassification occurs for the *Hypophthalmichthys molitrix* and *Ctenopharyngodon idella* species. The shape and size of these two fish species are strikingly similar. The primary cause of inaccurate classifications is background changes, which arise from the unconstrained acquisition of images from several fish markets. The authors believe that the inclusion of more samples is necessary to make a large and challenging dataset. Additionally, clubbing the other fish species is suggested to create a more diverse dataset and expand this field.

3.5 Summary

Automatic fish species recognition offers significant benefits in many areas of aquaculture industries, as well as educational and outreach activities. Real-time monitoring of fish catches is critical to maintaining sustainable fishing practices. It aids in identifying instances of overfishing, thereby contributing to the true preservation of biodiversity. Identifying fish species automatically can aid in conservation efforts by monitoring



Training Set				
TARGET OUTPUT	0	1	2	SUM
0	90 36.73%	3 1.22%	0 0.00%	93 96.77% 3.23%
1	0 0.00%	70 28.57%	0 0.00%	70 100.00% 0.00%
2	1 0.41%	0 0.00%	81 33.06%	82 98.78% 1.22%
SUM	91 98.90% 1.10%	73 95.89% 4.11%	81 100.00% 0.00%	241 / 245 98.37% 1.63%

(a)

Training Set				
TARGET OUTPUT	0	1	2	SUM
0	92 37.55%	1 0.41%	0 0.00%	93 98.92% 1.08%
1	1 0.41%	68 27.76%	1 0.41%	70 97.14% 2.86%
2	1 0.41%	0 0.00%	81 33.06%	82 98.78% 1.22%
SUM	94 97.87% 2.13%	69 98.55% 1.45%	82 98.78% 1.22%	241 / 245 98.37% 1.63%

(b)

Training Set				
TARGET OUTPUT	0	1	2	SUM
0	93 37.96%	0 0.00%	0 0.00%	93 100.00% 0.00%
1	1 0.41%	69 28.16%	0 0.00%	70 98.57% 1.43%
2	1 0.41%	0 0.00%	81 33.06%	82 98.78% 1.22%
SUM	95 97.89% 2.11%	69 100.00% 0.00%	81 100.00% 0.00%	243 / 245 99.18% 0.82%

(c)

Training Set				
TARGET OUTPUT	0	1	2	SUM
0	91 37.14%	2 0.82%	0 0.00%	93 97.85% 2.15%
1	1 0.41%	69 28.16%	0 0.00%	70 98.57% 1.43%
2	1 0.41%	0 0.00%	81 33.06%	82 98.78% 1.22%
SUM	93 97.85% 2.15%	71 97.18% 2.82%	81 100.00% 0.00%	241 / 245 98.37% 1.63%

(d)

Training Set				
TARGET OUTPUT	0	1	2	SUM
0	93 37.96%	0 0.00%	0 0.00%	93 100.00% 0.00%
1	1 0.41%	68 27.76%	1 0.41%	70 97.14% 2.86%
2	0 0.00%	0 0.00%	82 33.47%	82 100.00% 0.00%
SUM	94 98.94% 1.06%	68 100.00% 0.00%	83 98.80% 1.20%	243 / 245 99.18% 0.82%

(e)

Figure 3.24: Confusion matrix of the pretrained deep learning best performing model used in this study (Banerjee et al. [2])

populations and habitats. Traditional methods of fish identification involve trained personnel in fish anatomy for manual observations. It is time-consuming and prone to mistakes. Automatically identifying the fish using machine or deep learning plays a vital role in deploying various monitoring systems, such as underwater cameras or drones, to continuously collect data on fish populations. This chapter focuses primarily on recognizing Indian major carps and exotic carps in a non-cluttered environment. During the data preparation stage, manual segmentation of the fish body was performed to remove the surrounding fish and other objects. Various autoencoder architectures, such as simple autoencoder, deep autoencoder, and deep convolutional autoencoder, have been applied to extract the latent features of the data. Different classifiers such as Logistic Regression (LR), Naive Bayes (NB), K-Nearest neighbor (KNN), Support Vector Machine (SVM), and Random Forest (RF) are used to predict the testing samples. The results are compared with some popular traditional feature extraction methods. Some popular pre-trained deep learning architectures such as InceptionV3, Inception ResNetV2, MobileNetV2, VGG19, and VGG16 are utilized in classifying the three popular exotic carp found in India. The transfer learning technique is utilized to handle the small dataset sizes and apply the learned weights on large datasets like ImageNet. These networks are fine-tuned on the **JUDVLP-WBUAFS**: Fishdb-EC.v1 dataset in the fully connected network stage. Fish species recognition in the non-cluttered environment on the above two developed datasets **JUDVLP-WBUAFS**: Fishdb-IMC.v1 and **JUDVLP-WBUAFS**: Fishdb-EC.v1 dataset exhibits the effectiveness in the fishery industry and helps the common people in the market to identify fish without asking for help from someone. But, in the live fish market conditions, many fish species are found in overlapping conditions with different lighting and backgrounds. The current challenge now is to recognize fish species in the cluttered environment where many fish species are present. In the next chapter, different approaches for recognizing fish species in some cluttered environments utilizing object localization and segmentation techniques are discussed in detail.

Chapter 4

Fish Species Recognition in Cluttered Environment

This chapter presents some developments on fish species recognition in some cluttered environments, typically found in live fish markets. It can be challenging to recognize fish species in some cluttered environments due to their unorganized presence, often resulting in the placement of one fish on top of another. In such a situation, a significant portion of the fish's body is obscured, making it impossible to be taken into account for automatic recognition. Variations in light intensity, and backgrounds, and the presence of other fish and non-fish objects complicate the task and make it more challenging. The automatic recognition of fish species in cluttered environments, typically found in fish markets, aids in various tasks such as counting different fish, estimating size and weight, and identifying fish that meet consumers' nutritional requirements, among others. In this chapter, the solution to this problem is proposed using two approaches: one employs the object localization and recognition technique, while the other employs the semantic segmentation technique.

4.1 Introduction

Indian fish markets are generally unorganized, with moderate crowds present during the daytime. Consumers often struggle to distinguish between different types of fish in the market. Proper knowledge of fish taxonomy and experience are the key factors in identifying the fish properly. Recognizing fish in a cluttered environment becomes more challenging. In fish markets, different fish species are placed together, and mostly one fish is placed on top of the other fish, creating occlusion. The task is complex due to changes in light intensity, different background conditions, the presence of unknown fish species, and other objects. Fish identification and counting in cluttered environments are important tasks in various fisheries industries. The fishery industry requires trained personnel to manually identify and count the fish species. A lot of trained manpower is essential to performing this monotonous job. During the day, consumers have limited time in the market and often struggle to identify fish, necessitating assistance from someone with expertise in fish taxonomy. However, obtaining immediate assistance in a crowded market can be challenging. Consumers must be self-sufficient in identifying different fish, and this necessitates the development of an automatic application to identify fish species in a cluttered environment. Taking into account the industrial and societal needs, an automatic fish species recognition and counting framework is proposed. Deep learning object detection networks are used to localize and detect the fully visible fish species from the images. A rectangular bounding box is used in object detection and thus includes some portion of the background around the object.

4.2 Related Works

Identification and monitoring of fish species has become a popular area of research. Researchers have solved this problem using two different ways, traditional machine learning and deep learning. Different morphological features were used to identify the fish species. The skin texture, wavelet feature, different statistical features, and color moments were used to find feature vectors. Different machine learning algorithms were then used on these feature vectors. In 2012, Hu et al. [12] proposed a technique to use fish skin texture, wavelet feature, and different statistical features as feature vectors. A total of 540 images of six fish species were used in their study. For the classification, SVM was used. Rossi et al. [16] in 2016 proposed a technique to find different geometric features from the preselected anchor points. They have used 339 images of seven fish species, collected from the Turin (Italy) fish market. Tharwat et al. [19] combinedly used the fish skin texture, Weber Local Descriptor (WLD), and color features to classify 241 fish images of four fish species. Recently, different deep learning algorithms, such as modified AlexNet, VGGNet, etc., were used for the recognition of fish species. In 2018, Rachmatullah and Supriana [17] used a convolutional neural network to classify 15 different fish species. Hussain et al. [20] used a modified AlexNet, consists of four convolutional layers and two fully connected layers, to classify six fish species. The QUT fish dataset by Anantharajah et al. [21] was used for training, and testing, and validation was done using the LifeClef 2015 fish dataset by Joly et al. [22]. The modified AlexNet outperformed the AlexNet and VGGNet. Optimization-based VGGNet by Montalbo and Hernandez [23] was used on 530 images of the FishBase dataset. Rauf et al. [24] proposed a 32-layer CNN architecture for classifying six freshwater fish species in Asia. They did the experiments on the 438 images from the Fish-Pak dataset. The proposed CNN architecture was applied to the fish body, fish head, and fish scale separately. Banerjee et al. [1] proposed a dataset named **JUDVLP-WBUAFS**: Fishdb-IMC.v1 of 1500 images of Indian major carps and used an autoencoder to find latent features. They have applied different machine learning techniques as well as deep learning techniques to classify the species. Garcia et al. [50] used deep learning for automatic segmentation of fish images in commercial trawling and measured the size of the fish to reduce the catches of undersized fish. Banerjee et al. [2] proposed a dataset named **JUDVLP-WBUAFS**: Fishdb-EC.v1 and applied VGG-16, VGG19, InceptionV3, InceptionResNetV2, and MobileNet to classify the exotic carps.

4.3 Localization and Detection of Fish Species, Counting Fish Species in Cluttered Environment

Globally, a total of 34,800 fish species have been recognized in various aquatic ecosystems (Froese and Pauly [48]). Many other fish species exist but remain unrecognized. India is a river-centric country, with approximately thousands of freshwater fish species. India is the third-largest producer of fish and the second largest aquaculture fish producer. Out of the total aquaculture fish production, a lion's share of it is contributed by carps. Around 60% of the carp production in India is done by Indian Major Carps (IMC) which consists of *Labeo rohita*, *Labeo catla*, and *Cirrhinus mrigala*. In India, around 60% of the total population consumes fish, and the annual per capita consumption of the total Indian population is approximately 6.31 kg (MoFAHD [51]). Currently, Andhra Pradesh is the largest inland fish producer state in India, followed by West Bengal. Identifying different fish species in the fish market is difficult for consumers. Proper knowledge of fish taxonomy and experience are the key factors in correctly identifying the fish species. Recognizing and counting fish species is a critical task in various fisheries industries that require trained personnel. Manually identifying the species and counting is a monotonous process that requires a lot of trained manpower. Most fish markets in India are unorganized, with a moderate crowd present during the daytime. Due to a lack of time when purchasing fish at the market, consumers frequently struggle to identify them and may require assistance from someone with expertise in fish taxonomy. Nevertheless, it is not possible to obtain immediate assistance from experts; one must be self-sufficient, which necessitates the development of a suitable tool that uses machine learning to identify fish species in the market. Taking into account both industrial and societal needs, an automatic fish species recognition and counting framework is proposed, that is suitable in some cluttered environments. After applying a total of nine different augmentations to the training dataset, a dataset consisting of 2,281 images was prepared to carry out the task. A total of 19,995 instances of six fish species- *Labeo catla*, *Labeo rohita*, *Cirrhinus mrigala*, *Labeo bata*, *Hypophthalmichthys molitrix*, and *Ctenopharyngodon idella*, are present in the proposed dataset. Because the images were collected in cluttered environments, maximum fish species are obscured by surrounding fish species. The body part is not often visible in the fish market's cluttered environment. Only the head part of the fish is often visible. Therefore, a rectangular box is used to annotate only the fish heads. Two popular deep learning-based object detection networks, YOLOv3 by Redmon and Farhadi [52] and YOLOv5 by Jocher et al. [53], with different variants were used with the original and augmented datasets. The main contributions of this study are as follows:

- A diversified dataset for fish species recognition and counting with 2,281 images is proposed.

- Annotation for the ground-truth fish recognition is prepared to carry out different experiments using deep learning algorithms.

- A fish species recognition and counting framework is proposed for concerned stakeholders.

4.3.1 Dataset Preparation

A dataset named **JUDVLP-WBUAFS**: Fishdb-Detection.v1 with 400 images of six fish species, Catla (*Labeo catla*), Rohu (*Labeo rohita*), Mrigal(*Cirrhinus mrigala*), Bata (*Labeo bata*), Silver carp(*Hypophthalmichthys molitrix*), and Grass carp (*Ctenopharyngodon idella*) was created. A normal smartphone camera was used for collecting images from the different fish markets under cluttered conditions. The entire dataset collection process is explained in subsection 2.3. A total of 3,128 fish species are present in the proposed dataset. Out of the six fish species, *Labeo rohita* is the class with the most instances, and *Ctenopharyngodon idella* is the class with the least number of instances. The normal splitting of the dataset into training, validation, and testing parts might make it more imbalanced because the original dataset is already imbalanced. A stratified group shuffling based splitting technique was used to split the dataset into training, validation, and testing parts in a 60:20:20 proportion to maintain the balanced percentage allocation of each fish species. Following the data split, the training, validation, and testing parts comprised 209, 116, and 75 images respectively.

Deep learning networks need a sufficient amount of diversified data to work properly without overfitting. The proposed dataset consists of 400 images, which is not appropriate for deep learning networks. So, nine different augmentations like horizontal flip, vertical flip, blur, gaussian noise, hue saturation, RGBshift, 45°rotation, 90°rotation, and 180°rotation were applied to the training part of the proposed **JUDVLP-WBUAFS**:Fishdb-Detection.v1 dataset to prepare a large dataset that fits deep learning networks. After the augmentation, the training part consisted of 2,090 images and a total of 19,995 instances of six fish species.

4.3.2 Methodology

Object Localization and Detection

Object localization and detection are fundamental tasks in computer vision and image processing. These tasks involve identifying and locating objects within an image or a video frame.

Image classification is a fundamental problem in computer vision that involves categorizing an image into preset classes or categories. It involves assigning a label or a class to an entire image based on its content. The input to an image classification system is typically a single image, and the output is a label indicating the category to which the image belongs. Image classification maps an entire image into a single label, which means it considers that a single object is present in an image.

Object localization is about drawing a tight bounding box around the object that is supposed to be located in the image. It focuses specifically on determining the precise

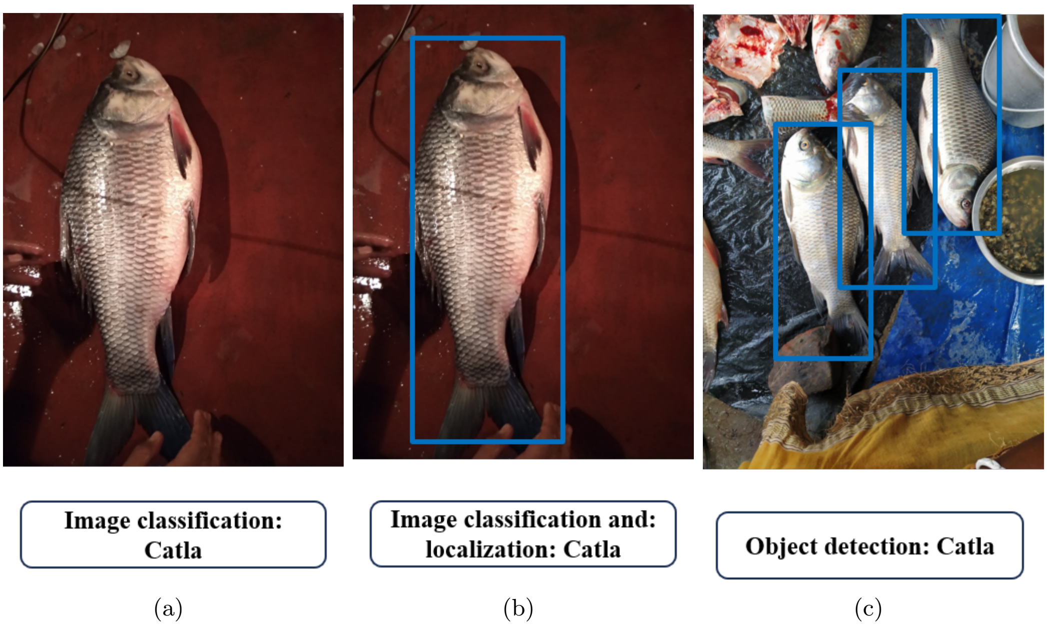


Image classification: Catla
Image classification and: localization: Catla
Object detection: Catla
(a)
(b)
(c)

Figure 4.1: Representation of (a) Image Classification, (b) Image classification and localization, and (c) Object Detection

location of objects within an image. Instead of identifying multiple objects and drawing bounding boxes around them like in object detection, object localization aims to locate a single object within an image by specifying a bounding box that tightly encloses it. Object localization algorithms usually output the coordinates of the bounding box surrounding the object.

Object detection entails identifying objects of interest within an image or video frame and drawing bounding boxes around them. Object detection algorithms not only classify the objects in the image but also provide their precise locations. It is the superset of object localization and differs from classification and localization in that it detects multiple objects in a scene compared to localizing and classifying a single object. The Figure 4.1 shows the image classification, image classification and localization, and object detection with an example.

Different Object Detection Technique

Object detection is performed using traditional computer vision techniques and deep learning techniques. Before 2014, object detection primarily relied on traditional computer vision techniques and handcrafted features. Developed by David Lowe in 1999, SIFT is a method for detecting and describing local features in images. SIFT features are invariant to image scale, rotation, and illumination changes, making them useful for object recognition and matching. In 2001, the Viola-Jones algorithm by Viola and Jones [54] was developed, which is based on Haar-like features (Viola and Jones [54]) and employs a cascade of classifiers trained using the AdaBoost algorithm (Freund and Schapire [55]). It was primarily used for face detection but could be extended to detect other objects with modifications. Introduced by Navneet Dalal and Bill Triggs in 2005, HOG (Dalal and Triggs [56]) is a feature descriptor used for object detection. It calculates the distribution of gradient orientations in an image to capture its shape and appearance. HOG features are commonly used with support vector machines (SVMs) or other classifiers for object detection tasks. Deformable Part Models (DPM) developed in 2008 represent objects as a collection of parts with flexible spatial relations. Deep learning techniques have come into the picture for object detection from 2014 onwards.

Two-stage object detectors are a class of algorithms used in computer vision for object detection tasks. These detectors consist of two main stages: region proposal generation and object classification. Region-based Convolutional Neural Networks(R-CNN) proposed by Ross Girshick et al. in 2014 (Girshick et al. [57]) operates in multiple stages: it first generates region proposals using selective search, then extracts features from each proposed region using a pre-trained CNN, and finally classifies and refines the bounding boxes using SVMs and linear regressions. While effective, R-CNN is computationally expensive due to its multi-stage approach. Faster R-CNN was introduced by Shaoqing Ren et al. in 2015 (Ren et al. [58]). It combines a Region Proposal Network (RPN) with a region-based classifier to detect objects. The RPN generates region proposals (bounding boxes) from the input image, which are then classified and refined by the region-based classifier. Mask R-CNN, introduced by Kaiming He et al. (He et al. [59]), is a modified version of Faster R-CNN by adding a branch for predicting segmentation masks for each object instance. In addition to bounding box prediction and classification, Mask R-CNN generates pixel-wise masks to precisely delineate object boundaries. It is widely used for instance segmentation tasks, where both object detection and segmentation are required. Cascade R-CNN, proposed by Zhaowei Cai et al. in 2018 (Cai and Vasconcelos [60]), improves object detection performance by employing a cascade of classifiers. It consists of multiple stages, with each stage focusing on refining the detection results of the previous stage by removing false positives and improving localization accuracy. Two-stage detectors have been the dominant approach in object detection for many years due to their high accuracy and flexibility. However, they tend to be slower, and it makes them less suitable for real-time applications where low latency is critical.

Single-stage object detectors are a class of algorithms used in computer vision for object detection tasks. Unlike traditional two-stage detectors that first propose regions of interest (RoIs) followed by classification and refinement, single-stage detectors directly predict both thew object bounding boxes and class probabilities in a single pass through the network. This approach typically results in faster inference times compared to two-stage detectors. YOLO (You Only Look Once) proposed by Joseph Redmon et al. in 2016 (Redmon et al. [61]), and SSD(Single Shot MultiBox Detector) proposed by Wei Liu et al. in 2016 (Liu et al. [62]) is the two popular single-stage object detection algorithms. RetinaNet was introduced by Tsung-Yi Lin et al. in 2017 (Lin et al. [63]) to address the class imbalance problem in object detection. It employs a focal loss function that down-weights the loss assigned to well-classified examples, reducing the influence of easy

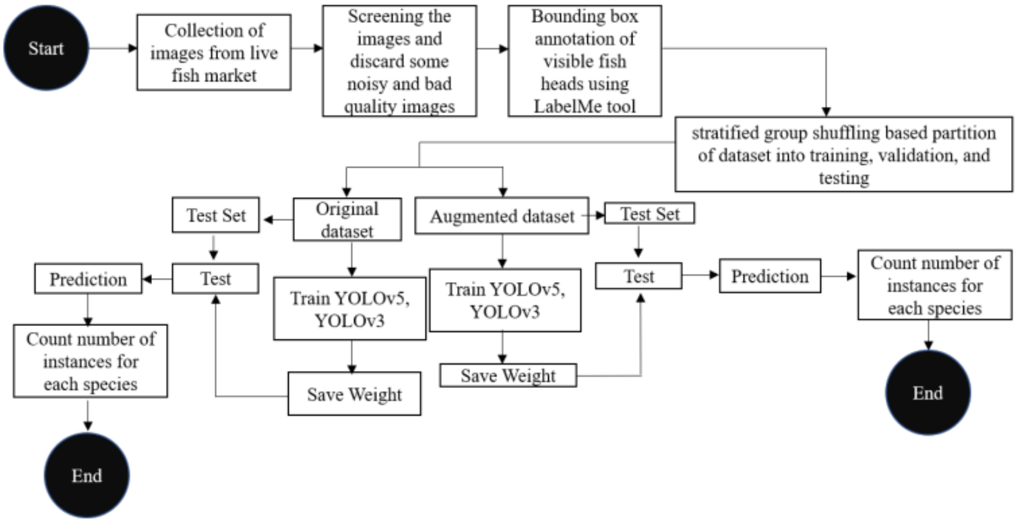

Start
Collection of images from live fish market
Screening the images and discard some noisy and bad quality images
Bounding box annotation of visible fish heads using LabelMe tool
stratified group shuffling based partition of dataset into training, validation, and testing
Test Set
Original dataset
Augmented dataset
Test Set
Prediction
Test
Train YOLOv5, YOLOv3
Train YOLOv5, YOLOv3
Test
Prediction
Count number of instances for each species
Count number of instances for each species
Save Weight
Save Weight
End
End

Figure 4.2: Workflow diagram of the proposed method of recognising and counting fish Species in some cluttered environment (Banerjee et al. [3])

negatives during training. EfficientDet, based on EfficientNet, proposed by Mingxing Tan et al. in 2019 (Tan et al. [64]) focuses on optimizing both accuracy and efficiency. It scales up the network depth and width using a compound scaling method to achieve better performance. Single-stage detectors have gained popularity due to their simplicity and efficiency. They are widely used in applications where real-time processing or low-latency inference is crucial.

Proposed Approach

This chapter addresses the problem of automatic fish species recognition and counting in cluttered environments by focusing solely on the head portion of the fish. The head portion of the fish plays an important role in differentiating fish species at a single glance. There are many fish species whose head portions are quite similar. This requires the use of other parts of the fish body for species recognition. The head portions of the six fish species considered in this study exhibit significant differences from one another. So, an object detection framework was designed to recognize and count fish species in live fish market conditions. The experiments were carried out on both the original and augmented datasets. YOLOv5, and YOLOv3, with some variants, were applied to the training dataset. A validation set was used at training time to learn and adjust weights. Finally, the trained weights were applied to the test dataset to evaluate the efficacy of the proposed approach. After the prediction of the bounding box for the visible fish heads, the number of instances for each fish species was calculated. The overall methodology is presented in Figure 4.2. The next few subsections explain, with a proper diagram, the architecture of the used deep learning algorithms.

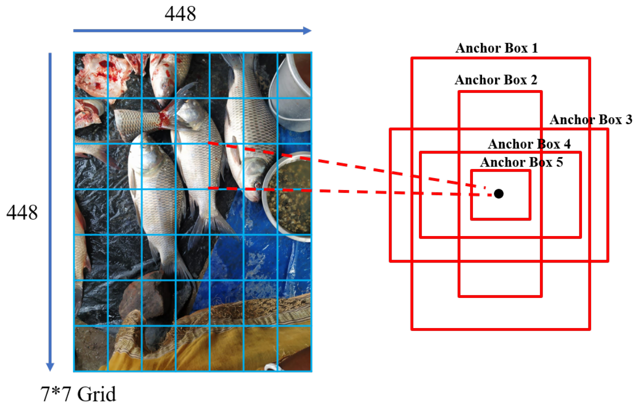


Figure 4.3: Label encoding process in YOLO-V1

YOLO Object Detection Model

The YOLO (You Only Look Once) object detection model(Redmon et al. [61]) is an end-to-end neural network architecture that provides bounding boxes as well as the probabilities of each class involved in the object detection task. YOLO is particularly known for its real-time performance, capable of detecting objects in images and video streams at impressive speeds. Compared to conventional multi-stage object detectors, YOLO performs object detection in a single forward pass of the neural network. The overall working of the first YOLO model (YOLO-V1) for object detection is discussed here.

The label encoding of the dataset is done at first based on the ground truth annotation. Here, the image is divided into 7 × 7 grids, which means 49 cells. Relative values of the x and y coordinates Δx, and Δy are calculated based on which bounding box the anchor point falls into. Also, the relative values of the width and height are calculated. This process is represented in Figure 4.3. The equations to calculate the relative values of (x,y,w,h) are given in Equation 4.1.

$$\begin{aligned}
sf &= image_{width}/no_{ofgrids} \\
\Delta x &= (x - x_a)/sf \\
\Delta y &= (y - y_a)/sf \\
\Delta w &= w/image_{width} \\
\Delta h &= h/image_{width}
\end{aligned} \tag{4.1}$$

Suppose the bounding box centre position coordinate is $(130, 230)$ and the width and height of the bounding box is $(260, 300)$. The grid cell in which the centre of the object falls has the extreme left coordinate as $(110, 200)$. As per the Equation 4.1, the original values$(130, 230, 260, 300)$ are converted into relative values $(0.31, 0.47, 0.58, 0.67)$ corresponding to the grid cell and generally denoted as $(\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{w}, \Delta\hat{h}, \hat{c})$, where the last term $\hat{c}$ represent objectness score. If the object is present in some grid cell, the $\hat{c}$ value is 1, otherwise 0. It is the distance to be covered from the grid cell's left position to the object's centre. It is called the targets of each grid cell in the image. Targets for all the grid cells and class probabilities are calculated. If there is no object in a grid position, all zeroes are placed in the target vector. The class probabilities are one-hot encoded. During the YOLO-V1 training phase, a predicted map of dimension $7 \times 7 \times (10 + n)$ is generated considering n classes in the dataset. In Figure 4.4, the block diagram of the overall network architecture of YOLO-V1 is presented. Loss is computed with the ground truth target vector and a backpropagation is performed for further update in the network.

Each grid cell generates two bounding boxes ($B = 2$) in YOLO-V1. Each bounding box has 5 parameters in the target vector as $(\Delta x_i, \Delta y_i, \Delta w_i, \Delta h_i, c_i)_{i=1}{}^B$ and n parameters in the class probabilities as $(p_1, p_2, ..., p_n)$. So, the number of parameters per grid cell is $((2 \times 5) + n = 10 + n)$. The final feature map from YOLO-V1 is $7 \times 7 \times 30$. Out of the two bounding boxes for each grid cell, the bounding box with the maximum confidence score is selected. The confidence score is calculated as $c = c \times p$, where $p = max(p1, p2, ..., p_n)$.

Loss in YOLO-V1 is calculated as a mutually exclusive mixture of objectness loss and no objectness loss. It gives a greater weightage to the objectness loss. The overall loss is calculated as per Equation 4.2.

$$L = \sum_{i=1}^{S^2} 1_i^{object} * L_{i,obj} + \lambda_{no_{object}} \sum_{i=1}^{S^2} 1_i^{no_{object}} \times L_{i,no_{object}} \tag{4.2}$$

The objectness loss is calculated as a combination of bounding box loss, confidence loss, and class probability loss (refer Equation 4.3). The bounding box loss, confidence loss, and class probability loss are calculated as per Equation 4.3 to 4.6. In these equations $(\Delta \hat{x_i}, \Delta h\hat{y_i}, \Delta \hat{w_i}, \Delta \hat{h_i})$ is the ground truth target vector and $(\Delta x_i, \Delta y_i, \Delta w_i, \Delta h_i)$ is the predicted box that has the largest Intersection of Union(IoU) with the ground truth box. The final loss in YOLO-V1 is given in Equation 4.7.

$$L_{i,object} = \lambda_{coord} L_{i,object}^{boundingbox} + L_{i,object}^{confidence} + L_{i,object}^{class} \tag{4.3}$$

$$L_{i,object}^{boundingbox} = (\Delta x_i - \Delta \hat{x_i})^2 + (\Delta y_i - \Delta \hat{y_i})^2 + (\sqrt{\Delta w_i - \Delta \hat{w_i}})^2 + (\sqrt{\Delta h_i - \Delta \hat{h_i}})^2 \tag{4.4}$$

$$L_{i,object}^{confidence} = (c_i - \hat{c_i})^2 \tag{4.5}$$

$$L_{i,object}^{class} = \sum i = 1^n (p_{i,c} - \hat{p_{i,c}})^2 \tag{4.6}$$

$$\begin{aligned} L = \lambda_{coord} \times \sum_{i=1}^{S^2} 1_i^{object} \times (\Delta x_i - \Delta \hat{x_i})^2 + (\Delta y_i - \Delta \hat{y_i})^2 + (\sqrt{\Delta w_i - \Delta \hat{w_i}})^2 \\ +(\sqrt{\Delta h_i - \Delta \hat{h_i}})^2) + \sum_{i=1}^{S^2} 1_i^{object} \times (c_i - \hat{c_i})^2 + \sum_{i=1}^{S^2} 1_i^{object} \times \sum_{i=1}^{n} (p_{i,c} - \hat{p_{i,c}})^2 \\ +\lambda_{no\_object} \sum_{i=1}^{S^2} 1_i^{no\_object} \times \sum_{i=1}^{B} (c_{i,j} - \hat{c_{i,j}})^2 \end{aligned} \tag{4.7}$$

YOLO-V1 is much faster than the two-stage detectors because of its simple architecture

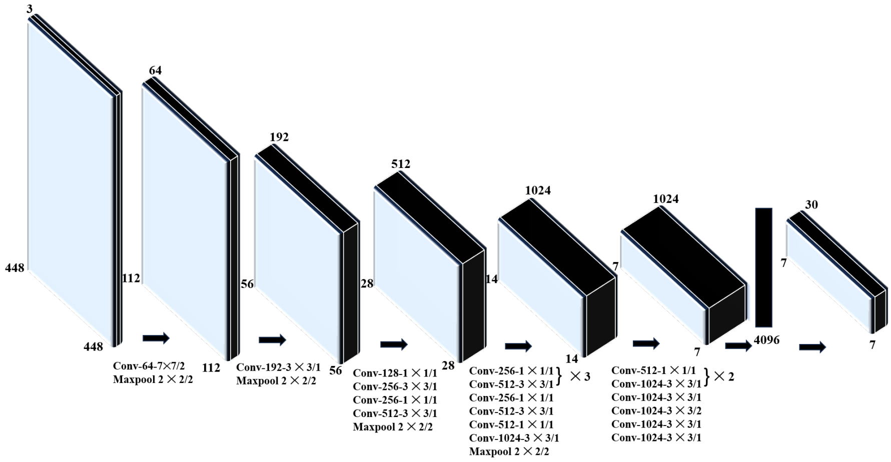
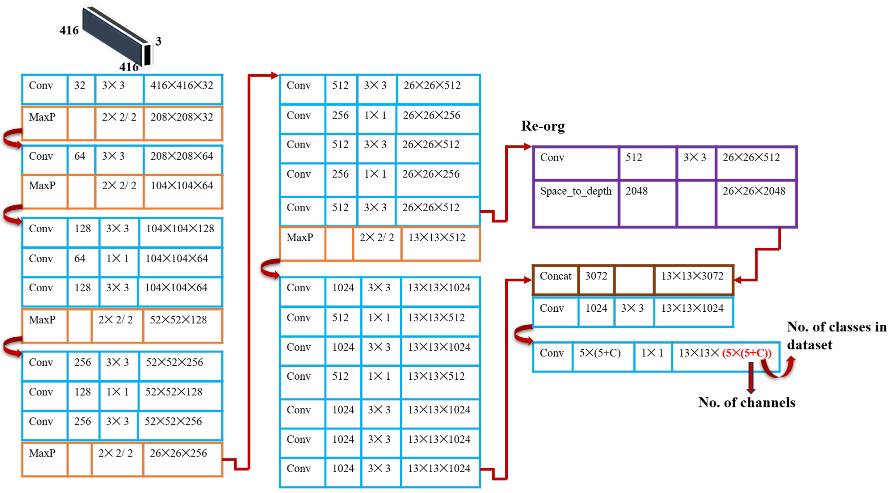


Figure 4.4: Architecture of YOLO-V1



Conv	32	3× 3	416×416×32
MaxP		2× 2/ 2	208×208×32
Conv	64	3× 3	208×208×64
MaxP		2× 2/ 2	104×104×64
Conv	128	3× 3	104×104×128
Conv	64	1× 1	104×104×64
Conv	128	3× 3	104×104×64
MaxP		2× 2/ 2	52×52×128
Conv	256	3× 3	52×52×256
Conv	128	1× 1	52×52×128
Conv	256	3× 3	52×52×256
MaxP		2× 2/ 2	26×26×256

Conv	512	3× 3	26×26×512
Conv	256	1× 1	26×26×256
Conv	512	3× 3	26×26×512
Conv	256	1× 1	26×26×256
Conv	512	3× 3	26×26×512
MaxP		2× 2/ 2	13×13×512
Conv	1024	3× 3	13×13×1024
Conv	512	1× 1	13×13×512
Conv	1024	3× 3	13×13×1024
Conv	512	1× 1	13×13×512
Conv	1024	3× 3	13×13×1024
Conv	1024	3× 3	13×13×1024
Conv	1024	3× 3	13×13×1024

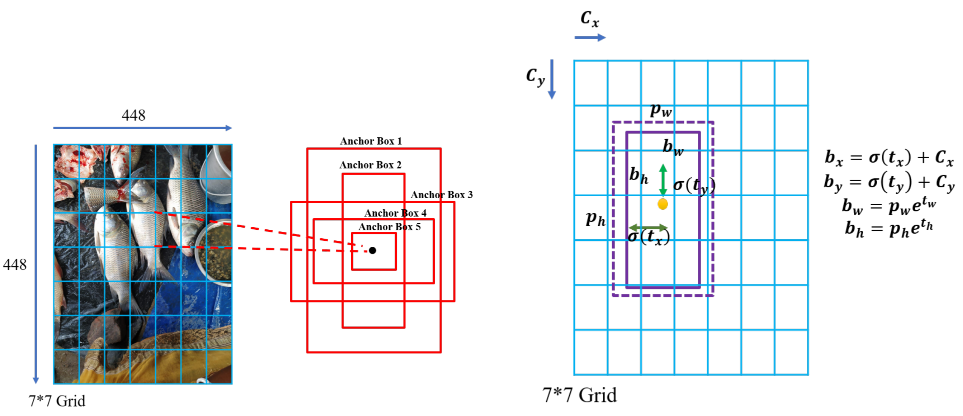
Conv	512	3× 3	26×26×512
Space_to_depth	2048		26×26×2048

Concat	3072		13×13×3072
Conv	1024	3× 3	13×13×1024
Conv	5×(5+C)	1× 1	13×13× (5×(5+C))

Figure 4.5: Architecture of YOLO-V2

and one-shot detections. But YOLO-V1 can detect a maximum of two objects per class per grid cell and limits the ability to detect the objects in nearby locations. In a cluttered environment, where many objects are present in nearby locations, YOLO-V1 can not be effective. Also, as YOLO-V1 only sees the images with a single resolution, the detection of objects with different aspect ratios is not possible. These problems are overcome by the next version of YOLO, YOLO-V2, which came with several modifications with the approximately same speed and capability of recognizing 9000 categories. YOLO-V2 is sometimes alternatively called YOLO9000 (Redmon and Farhadi [65]), as it can recognize, 9000 different classes.

YOLO-V2 (Figure 4.5) used ImageNet to pre-train the model at 224 × 224, and then again fine-tuned the model for ten epochs on ImageNet with the resolution of 448 × 448. Due to this, there is an improvement in the network performance on the high-resolution input. YOLOv2 improved upon the architecture of YOLOv1 by utilizing fully convolutional layers, which means removing the last dense layers of YOLO-V1 and replaced with the convolutional layers. The batch normalization technique was used as



(a) (b)

Figure 4.6: (a) Representation of different anchor boxes in YOLO-V2 (b) Calculation of bounding boxes coordinates in YOLO-V2

a regularizer in all convolutional layers to reduce the overfitting problem. The anchor box concept was used to predict the bounding boxes. Some prior boxes with predefined shapes were used to match the prototypical shapes of the objects. K-means clustering algorithm was used to find some good prior boxes and five(5) prior boxes were selected. The network predicted five bounding boxes per grid cell, b_x, b_y, b_w, b_h, and b_0. These values were calculated as given in Figure 4.6. In YOLO-V2, one pooling layer of YOLO-v1 was removed to obtain a 13×13 feature map. One pass through layer was used, that takes the $26 \times 26 \times 512$ feature maps and based on spatial resampling adjacent features were stacked in different channels and a feature map of 13×3072 was generated. The final two convolutional layers generate 13×1048 and $13 \times 5 \times (5 + C)$ feature maps, where C is the number of classes in the dataset. In YOLO-V2 Darknet-19 architecture was used with 19 convolutional layers and 5 max-pooling layers. To reduce the number of parameters in the network, 1×1 convolutional network between the 3×3 convolutional network.

YOLOv3

Detecting small objects needs to consider the fine-grained features. Skip connections and Feature Pyramid Networks (FPN) in the network architecture play an important role here. In YOLO-V2 Darknet-19 is used, and no skip connections are present. Small object detection is a challenging problem in YOLO-V2. The next improvement in YOLO family is YOLOV3 (Figure 4.7) which employs Darknet-53 and only 3×3 and 1 convolutional blocks are present (Redmon and Farhadi [52]). There is no max pooling layer present in YOLO-V3. As no max pooling layer is used in YOLO-V3, no information loss happens. A total of 3 anchor boxes are used in YOLO-V3, dealing with small, medium, and large-scale detection. In layers 82, 94, and 106 the large scale, medium scale, and small scale predictions are done. For each grid cell, three boxes are predicted for three different scales- 13×13, 26×26, and 52×52 with five values $(offset(x, y), scales(w, h), objectness_{score})$ and corresponding class probabilities. Considering all the scales, a total of 3549 grids

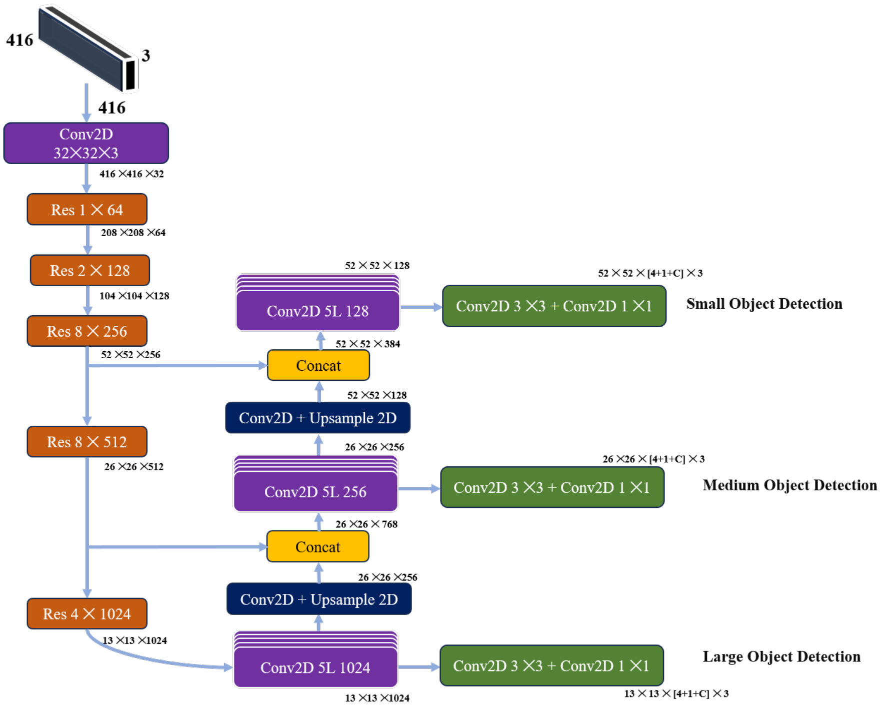


Figure 4.7: Architecture of YOLO-V3

are present and as each grid cell predicted 3 boxes, a total of 10647 boxes are predicted in YOLO-V3. Compared to YOLO-V3, YOLO-V2 predicted 845 boxes, and YOLO-V1 predicted only 98 boxes. As the number of box predictions increases, the performance of the model is also enhanced. The boxes are predicted in the same way as YOLO-V2 (see Figure 4.5). Multi-label object detection and prediction are possible in YOLO-V3. This means for a single object, multiple-label prediction is possible. Here, the summation of the confidence probabilities of all predicted labels is not equal to one. For each label or node, independent logistic classifiers are used and a binary cross entropy loss function is utilized here. The confidence loss and classification loss are updated compared to YOLO-V2 as given in Equation 4.9.

$$\begin{aligned} L_{confidence} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{object} [\hat{C}_i log(c_i) + (1 - \hat{C}_i) log(1 - C_i)] + \\ \lambda_{no\_object} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{no\_object} [\hat{C}_i log(C_i) + (1 - \hat{C}_i) log(1 - C_i)] \end{aligned} \tag{4.8}$$

$$L_{classification} = \sum_{i=0}^{S^2} 1_{ij}^{object} \sum_{Cclasses} [\hat{P}_i(c) log(P_i(c)) + (1 - \hat{P}_i(c)) log(1 - P_i(c))] \tag{4.9}$$

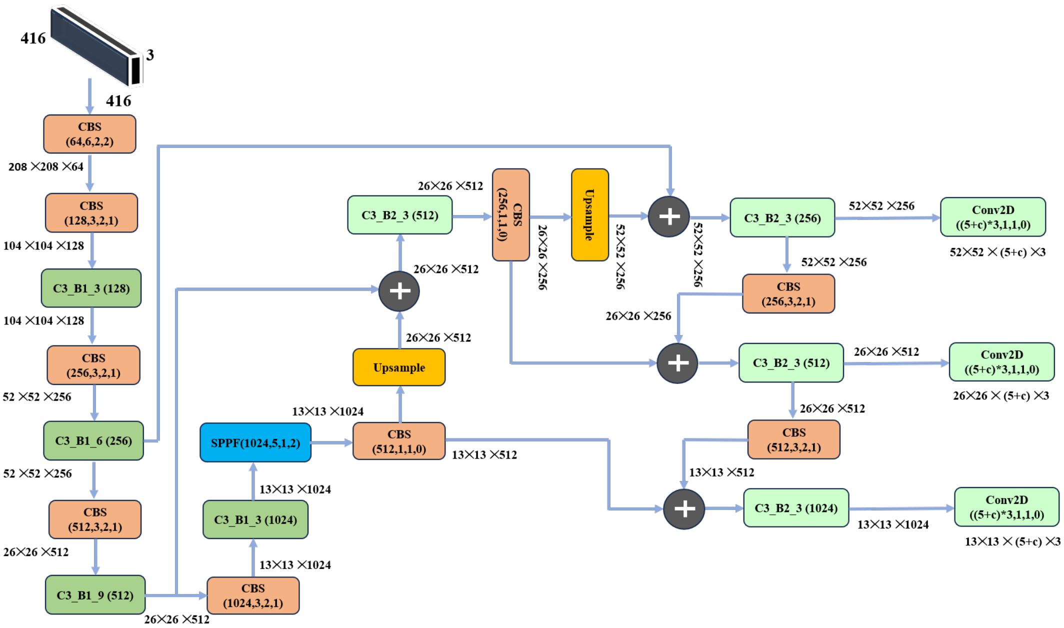
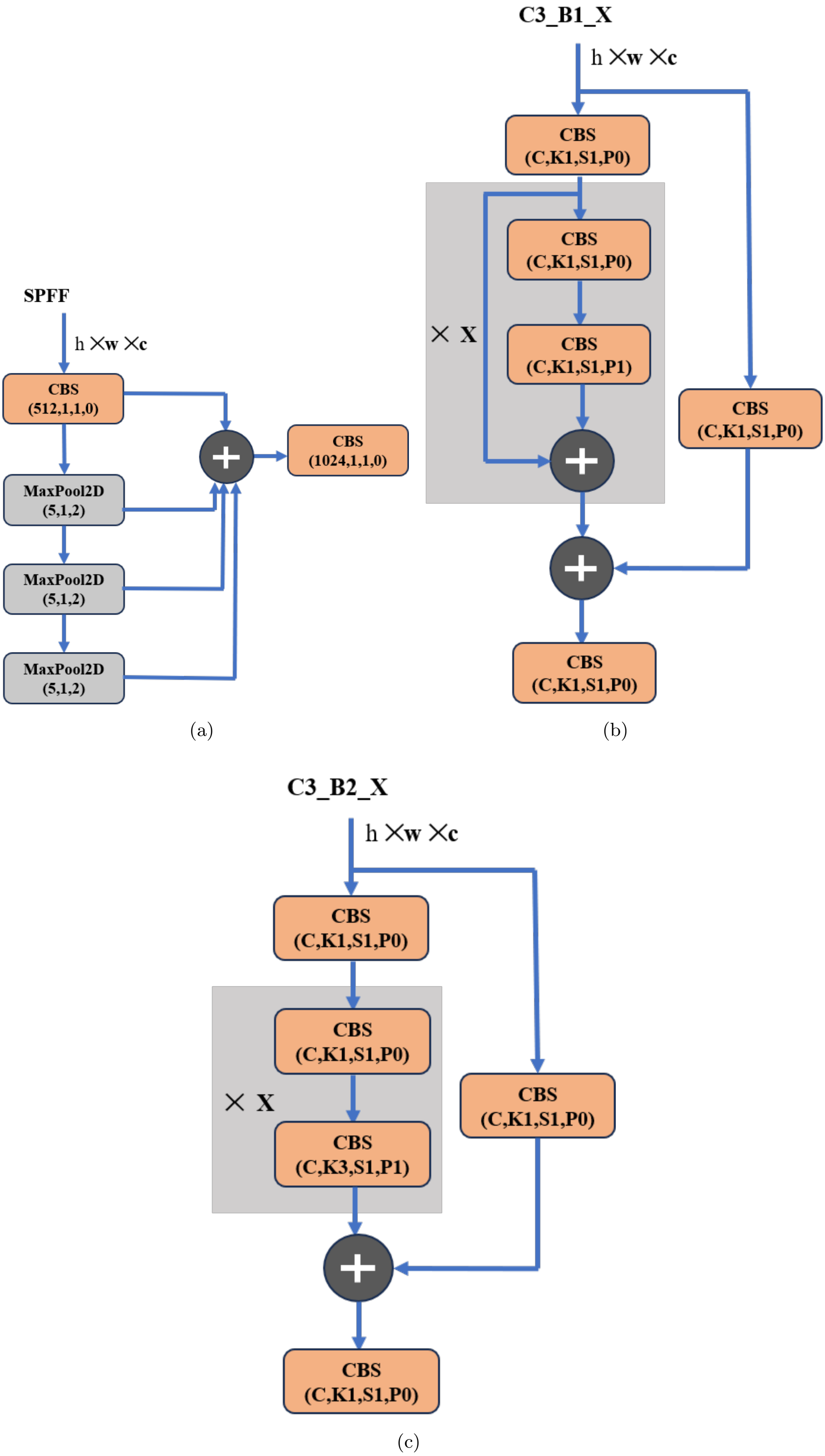


Figure 4.8: Architecture of YOLO-V5

YOLOv5

YOLOv5 is designed using PyTorch, which uses CSPDarkNet53 as its backbone (Jocher et al. [53]). The residual and dense blocks in DarkNet53 enabled the information to flow to the deep layers of the network. Due to this, redundant gradients arise and create problems. Cross Stage Partial Network (CSPNet) tackles this issue by discarding these gradient flows. Path Aggregation Network (PANet) Liu et al. [66] is used as a neck to warm up the information flow by adopting FPN, which uses several top-down and bottom-up paths. The high-level as well as the low-level features are propagated in the model. The architecture of YOLO-V5 is presented in Figure 4.8. YOLOv5 uses YOLO head, which generates three different feature maps that enable multi-scale predictions. The Focus layer replaces the first three layers of YOLOv3 and uses a single layer instead. The C3 layer consists of three convolutional layers and a single module with cascaded bottlenecks (refer Figure 4.9). Spatial Pyramid Pooling (SPP) is a pooling layer used to remove the fixed size constraint of the model, and an upsampling layer is used to upsample the previous layer's output (refer Figure 4.9). The Concat layer is used to slice the previous layer into different output paths in a given dimension. As activation function, Leaky ReLU (Maas et al. [67]) and Sigmoid (Bellman [68]) are used in YOLOv5. YOLOv5 generates the class of object detected, bounding box, and the confidence score of the object predicted. It uses Binary Cross Entropy (BCE) to compute class loss and objectness loss. The Complete Intersection over Union (CIoU) is used to calculate the location loss. The loss function used in YOLOv5 is calculated as a combination of class loss, objectness loss, and location loss.



(a) (b)

(c)

Figure 4.9: (a) SPPF architecture (b) C3 layer with Bottleneck 1 and (c) C3 layer with Bottleneck 2 architecture in YOLO-V2

4.3.3 Metrics Used

4.3.4 True Positive, True Negative, False Positive, False Negative

- **True Positive (TP)**: The case where the learned model correctly predicts the positive samples. This means that it is the instance where both the predicted and actual samples are positive.

- **True Negative (TN)**: The case where the learned model correctly predicts the negative samples. This means that it is the instance where both the predicted and actual samples are negative.

- **False Positive (FP)**: It is the case where the learned model incorrectly predicts the positive samples. This means it is the instance where the predicted sample is positive, but the actual sample is negative.

- **False Negative (FN)**: It is the case where the learned model incorrectly predicts the negative samples. It is the instance where the predicted sample is negative, but the actual sample is positive.

Intersection over Union

Intersection over Union (IoU) is a metric used to evaluate the performance of object detection algorithms, particularly in tasks such as image segmentation and object localization. It measures the overlap between predicted bounding boxes and ground truth bounding boxes.

- **Intersection (I)**: Calculate the area of overlap between the predicted bounding box and the ground truth bounding box.

- **Union (U)**: Calculate the total area covered by both the predicted bounding box and the ground truth bounding box, including the overlapping region.

- **IoU Calculation**: Divide the intersection area (I) by the union area (U).

$$IoU = \frac{Areaofoverlap}{Areaunderunion} \tag{4.10}$$

The IoU value ranges from 0 to 1, where:

- IoU = 0 means no overlap between the predicted and ground truth bounding boxes.

- IoU = 1 means the predicted bounding box perfectly matches the ground truth bounding box.

Higher IoU values indicate better object localization performance.

Average Precision

Average Precision (AP) is a widely used evaluation metric in the field of object detection and information retrieval, particularly for object detection.

- **Precision-Recall Curve**: First, the precision and recall values are computed at different thresholds for a given model's predictions.
  - Precision measures the ratio of true positive predictions to the total number of positive predictions (true positives and false positives) at a given threshold.
  - Recall measures the ratio of true positive predictions to the total number of actual positives (true positives and false negatives) at a given threshold.
- **Interpolated Precision**: To smooth out the precision-recall curve and make the evaluation more robust, interpolated precision is computed. At each recall value, the maximum precision value from that point to the end of the curve is retained.
- **Average Precision (AP)**: The average precision is calculated by taking the mean of the interpolated precision values at different recall levels.

$$AP = \frac{\sum_n (R_n - R_{n-1}).p_n}{R_{max}} \tag{4.11}$$

  Where R_n and $P - n$ are the recall and precision values at each operating point (threshold) on the precision-recall curve, and R_{max} is the maximum recall value.

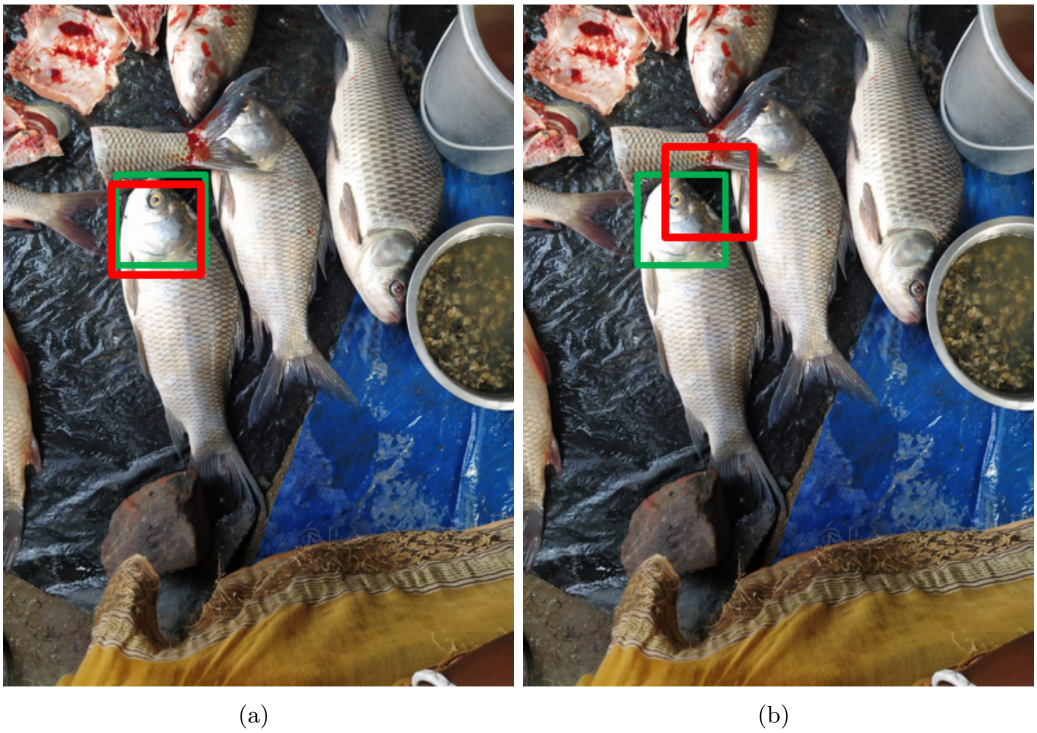
Average Recall

Average Recall (AR) is a metric used to evaluate the performance of object detection systems, particularly in tasks such as image classification and object detection.

- **Recall at Different Confidence Thresholds:**
  - For each class, the recall is computed at different confidence thresholds.
  - Recall measures the ratio of true positive predictions to the total number of actual positives (true positives and false negatives) at a given threshold.
- **Average Recall (AR)**:
  - The Average Recall is calculated by averaging the recall values across different classes.

$$AR = \frac{1}{N} \sum_{i=1}^{N} \frac{\sum_{c=1}^{C} TP_{i,c}}{\sum_{c=1}^{C} TP_{i,c} + FN_{i,c}} \tag{4.12}$$

    Where N is the total number of classes, C is the number of classes, $TP_{i,c}$ is the number of true positive detections for class c at confidence threshold i, and $FN_{i,c}$ is the number of false negatives for class c at confidence threshold i.



(a)

(b)

Figure 4.10: True positive and False positive case in object detection work (a) True positive case with IoU of 0.85 (b) False positive case with IoU of 0.35

Mean Average Precision

- **Precision**: It is calculated as the number of true positive results divided by the number of all positive results, which means the summation of true positives or false positives.

$$Precision = \frac{TP}{TP + FP} \tag{4.13}$$

  In object detection tasks, precision is calculated at some given IoU threshold. In Figure 4.10(a), TP is represented with an approximate IoU of 0.85 with respect to ground truth, and in Figure 4.10(b), FP is represented with an approximate IoU of 0.35 with respect to ground truth.

- **Recall**: It is calculated as the number of true positive results divided by the number of all predictions, which means the summation of true positives or false negatives.

$$Recall = \frac{TP}{TP + FN} \tag{4.14}$$

Mean Average Precision(mAP) is simply the mean of Average Precision(AP) across multiple classes. The AP is calculated as discussed in the subsection 4.3.4. In the object detection task, the mAP is calculated by finding the AP for each class and taking

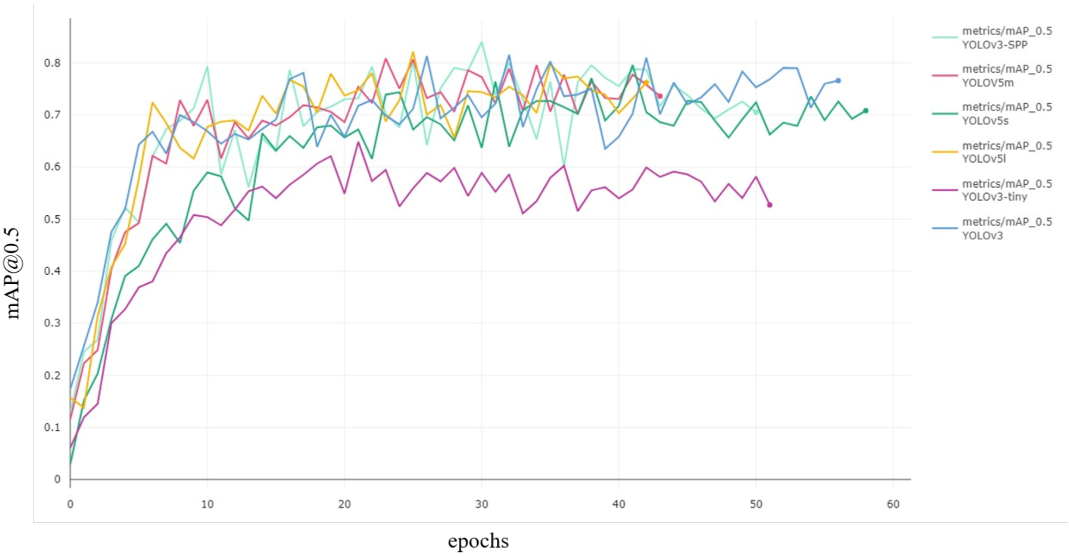


Figure 4.11: mAP@0.5 value versus epochs for different deep learning object detection models used on **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset (Banerjee et al. [3])

an average over a number of classes.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{4.15}$$

mAP is particularly useful in tasks where there are multiple classes or categories, and the performance of the model needs to be evaluated across all of them. It provides a comprehensive assessment of the ability of the model to correctly identify relevant instances across various levels of confidence.

4.3.5 Experiment Protocols and Results

At first, the YOLOv3, YOLOv3-tiny, and YOLOv3-SPP networks were applied to the original dataset and augmented dataset individually. Before feeding the images to the YOLO network, they were resized into 416 × 416. The dataset.yaml is the key file to start the training process. The training, testing, and validation paths were set to the specific directory, and the number of classes was set to 6. Using a stochastic gradient descent optimization function with a momentum of 0.9, a learning rate of 0.001, and a weight decay of 0.005, the networks were trained for 200 epochs with a patience value of 20. It means that if there is no improvement in the mAP value after 20 epcohs, then the training process is stopped. When tested on the original dataset, the mAP@0.5 values of 0.678, 0.522, and 0.764 were obtained on the validation dataset by YOLOv3, YOLOv3-tiny, and the YOLOv3-SPP network, respectively. Next, the augmented dataset was used in the training phase with the same experimental parameter configuration. In the augmented dataset, mAP@0.5 values of 0.813, 0.65, and 0.84 were achieved on the validation dataset by YOLOv3, YOLOv3-tiny, and the YOLOv3-SPP network, respectively. The best mAP@0.5 values of 0.764 and 0.84 were achieved on the original and augmented datasets

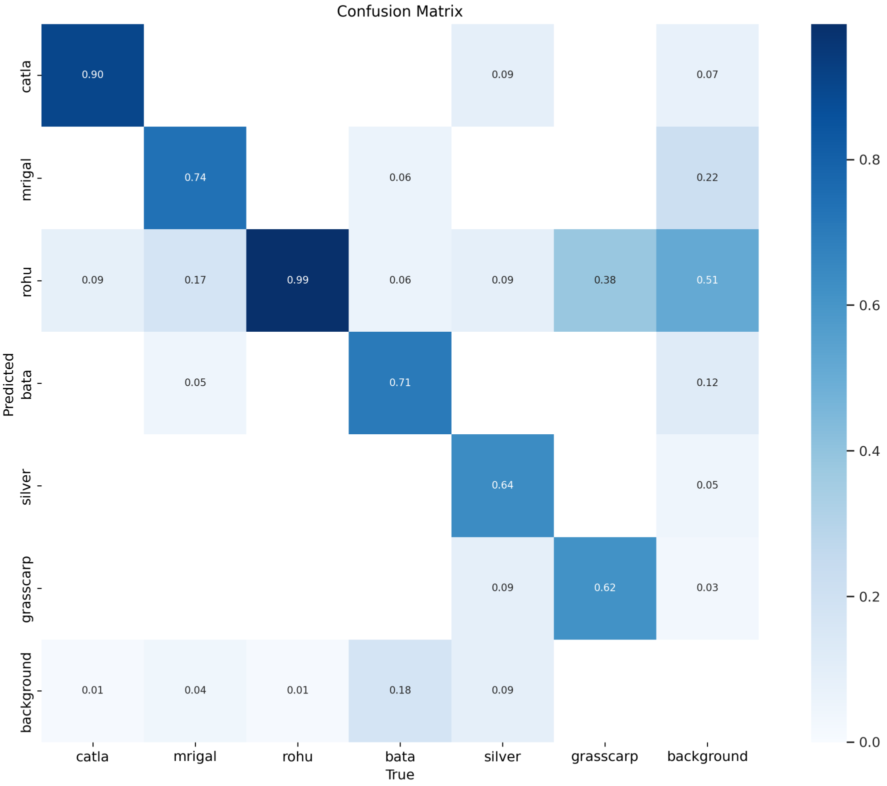


Figure 4.12: Confusion matrix of best performing YOLOv3-SPP model on **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset (Banerjee et al. [3])

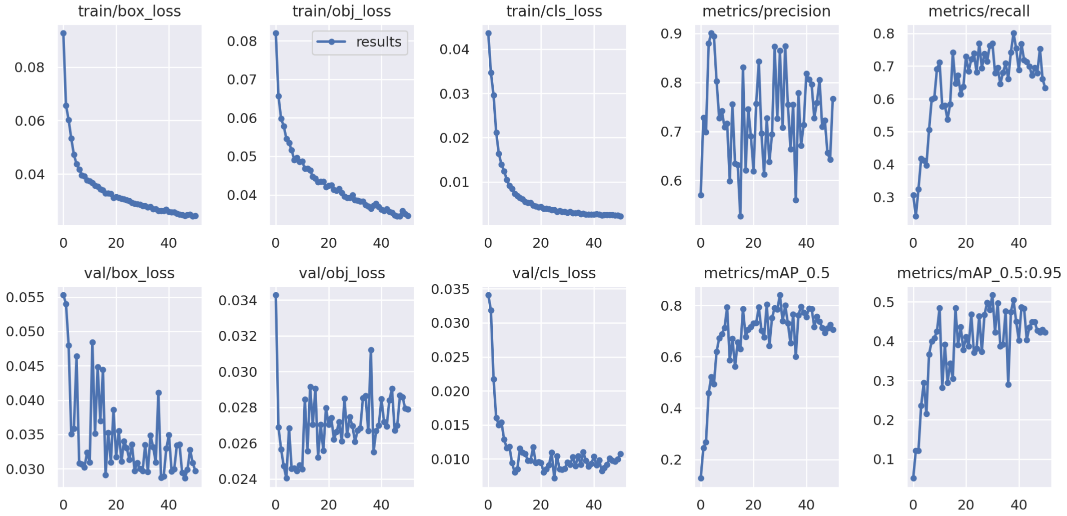


Figure 4.13: Results of best performing YOLOv3-SPP model on **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset (Banerjee et al. [3])

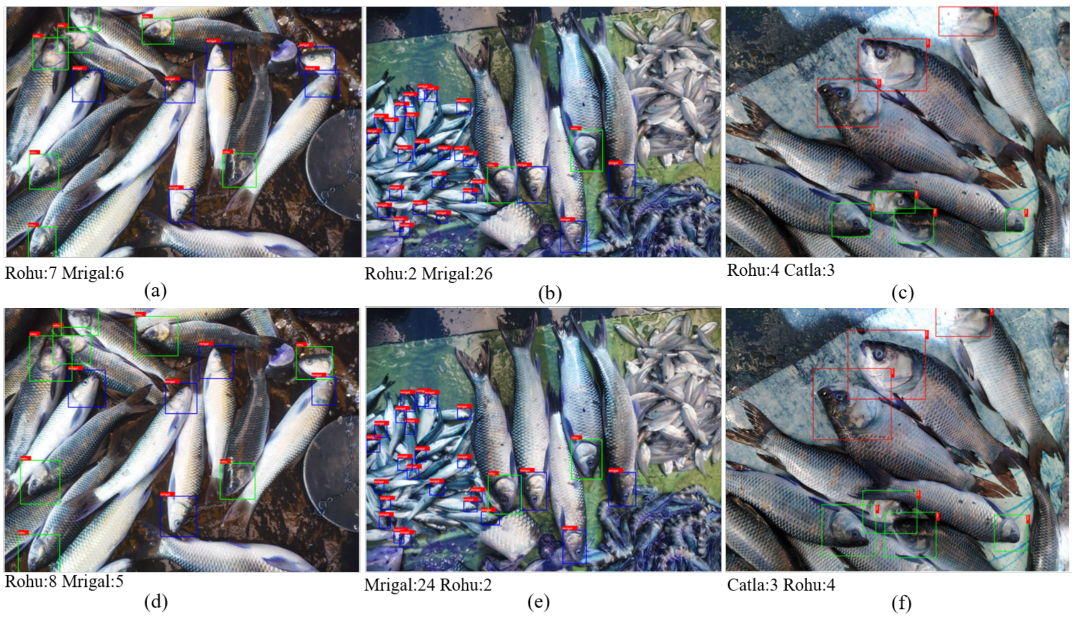
by YOLOv3- SPP. YOLOv5l, YOLOv5m, and YOLOv5s networks were applied in the same way as the YOLOv3 versions to the original and augmented datasets. When using the original dataset, mAP@0.5 values of 0.763, 0.668, and 0.536 were achieved by the YOLOv5l, YOLOv5m, and YOLOv5s networks, respectively. The same set of experiments was done on the augmented dataset. The mAP@0.5 values of 0.78, 0.77, and 0.81 were achieved on the validation dataset by YOLOv5l, YOLOv5s, and Yolov5m, respectively. In the augmented dataset, best mAP@0.5 of 0.763, and 0.81 was achieved by YOLOv5l, and YOLOv5m model. The YOLOv3-SPP model achieved the best mAP@0.5 of 0.84 when the augmented dataset was used. The class-wise precision, recall, and mAP@0.5 values on the augmented dataset are shown in Table 4.1 and overall precision, recall, and mAP@0.5 value of different models applied on the **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset is presented in Table 4.2. The mAP@0.5 value vs. epochs graph for different models used here is shown in Figure 4.11. For the species *Labeo catla*, *Labeo rohita*, and *Cirrhinus mrigala* the mAP@0.5 value was high. The mAP@0.5 of 0.677 for *Labeo bata* was the lowest compared to other classes. The head portion of *Labeo bata* and the small size *Cirrhinus mrigala* are very similar, and even the trained personnel have problems recognizing them properly. Also, the number of instances of *Labeo bata* was very low compared to *Cirrhinus mrigala.* Though the number of instances of *Hypophthalmichthys molitrix* and *Ctenopharyngodon idella* species were very few,nthe head part of these species are very different compared to the other species used. In some cases, it is difficult to differentiate between large size *Labeo rohita* and *Ctenopharyngodon idella* using the head part. The confusion matrix of the YOLOv3-SPP model is shown in Figure 4.12. In Figure 4.13 the class loss, box loss, precision, recall, and mAP of the training and validation dataset is presented. Some of the ground truth annotations and prediction results, along with the number of instances for each fish species, are shown in Figure 4.16. From the confusion matrix, it is clear that 17% *Cirrhinus mrigala* was misclassified as *Labeo rohita*, and 38% *Ctenopharyngodon idella* was misclassified as *Labeo rohita.* Majority of the time, the bounding box miss, which means ground truth bounding boxes are detected as background occurs in *Labeo bata* species. A total of 38% *Labeo bata* was misclassified as background.

Table 4.1: Class wise precision, recall, and mAP@0.5 value for best performing YOLOv3-SPP model on **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset (Validation dataset consists of 116 images) (Banerjee et al. [3])

	Precision	**Recall**	**mAP@0.5**
Laebo catla	0.958	0.89	0.962
Cirrhinus mrigala	0.871	0.722	0.843
Labeo rohita	0.88	0.937	0.94
Labeo bata	0.814	0.588	0.677
Hypophthalmichthys molitrix	0.812	0.788	0.823
Ctenopharyngodon idella	0.844	0.692	0.797

Table 4.2: Precision, Recall, and mAP@0.5 value of different models used on **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset (Banerjee et al. [3])

Model	Precision	Recall	mAP@0.5
YOLOv3	0.841	0.727	0.813
YOLOv3-tiny	0.774	0.699	0.78
YOLOv3-SPP	0.863	0.77	0.84
YOLOv5s	0.803	0.733	0.771
YOLOv5m	0.785	0.736	0.808
YOLOv5l	0.774	0.699	0.78



(a) (b) (c)

(d) (e) (f)

Figure 4.14: Some ground truth annotations and prediction results using the proposed technique: (a, b, c)- represent ground truth, (d,e,f)- represent prediction (Banerjee et al. [3])

4.3.6 Conclusion

This study used different variants of YOLOv3 and YOLOv5 to automatically recognize fish species and count the number of instances of each species in the cluttered environments, typically found in the fish market. A total of six species named *Labeo catla*, *Labeo rohita*, *Cirrhinus mrigala*, *Labeo bata*, *Hypophthalmichthys molitrix*, *Ctenopharyngodon idella* were considered for recognition and counting purposes. In live fish markets, different fish species are crammed into a small space. As a result, it is difficult to identify the species by following the taxonomy of the entire fish body. An augmented dataset named **JUDVLP-WBUAFS**: Fishdb-Detection.v1 consisting of 2,281 images with 19,995 instances was prepared and annotated only the fish heads using bounding boxes for detection and recognition of the fish species. YOLOv3, YOLOv3-tiny, and YOLOv3-SPP networks under the YOLOv3 family have been used, and the best mAP@0.5 of 0.764 and 0.84 was generated on the original and augmented dataset. Also, YOLOv5l, YOLOv5m, and YOLOv5s networks under the YOLOv5 family have been applied, and the best mAP@0.5 of 0.764 and 0.81 was achieved on the original and augmented dataset. With a mAP@0.5 of 0.84, YOLOv3-SPP outperformed all other models used in this study. Though the overall mAP value achieved is encouraging, there were some misclassifications present, mainly in the *Labeo bata* and *Ctenopharyngodon idella.* The number of instances of *Labeo bata*, and *Ctenopharyngodon idella* were very small, and the size of *Labeo bata* was very small compared to other species. This study on the recognition and counting of fish species in live fish markets benefits both the common people and various fisheries industries. The inclusion of more fish species will increase the number of instances for the minority classes. Also, reducing the size of the model and its deployment in a real-world scenario, which is an open research area in this field, will go a long way.

4.4 Segmentation of Freshwater Fish Species in Cluttered Environment

4.4.1 Dataset Preparation

No standard dataset was available that can be used for fish species recognition based on semantic segmentation. A dataset containing 200 images of live fish market data was created. Five fish species named: Catla(*Labeo catla*), Rohu(*Labeo rohita*), Mrigal(*Cirrhinus mrigala*), bata(*Labeo bata*), and Silver carp(*Hypophthalmichthys molitrix*) were considered for the automatic recognition in a cluttered environment. The primary goal was to compare the performance with the fish species recognition and counting framework, only the fish heads were annotated using the LabelMe tool. Different fish species were annotated with different colors. The color schemes (R, G, and B) of the ground truth masks were: *Labeo catla*– (0, 128, 0), *Labeo rohita*– (128, 0, 0), *Cirrhinus mrigala*– (0, 128, 128), *Labeo bata*- (0, 0,128), *Hypophthalmichthys molitrix*- (128, 0, 128). There were other fish species present in the images except the five above-mentioned fish

Species	**Count**
Labeo bata	416
Labeo rohita	5984
Hypophthalmichthys molitrix	272
Labeo catla	1248
Cirrhinus mrigala	1728

Table 4.3: Number of samples per fish species in the training dataset

species. During the annotation process, all these species were not annotated. The entire process of dataset preparation and annotation is described in section 2.3. The dataset was divided into training, and testing parts in a 60:40 ratio. Further, training part was divided into training and validation set in a 75:25 ratio. As the dataset was small for deep learning problems, different augmentations were applied to the training dataset. Rotation (90°, 180°), horizontal flip, vertical flip, intensity rescaling, gamma correction, sigmoid correction, and logarithmic correction augmentations were applied. These augmentations were done both on ground truth images and ground truth masks. After the augmentation, the dataset consisted of 720 images in the training set. Table 4.3 provides the number of samples for each fish species.

4.4.2 Methodology

Image Segmentation

Image segmentation is the process of partitioning the image into different groups based on some common characteristics. The goal is to group together pixels that belong to the same object or share similar characteristics, such as color, texture, or intensity. Image segmentation plays a crucial role in various applications, including object detection and recognition, medical image analysis, autonomous driving, and satellite image processing. Broadly, there are two types of image segmentation: semantic segmentation and instance segmentation. Semantic segmentation involves labelling each pixel in an image with a class label, without distinguishing between different instances of the same class. Every pixel in the image is assigned a single class label and represented by a single color. Instance segmentation, on the other hand, not only labels each pixel with a class but also distinguishes between different instances of the same class. Different instances of the same object are assigned different colors. Classification is the coarse inference where a single class label is assigned to an image, and segmentation is the fine inference with dense predictions with labels for each pixel in the image. In this work, semantic segmentation was used to recognize the fish species in some cluttered environments.

Proposed Approach

Image segmentation is a process of partitioning the image into multiple sections and locating objects of interest. This study uses Semantic segmentation, a deep learning technique, which assigns a class label to each pixel of the image. The number of class

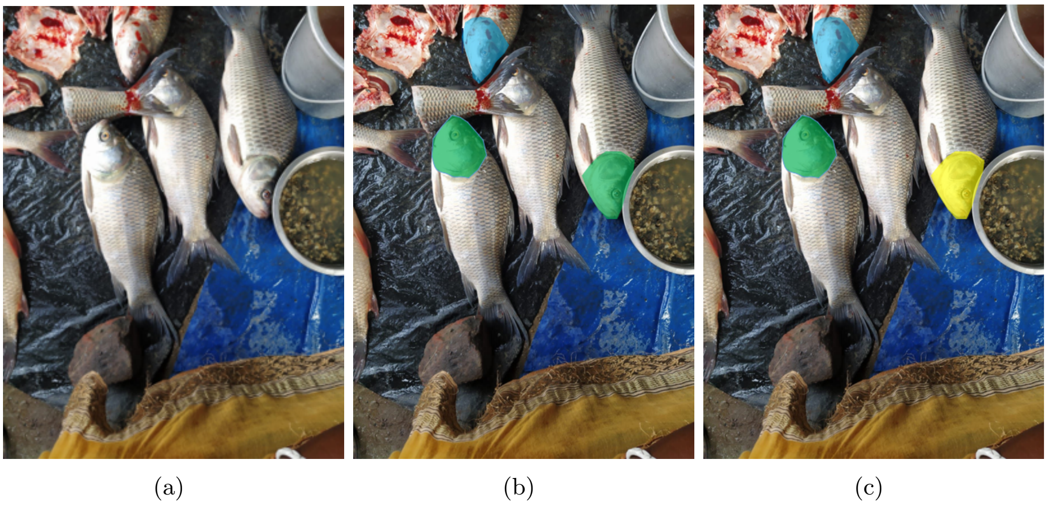


Figure 4.15: Representation of image segmentation (a) Original image (b) Semantic segmentation (c) Instance segmentation

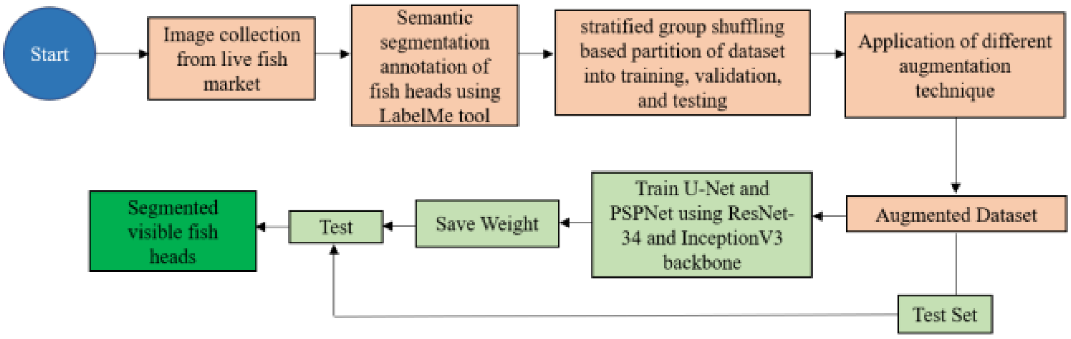

Start
Image collection from live fish market
Semantic segmentation annotation of fish heads using LabelMe tool
stratified group shuffling based partition of dataset into training, validation, and testing
Application of different augmentation technique
Augmented Dataset
Train U-Net and PSPNet using ResNet-34 and InceptionV3 backbone
Save Weight
Test
Segmented visible fish heads
Test Set

Figure 4.16: Flow diagram of fish species recognition in some cluttered environment using semantic segmentation (Banerjee et al. [4])

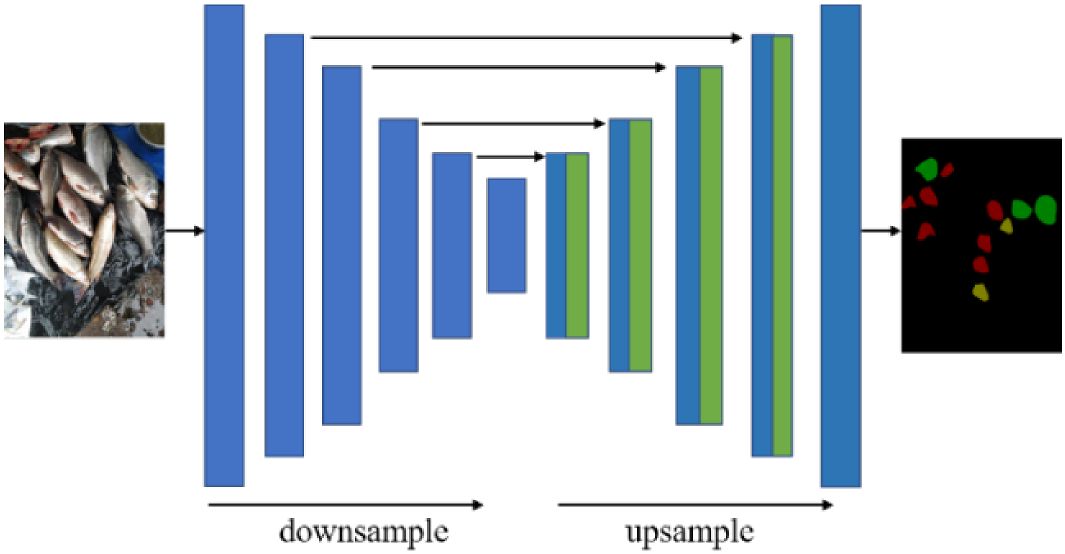


Figure 4.17: Block diagram of U-Net architecture used in this study (Banerjee et al. [4])

labels is equal to the number of different objects present in the image. This study aims to segment a total of five fish species, and the semantic segmentation of the fish heads involves the presence of five class labels. Two popular segmentation architectures based on deep learning, named UNet by Ronneberger et al. [69] and Pyramid scene parsing network (PSPNet) by Zhao et al. [70], were used. U-Net is the encoder-decoder based architecture popularly used in medical image segmentation problems. PSPNet is another encoder decoder network that uses the global context of the image to predict the local context. To train these networks from scratch, a substantial amount of data is required. Also, training a large dataset from scratch needs a lot of time. Some pre-trained CNN backbones were used in the encoder part to extract meaningful features. The encoder part of the network finds latent low-dimensional information, and the decoder part decodes this information back to the original shape of the input with different segmented labels. Two different pre-trained backbone networks on the ImageNet dataset, named ResNet, and InceptionV3 were used in the experiments as encoder networks. In Figure 4.16 the flow diagram of the proposed methodology is presented.

U-Net: U-Net is a deep learning semantic segmentation architecture with encoder and decoder parts. The encoder part encodes the information into a lower dimension, and the decoder part decodes it to the original input dimension with class labels, thus making a U-shape architecture. U-Net uses skip connections to transmit information from the encoder part to the decoder part at various convolution levels. In the initial layers, the encoder part finds lower-level details in the sample, and gradually, in the deeper layer, the high-level features are encoded. To segment the present objects of interest in an image, both low-level and high-level feature maps play a crucial role. Hence, the skip connections between the encoder and decoder part enable to use feature map of the encoder part from different levels and concatenate with the decoder part to use the global and local

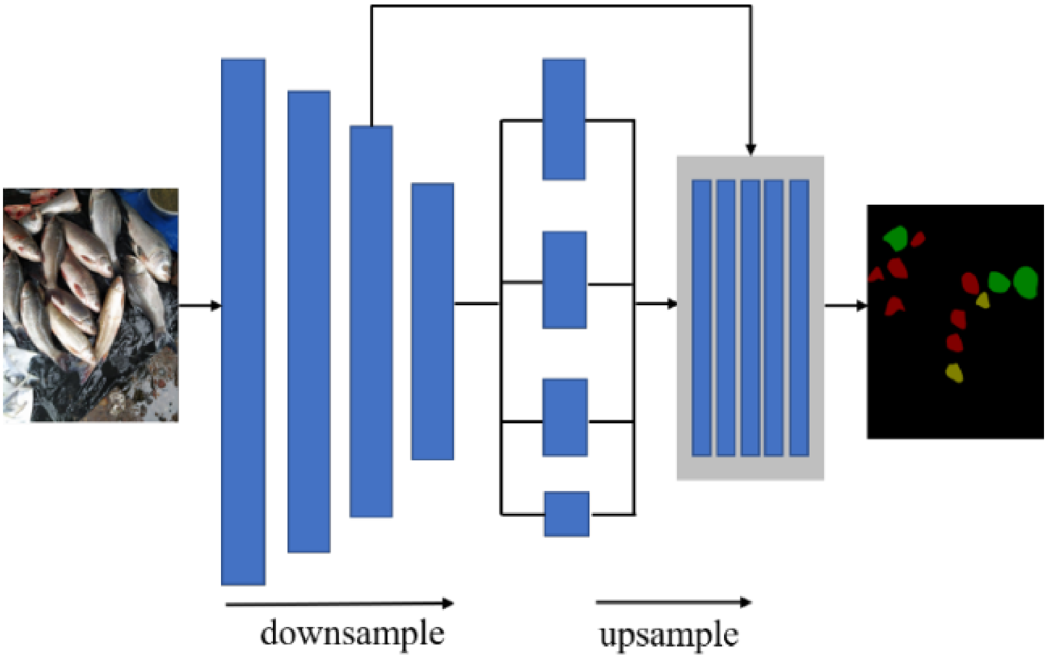


Figure 4.18: Block diagram of PSPNet architecture used in this study (Banerjee et al. [4])

information. The block diagram of the U-Net used in this study is presented in Figure 4.17.

PSPNet: PSPNet (Figure 4.18) is an encoder-decoder network with a CNN backbone and dilated convolutions. The pyramid pooling module is the main part of PSPNet, enables to use the global information in the image. The global information, along with the local context, helps to assign a class label to each pixel in the image. Pooling is done at different sizes on the feature map generated by the encoder part, and then it passes through the convolution layer. Upsampling is done on these pooled features to map them to the original feature map. Then, the upsampled features are combined with the original feature map and passed to the decoder part of the architecture. Overall, context aggregation is done because features are fused at different levels.

Loss Function: In the **JUDVLP-WBUAFS**: Fishdb-Segmentation.v1 dataset, data imbalance is present. The majority classes are *Labeo catla*, *Labeo rohita*, and *Cirrhinus mrigala*. The minority classes are *Labeo bata* and *Hypophthalmichthys molitrix*. To handle this imbalance, a combination of dice loss (Sudre et al. [71]) and focal loss (Lin et al. [63]) was used. The dice loss function is described in Eqn. 4.16.

$$diceloss = 1 - \frac{2\sum_{i=1}^{N} p_i g_i}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} g_i^2} \tag{4.16}$$

$$focalloss = -\sum_{i=1}^{N} (1 - p_i)^{\mu} log_b(p_i) \tag{4.17}$$

Here the p_i, and g_i represent the pixel values of predictions and ground truth accordingly.

Dice loss is the same as Intersection over Union (IoU) and ranges between 0 and 1. Focal loss assigns more weight to the complex sample and less weight to the simple one. It uses a down weighting technique to reduce the effect of easy samples on the loss function, and increase the attention on the complex sample. This technique effectively addressed the imbalance problem in the dataset. Equation. 4.17 describes the focal loss function. Here, µ is the focusing parameter used to adjust the attention on the samples during training.

4.4.3 Metrics Used

The segmentation mask predicted by the proposed technique was evaluated by IoU because IoU uses intersection over union between the ground truth and predicted masks. The proposed technique predicts the test image and generates segmentation masks for the fish heads that are clearly visible. IoU is calculated between predicted masks and ground truth masks. In Eqn. 4.18, the formula for calculating IoU is given. A higher value of IoU indicates that the deep learning techniques have properly generated the segmentation masks. Here, GT represents the ground truth label mask, and PRED represents the predicted label mask. Also, the class-wise IoU can be calculated using Eqn. 4.19. The mean IoU given in Eqn. 4.20 is the final measure to evaluate the performance of the proposed techniques on the proposed dataset.

$$IoU = \frac{GT \cap PRED}{GT \cup PRED} \tag{4.18}$$

$$IoU_c = \frac{TP_c}{TP_c \cup FP_c \cup FN_c} \tag{4.19}$$

$$meanIoU = 1/C \sum_c IoU_c \tag{4.20}$$

Table 4.4: Mean IoU per class using U-Net and PSPNet with ResNet34 and InceptionV3 backbone network (Banerjee et al. [4])

Network	U-Net		PSPNet	
Backbone	ResNet-34	InceptionV3	ResNet-34	InceptionV3
Labeo bata	0.80	0.96	0.99	0.99
Labeo rohita	0.84	0.87	0.92	0.91
Hypophthalmichthys molitrix	0.96	0.95	0.98	0.90
Labeo catla	0.77	0.85	0.91	0.89
Cirrhinus mrigala	0.66	0.69	0.86	0.85

4.4.4 Experiment Protocols and Results

Deep learning plays an important role in the semantic segmentation of the objects in the image. This study employed U-Net and PSPNet, along with a pre-trained CNN, to form the foundation of the encoder segment. The ground-truth segmentation mask was prepared using the LabelMe tool. Two popular pre-trained CNNs on ImageNet, named

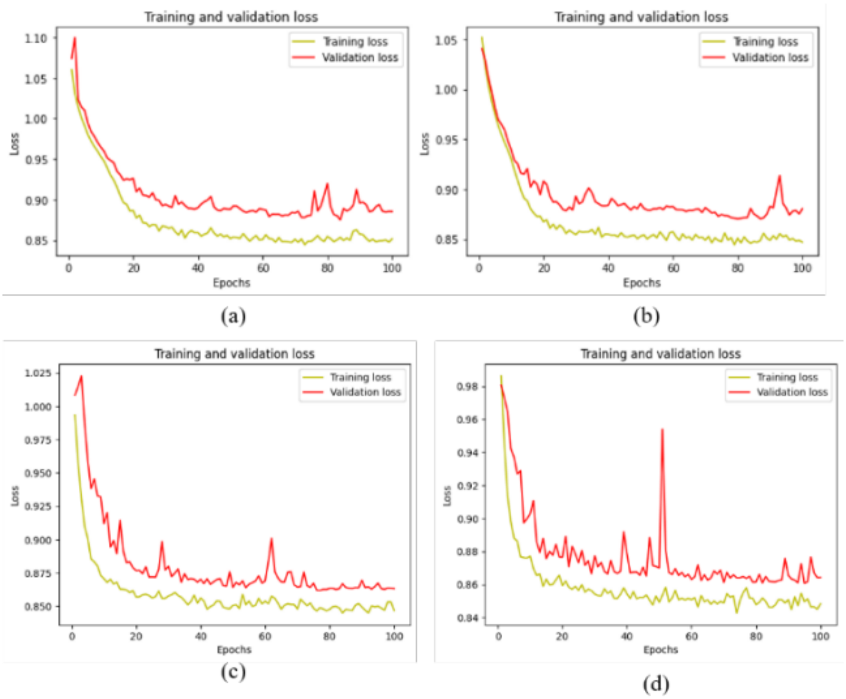
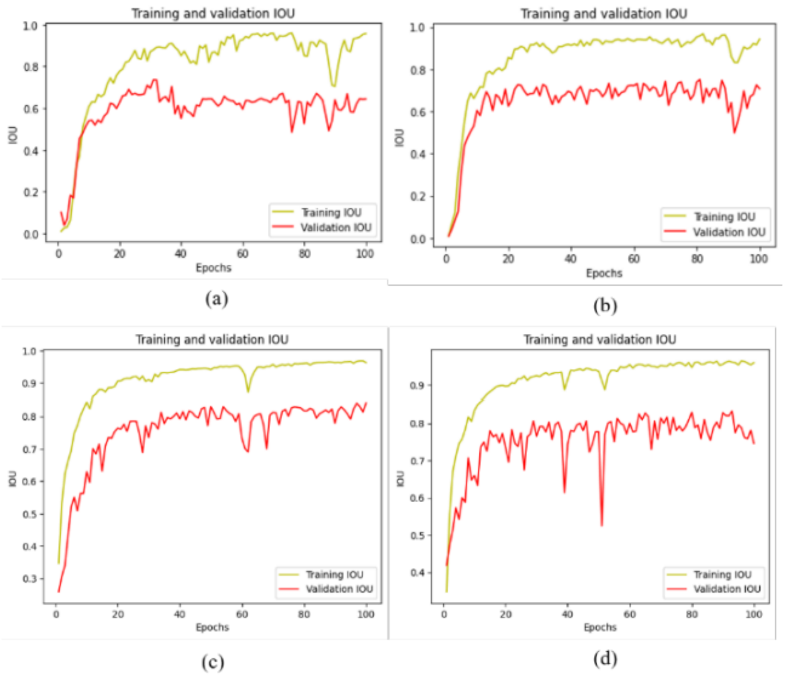


Figure 4.19: Loss graph over the epochs: (a) U-Net with ResNet34 backbone (b)U-Net with InceptionV3 backbone (c) PSPNet with ResNet34 backbone (d) PSPNet with InceptionV3 backbone (Banerjee et al. [4])

ResNet34 and InceptionV3, were used as the backbone in both U-Net and PSPNet. Due to the imbalance in the dataset, class-wise weights were calculated. The background class was given the minimum weight because it contains the major section in many of the images. These class weights were used to calculate the dice loss, and finally, it was combined with focal loss. The IoU metric was used as the evaluation metric to measure the quality of the predictions by U-Net and PSPNet. The Adam optimizer was used with an initial learning rate of 0.0001. A softmax activation function was used in this multilabel segmentation problem. All the experiments were executed for 100 epochs with a batch size of 8. Using ResNet34 and InceptionV3 in U-Net, an IoU of 0.665 and 0.697 was achieved. When the same backbone was used in PSPNet, a mean IoU of 0.76, and 0.747 was achieved. The loss values over the epochs for all the experiments are presented in Figure 4.19. The validation loss is decreasing almost in the same fashion as the training loss. Only in PSPNet with the InceptionV3 backbone, a big spike in validation loss around 50 epochs is seen, but after that, it again minimizes the loss. The mean IoU score over the epochs is presented in Figure 4.20. The best mean IoU of 0.76 is achieved by PSPNet with a ResNet34 backbone. Class-wise mean IoU is given in Table 4.4. The mean IoU of *Cirrhinus mrigala* is lower compared to other classes because there are more variations in size in this category. Though the number of samples of *Hypophthalmichthys molitrix* are less compared to other classes, the IoU score is better because the head shape for this category is different from the other fish species. The mean IoU of *Labeo catla*, and *Labeo rohita* is quite good because the number of samples in these categories is higher than in



(a)

(b)

(c)

(d)

Figure 4.20: Mean IoU graph over the epochs: (a) U-Net with ResNet34 backbone(b) U-Net with InceptionV3 backbone (c) PSPNet with ResNet34 backbone (d) PSPNet with InceptionV3 backbone (Banerjee et al. [4])

other classes. Some of the predictions made by these networks are shown in Figure 4.21. In Figure 4.22, some wrong predictions are presented. Different viewing angles, zoom effects, and size variations make it difficult to distinguish some of the fish species.

4.4.5 Conclusion

The study aims to propose a semantic segmentation of five different fish species named *Labeo catla*, *Labeo rohita*, *Cirrhinus mrigala*, *Labeo bata*, and *Hypophthalmichthys molitrix* in a cluttered environment. The fish heads are segmented automatically using two deep learning-based segmentation networks, U-Net and PSPNet. A dataset named **JUDVLP-WBUAFS**:Fishdb-Segmentation.v1 is proposed with around 800 images taken in some cluttered environment, with 200 images annotated. Using different augmentations, a training dataset with 720 images is prepared. The minority classes in the dataset are *Labeo bata* and *Hypophthalmichthys molitrix*. The proposed system is tested on 80 images, which are not used at the time of training. Two popular deep-learning semantic segmentation architectures, U-Net and PSPNet were used with ResNet34, and InceptionV3 pre-trained backbone in the encoder part. The PSPNet with ResNet34 backbone achieved the best mean IoU of 0.76. The results are promising in the context of a cluttered environment. The segmentation results could be utilized to count the number of fish species and sort different fish species. The fishery industries could use these applications for the recognition and counting of fish species in the minimum amount of time. There is a scope to explore different state-of-the-art deep learning networks in other domains here. In the future, the dataset size could be increased by labelling the rest of the images, and

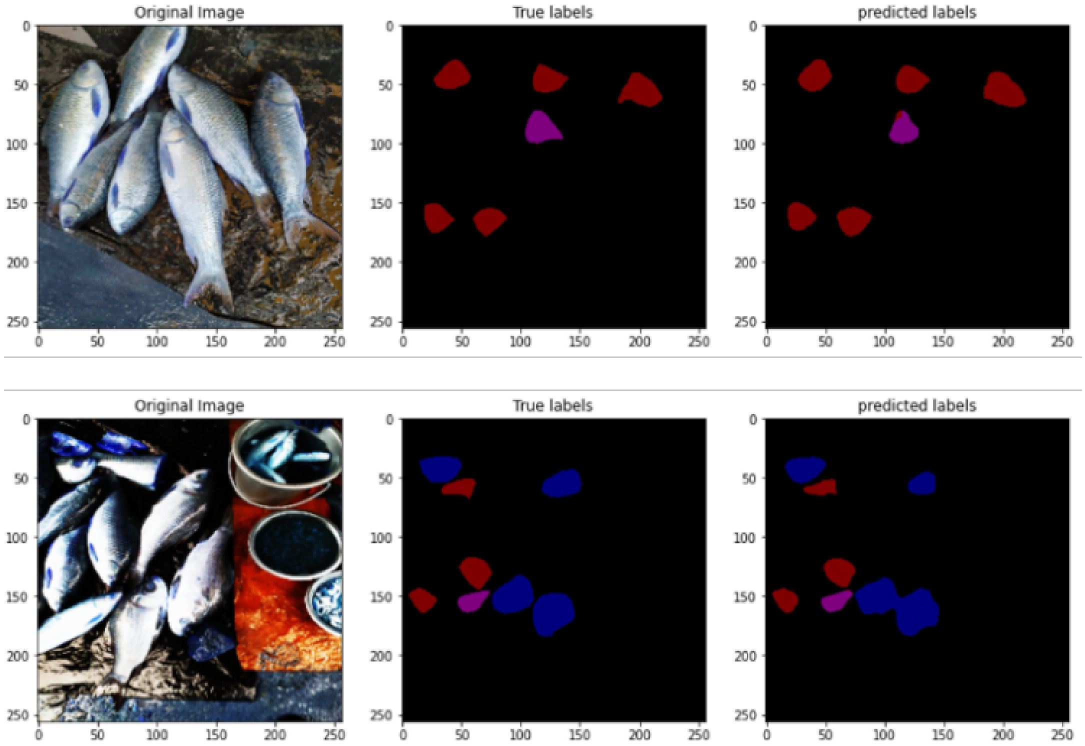


Figure 4.21: Original image, ground truth segmentation, and predicted segmentations (Banerjee et al. [4])

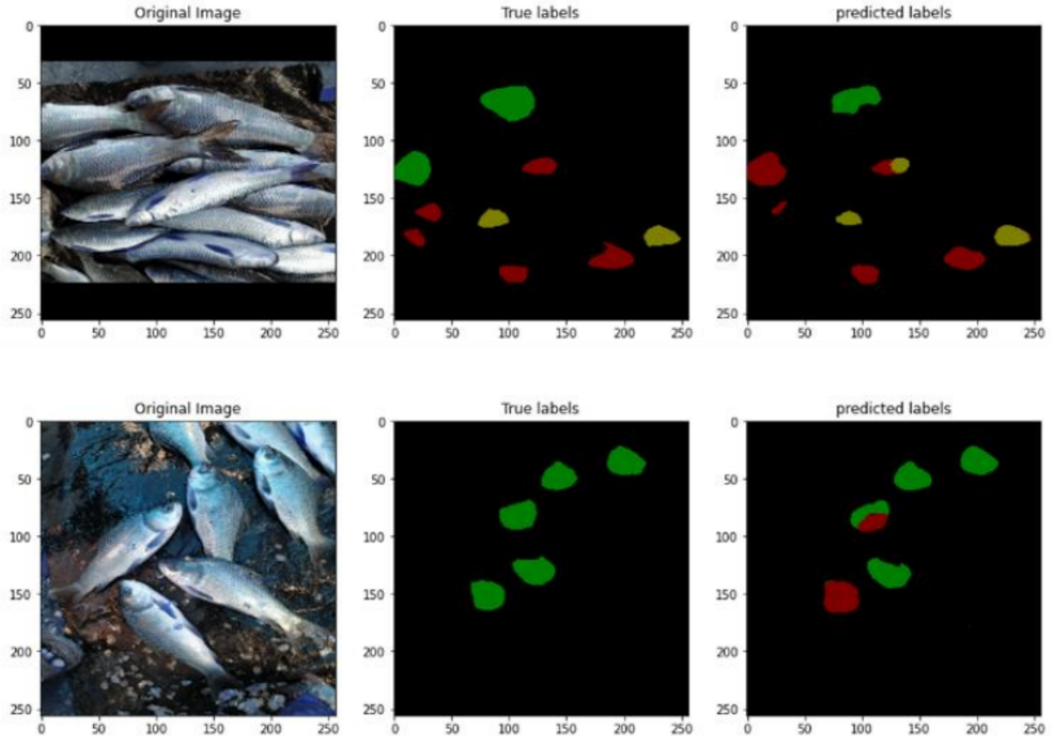


Figure 4.22: Original image, ground truth mask, and some wrong prediction mask (Banerjee et al. [4])

the proposed system could be retrained with the learned weights. Also, a variety of fish species could be included in the dataset to make it more challenging.

4.5 Summary

Recognition of fish species in some cluttered environments, like in the live fish markets, can be challenging due to various factors such as variability in fish appearance, varying lighting conditions, and the presence of other objects and people. Purchasers in fish markets face difficulties in identifying fish in such cases because the full fish body is not visible. The proper knowledge of fish taxonomy is essential here to recognize fish from the minimal view regions. This chapter presents some machine learning techniques to automatically recognize fish species in live fish markets. Only the fish heads are utilized in these approaches to identify fish because it is the only part of the fish that is visible across maximum samples in the dataset. Six fish species are recognized with a mean IoU@0.5 score of 0.84, using an approach based on object localization and identification. Here, each fish head in the image is annotated using a rectangle, and the fish is recognized from the meaningful visual features extracted from the head only. A semantic segmentation technique is applied to segment the fish head from the background and other objects in the image to focus the recognition process on the fish head. A separate dataset is prepared for this task, and after applying some semantic segmentation algorithms, the mean IoU@0.5 0.76 is achieved. Several augmentations are applied in the training part of the dataset to increase the size. The transfer learning strategy is used in both approaches to transfer knowledge from a large dataset and use it in the end task. With the success of these approaches in recognizing fish in some real cluttered environments found in fish markets, purchasers can take the benefits by using the automatic tool to do so in their busy time in the market without taking help from someone. A case study on the development of a mobile application is discussed in the next chapter that can be used to recognize fish in the market that meet the purchaser's nutritional requirements.

Chapter 5

Application of Fish Species Recognition in Cluttered Environment: Case Study

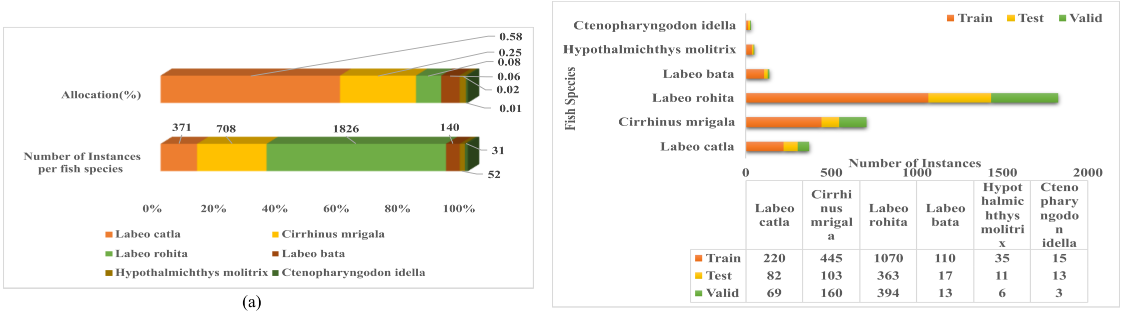
Predominantly , six nutrients are present in almost all food items: proteins, carbohydrates, fat, vitamins, minerals, and water. Along with proteins, carbohydrates, and fat, sufficient vitamins and minerals are essential for the body to function properly and strengthen the immunity system. Vitamins and minerals must be taken through foods because the human body cannot produce these internally. A minimal amount of these two micro-nutrients is needed in the body, but too little and too much intake may lead to severe health conditions. Fish is one of the major food sources rich in proteins, vitamins (particularly D and B12), minerals, and omega-3 fatty acids. In the live fish market, individuals face difficulty in hand-picking fish species that meet the daily requirement of appropriate nutrients in the body. This study helps individuals select the fish species that meet their nutritional needs. A dataset comprising 400 images of a total of six fish species, *Labeo catla* (Catla), *Labeo rohita* (Rohu), *Cirrhinus mrigala* (Mrigal), *Labeo bata* (Bata), *Hypophthalmichthys molitrix* (Silver carp), and *Ctenopharyngodon idella* (Grass carp) has been developed by collecting images from live market conditions in some cluttered environments. Two different annotated datasets were developed, fish heads and fish whole body, for the localization and identification of fish species. Two popular object detection deep learning networks, YoLov3 and YoLov5, were used with different model variations to address the problem. Different augmentation techniques were applied to the original dataset to avoid overfitting in deep learning models. These trained networks were finally ensembled using Testing Time Augmentation (TTA) and the Weighted Box Fusion (WBF) technique. Here, one strong region proposal is generated from multiple region proposals. NMS box suppression was used to fine-tune the results further. Using the experimentally selected WBF and TTA-enabled ensemble of YOLO models, the mean Average Precision (mAP) at the IoU threshold of 0.5, 0.91, and 0.79 are achieved in the fish head annotations and fish whole body annotation dataset. A mobile application has been developed for the purchasers to choose fish that meet individuals' nutritional needs. The purchaser captures an image in the fish market and provides various nutritional needs in the application. The proposed system localizes and identifies fish that meet the purchasers' nutritional needs. Indeed, this study will help individuals make informed decisions about selecting appropriate food to maintain a healthy diet.

5.1 Introduction

Possessing the nutritional content of the foods that people consume daily, it becomes essential to follow a healthy diet to strengthen immunity and fight against prevailing diseases. Recently, a balanced diet has become increasingly important in human life. A balanced diet provides the appropriate nutrients, both quantitatively and qualitatively, for the human body to work efficiently. Different lifestyles and chronic diseases, such as diabetes, obesity, cardiovascular diseases, impaired growth and development in children, etc., are triggered by not eating foods rich in proteins, carbohydrates, fat, vitamins, minerals, and water. Proper consumption of macronutrients (carbohydrates, proteins, and fat) and micronutrients (vitamins and minerals) daily reduces different lifestyle diseases. Different persons have different nutritional needs depending on several factors, such as gender, age, sex, and specific health issues. Understanding the nutritional content of foods enables people to identify foods with specific nutrients and make informed decisions to meet their eccentric requirements. Nutritional knowledge allows individuals to compare the nutritional value of different foods and make decisions accordingly.

In this study, one of the major food sources, fish, was taken as part of the research. Daily intake of fish provides several health benefits to the individual because it is rich in Omega-3 fatty acids, especially eicosapentaenoic acid (EPA) and docosahexaenoic acid(DHA), proteins, carbohydrates, vitamins, and minerals, thereby contributing to a portion of the balanced diet. India has a wide coastline of over 7,500 kilometres, bringing forth manifold marine fish species. The country also has huge inland water resources, including rivers, lakes, and ponds, which contribute to producing freshwater fish. India is the world's third largest fish producer, accounting for 7.96 % of global production. The total fish production of India during FY 2022-23 was estimated at 16.25 Million Metric Tons (MMT), with a contribution of 12.12 MMT from the inland sector and 4.13 MMT from the marine sector. Fish is found to be an affordable and rich source of proteins, vitamins, and minerals and is one of the healthy food sources to attenuate nutrient deficiency.

It is tough for common individuals to recognize different fish species in live market conditions, as well as to remember the nutritional contents of different fish species. Consumers have varying nutritional needs depending on their health and overall body balance. These days, it is crucial to choose the right food to fulfil individual nutritional needs. These problems are tackled in this study through two stages or parts. In the first stage, a pipeline consisting of dataset collection, dataset annotation, application of deep learning object detection networks, and result analysis is designed for automatic localization and identification of fish species under some cluttered environments. In the second stage, a mobile application working on an Android (or other) device with internet connectivity is developed, where after capturing the image consisting of fish species, the purchaser will choose their nutritional needs, and the system then localizes and identifies



	Labeo catla	Cirrhinus mrigala	Labeo rohita	Labeo bata	Hypothalmichthys molitrix	Ctenopharyngodon idella
🟧 Train	220	445	1070	110	35	15
🟨 Test	82	103	363	17	11	13
🟩 Valid	69	160	394	13	6	3

(a)

(b)

Figure 5.1: Dataset statistics- (a) Number of fish specimens and their allocation percentage (b) Data statistics after dataset split

only those species that meets the purchaser's nutritional needs. This study will help individuals to consume the proper fish items to maintain a balanced diet. The major contributions of this study are-

- Designed a pipeline to identify fish species in the live fish market condition that meets individual nutritional needs.

- Created a standard dataset of six freshwater fish images, *Labeo catla* (Catla), *Labeo rohita* (Rohu), *Cirrhinus mrigala* (Mrigal), *Labeo bata* (Bata), *Hypophthalmichthys molitrix* (Silver carp), and *Ctenopharyngodon idella* (Grass carp) in the live fish market, with 400 images in different lighting conditions. Fish head annotations and fish whole body annotations were developed to prepare two datasets named **JUDVLP-WBUAFS**: Fishdb-Detection.v1, and **JUDVLP-WBUAFS**: Fishdb-Detection.v2.

- Designed a weighted ensemble of two popular deep learning object detection networks of the YOLO family, YOLOv3, and YOLOv5 using WBF to track and identify the fish species in cluttered fish market environments. Different versions of YOLOv3, and YOLOv5 are used with normal settings and Testing Time Augmentations (TTA) settings.

- Developed a mobile application that purchasers can easily use to identify fish species in cluttered fish market conditions that meet their nutritional requirements. The application works either in the web browser or on a mobile device with an internet connection.

The identification and selection of fish species in a cluttered fish market condition based on consumers' nutritional needs is a challenging job. As per the author's knowledge, the problem statement of this study has not been taken up by the researchers. Recently, several works have been done in the field of fish species classification and segmentation. All the studies can be categorized into two sections: machine learning applications and deep learning applications. Research on fish species classification was started in 1994 and in recent times, it has become a challenging and popular research problem.

Castignolles et al. [5] extracted twelve geometrical features from the fish image, and static threshold segmentation of the fish was performed. A combination of moment variants and several geometrical features was proposed by Zion et al. [72] to classify three different carps, *Cyprinus carpio*(Common carp), *Oreochromis sp* (Nile tilapia), and *Mugil cephalus* (Flathead grey mullet), from 382 images collected by CCD camera setup. In traditional machine learning, the shape feature of the objects plays a vital role in classification tasks. Shape features such as line segments, polygon approximation, Fourier descriptors, etc. were extracted from the fish contour representation to classify 22 images of 9 target species. The tiny dataset used by Lee et al. [7, 9] in their study does not make a validated decision model. The extraction of different robust features using Potential Local Geometric Features (PLGF) and shape measurements was studied by Alsmadi et al. [11]. In their work, a hybrid approach using the back-propagation classifier and genetic algorithm was used to optimize the classification performance. 350 fish images of 20 fish families were used in their work. A combination of color, statistical, and wavelet features was studied by Hu et al. [12] to classify 540 images of six freshwater fish species of China, Grass carp, Silver carp, Bighead carp, Snake head murrel, Wuchang bream, and red-bellied pacu. Multiclass SVM (MSVM) was used as a classifier in their study. Rossi et al. [16] extracted different texture features from morphological traits using pre-selected anchor points in the fish body part. They collected 339 images of seven species from the Turin (Italy) fish market. Another interesting work of combining fish skin texture using Weber Local Descriptor (WLD) and color features was proposed by Tharwat et al. [19] to classify 241 images of four fish species. In their study, an AdaBoost classifier was used for classification purposes. Bangladesh is another river-centric country, with a large proportion of people consuming fish daily. Six different fish species of Bangladesh, *Ompok pabda* (Pabda), *Puntius sophore* (Puti), *Colisa fasciata* (Kholse), *Botia dario* (Boumach), *Nandus nandus* (Meni), and *Awaous guamensis* (Bele), were used for the classification purposes by Sharmin et al. [73]. They have applied a combination of color intensity, geometric features, spectral features, and GLCM. An ensemble of decisions from the K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) was used for classification. Fish classification using deep learning techniques started mainly in 2018. Hussain et al. [20] proposed a modified AlexNet with four convolution layers and two fully connected layers to classify six fish species. QUT fish dataset was used for training and validation, and testing was done using the Lifeclef 2015 fish dataset. An optimized VGGNet was proposed by Montalbo and Hernandez [23] for classifying 530 images of the FishBase dataset. Rauf et al. [24] designed a 32-layer customized CNN to classify 438 images of the Fish-Pak dataset. They separately applied the CNN on the fish's body, head, and skin. A hybrid DL model using a pre-trained VGG16 model and a stacked ensemble model was proposed by Chhabra et al. [74]. A total of 435 fish images of 8 different fish species, Cod, Mackerel, Platy, Pollock, Salmon, Swordtail, Tilapia, and Zebra danio, were used in their study. Banerjee et al. [1] proposed a dataset named **JUDVLP-WBUAFS**: Fishdb-IMC.v1 of 1500 images of Indian major carps and used deep convolutional autoencoder to find latent features. They have applied different machine learning techniques as well as deep learning techniques to classify the species. Garcia et al. [50] used deep learning for automatic segmentation of fish images in commercial trawling and measured the size of the fish to reduce the catches of undersized fish. Banerjee et al. [2] proposed a dataset named **JUDVLP-WBUAFS**: Fishdb-EC.v1 and used VGG-16, VGG19, InceptionV3, InceptionResNetV2, and MobileNet for the classification of three exotic carps. In 2023, Banerjee et al. [3] proposed a framework for recognizing and counting freshwater fish species in a cluttered environment. A total of six fish species, *Labeo catla*, *Labeo rohita*, *Cirrhinus mrigala*, *Labeo bata*, *Hypophthalmichthys molitrix*, and *Ctenopharyngodon idella*, were used in their study. A dataset named **JUDVLP-WBUAFS**: Fishdb-Detection.v1 with 400 images was prepared by collecting images from the different live fish markets in West Bengal under unconstrained environments. Only the fish heads were considered for the recognition of the fish species.

The **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset, reported in this study, is the only dataset designed for localizing and classifying freshwater fish species in a cluttered environment. In the present work, the fish head and visible portions of the body were used for the experiment. Based on the required nutritional requirements, the species are identified automatically, and the user can select the appropriate fish.

5.2 Dataset Preparation

In this case study, two different datasets were utilized, named **JUDVLP-WBUAFS**: Fishdb-Detection.v1 and **JUDVLP-WBUAFS**: Fishdb-Detection.v2. The **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset is utilized for detecting some fish species under some cluttered environment(Chapter 4). Only the fish heads were annotated and utilized for localization and identification purposes. In this case study, along with the fish head annotation, the visible part of the fish body is annotated. The **JUDVLP-WBUAFS**: Fishdb-Detection.v2 dataset comprises the fish full body annotation. In the cluttered environment of fish markets, it is very tough to see the full fish body, because of the occlusion of one fish by another fish. The full-body annotation aims to compare the localization and detection performance of head annotations and full-body annotations. During the preparation of the **JUDVLP-WBUAFS**: Fishdb-Detection.v2 dataset, 400 images present in the **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset were used. Using the stratified group shuffling, these images were divided into training, validation, and testing parts with 209, 116, and 75 images. As the dataset size was small, nine different augmentation techniques such as horizontal flip, vertical flip, blur, gaussian noise, hue saturation, RGBshift, 45°rotation, 90°rotation, and 180°rotation were applied to the training part only. After augmentations were done, the training part consisted of 2,090 images with 19,995 fish instances. In the Algorithm 3, the dataset annotation and augmentation process is given and the entire process of the dataset preparation is explained in Chapter 2. Figure 5.1(a) represents the allocation of different fish species with the count of instances per fish species and Figure 5.1(b) represents the data distribution in the training, validation, and testing phases.

Table 5.1: Information of different nutritional facts of six different fish species in this study

	Catla	Rohu	Bata	Mrigal	Silver carp	Grass Carp
Calories (Kcal/100g)	89.92	88.49	100.52	96.35	103.97	81.50
Proteins (g/100g)	16.18	15.98	15.64	16.19	17.2	15.2
Fat (g/100g)	2.8	2.73	3.74	3.51	4.1	1.1
Calcium (mg/100g)	161.11	219.03	211.27	174.82	903	54
Sodium (mg/100g)	198.31	202.11	4.23	184.78	96	73
Potassium (mg/100g)	283.91	267.5	14.51	142.59	225	300
Iron (mg/100g)	1.61	2.19	0.07	1.48	4.4	0.46
Manganese (mg/100g)	0.32	0.44	0.07	0.26	-	0.02
Zinc (mg/100g)	1.35	1.98	0.02	0.89	1.4	0.91
Selenium (mg/100g)	0.23	0.68	-	0.17	12	31
EPA (mg/100g)	0.19	0.03	140	0.24	0.12	0.03
DHA (mg/100g)	0.13	0.01	120	0.04	0.03	0.03
ω**3 (mg/100g)**	0.63	0.213	420	0.48	1.04	0.32
ω**6 (mg/100g)**	0.24	0.23	410	0.09	0.94	0.39
Vitamin A (IU/100g)	30.53	4.22	207.63	42.43	-	-
Vitamin D (IU/100g)	102.4	36.08	351.6	203.2	0.24	-
Vitamin E (IU/100g)	0.48	0.54	-	0.78	0.49	-
Vitamin K (mcg/100g)	1.21	0.41	-	0.15	-	-

5.3 Nutritional information in the fish species

Malnutrition is one of the major consequences of low consumption of different micronutrients. Many food items are available with an adequate amount of micronutrients, and fish is one of the most consumed foods in India, which contains sufficient nutrients to maintain a healthy diet. In this study, six different freshwater fish species were considered. These fish species have a good amount of calorie and protein content. Silver carp (103.97 Kcal/100 g) and Bata (100.52 Kcal/100 g) fish carry the most calorie content, Catla (16.18 gm/100 g) and Mrigal (16.19 g/100 g) fish carry the most protein content out of all the species. Maximum calcium is found in Silver carp (903 mg/100 g), followed by Rohu (219.03 mg/ 100 g) fish. Catla fish carry a maximum potassium of 283.91 mg/100 g, followed by Rohu fish, carrying a potassium content of 267.5 mg/ 100 g. Maximum iron is found in Rohu fish (2.19 mg/ 100 g), and maximum zinc is found in Rohu fish (1.98 mg/ 100 g). Bata fish is enriched in DHA, $\omega3$, $\omega6$ with 120 mg/100 gm, 420 mg/100 g,

```
Algorithm 3: Algorithm for Dataset Annotation and Augmentation
  Input: Collection of 400 images, each image generally contains different species
         out of- Labeo catla (Catla), Labeo rohita (Rohu), Cirrhinus mrigala
         (Mrigal), Labeo bata (Bata), Hypophthalmichthys molitrix (Silver carp),
         and Ctenopharyngodon idella (Grass carp).
  Output: JUDVLP-WBUAFS: Fishdb-Detection.v1 and
          JUDVLP-WBUAFS: Fishdb-Detection.v2 dataset
  Split the C into training $C_{Tr}$, validation $C_v$, and testing $C_{Te}$ part with 209, 116,
    and 75 images using a stratified group shuffling technique. Import each dataset
    part in the LabelMe tool.
  Draw a rectangular bounding box around each fish head and label each box with
    its corresponding class. ;   // JUDVLP-WBUAFS:Fishdb-Detection.v1 dataset
  Draw a rectangular bounding box around the visible portion of the fish body and
    label each box with its corresponding class. ;
    // JUDVLP-WBUAFS:Fishdb-Detection.v2 dataset
  Annotations of each image in JUDVLP-WBUAFS: Fishdb-Detection.v1 and
    JUDVLP-WBUAFS: Fishdb-Detection.v2 datasets are checked by domain
    experts to ensure accuracy and consistency.
  The .json file for each image is converted into YOLO-specific format
    $< class, X_{center}, Y_{center}, width, height >$.
  The training part $C_{tr}$ of JUDVLP-WBUAFS: Fishdb-Detection.v1 and
    JUDVLP-WBUAFS: Fishdb-Detection.v2 dataset is augmented with nine
    different augmentation- horizontal flip, vertical flip, blur, gaussian noise, hue
    saturation, RGBShift, 45°rotation, 90°rotation, and 180°rotation. A total of 2090
    images are prepared in the training part from 209 images using the nine different
    augmentations. Images are augmented as well as .json files are generated. .json
    files are finally converted into YOLO-specific format.
```

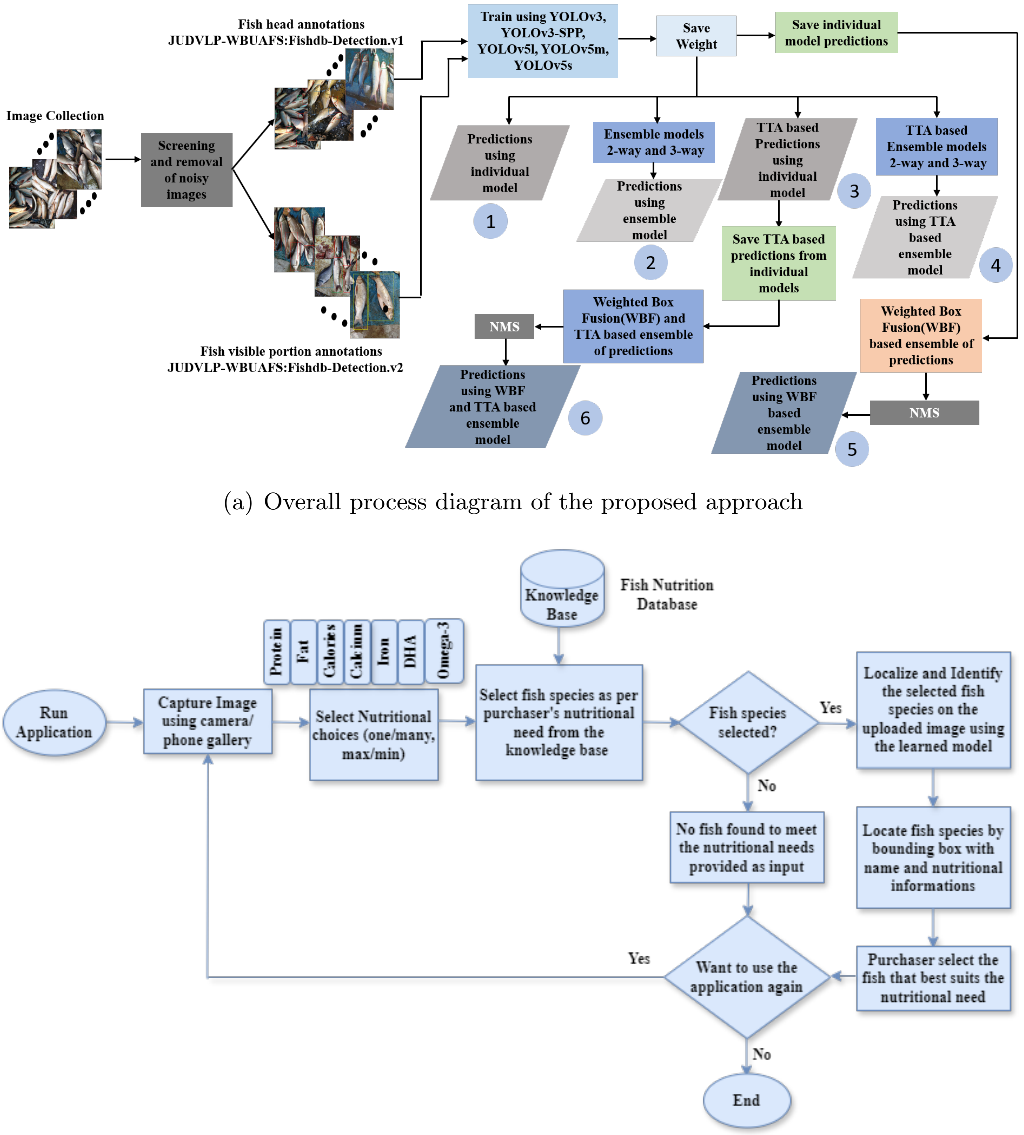
and 410 mg/ 100 g. Of all the fish species used in this study, Bata fish is also enriched in vitamin A and vitamin D. All the nutritional facts of these species were taken from Paul et al. [75], Paul et al. [76], Paul et al. [77], DEPARTMENT OF AGRICULTURE [78], Ashraf et al. [79], Pyz-Lukasik and Kowalczyk-Pecka [80].

5.4 Methodology

In this study, two different annotations, head annotation and whole body annotation, dataset named **JUDVLP-WBUAFS**: Fishdb-Detection.v1 and **JUDVLP-WBUAFS**: Fishdb-Detection.v2 is used. The entire study is designed in two stages as stated in the introduction 5.1.

Localization and Identification:

In the first stage, automatic localization and identification of the fish species under some cluttered environment is solved. Here, a total of three experiment protocols are designed and utilized in the experiments. In the first experiment protocol, different versions of YOLOV3 (YOLOV3, YOLOV3-SPP), and YOLOV5 (YOLOV5l, YOLOV5m, YOLOV5s) in a normal setting and Testing Time Augmentation (TTA) enabled setting

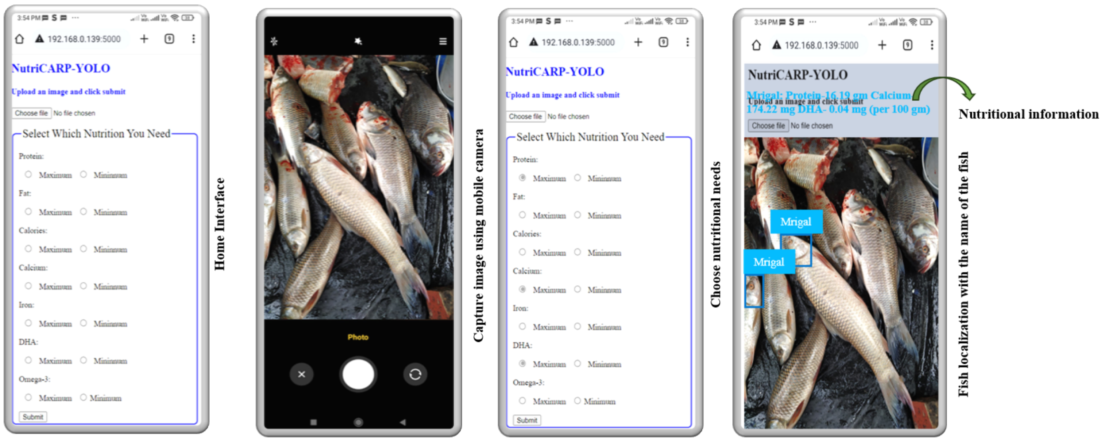

Fish head annotations
JUDVLP-WBUAFS:Fishdb-Detection.v1
Image Collection
Screening and removal of noisy images
Fish visible portion annotations
JUDVLP-WBUAFS:Fishdb-Detection.v2
Train using YOLOv3, YOLOv3-SPP, YOLOv5l, YOLOv5m, YOLOv5s
Save Weight
Save individual model predictions
Predictions using individual model
1
Ensemble models 2-way and 3-way
Predictions using ensemble model
2
TTA based Predictions using individual model
3
Save TTA based predictions from individual models
TTA based Ensemble models 2-way and 3-way
Predictions using TTA based ensemble model
4
Weighted Box Fusion(WBF) and TTA based ensemble of predictions
NMS
Predictions using WBF and TTA based ensemble model
6
Weighted Box Fusion(WBF) based ensemble of predictions
NMS
Predictions using WBF based ensemble model
5

(a) Overall process diagram of the proposed approach

Knowledge Base
Fish Nutrition Database
Protein
Fat
Calories
Calcium
Iron
DHA
Omega-3
Run Application
Capture Image using camera/ phone gallery
Select Nutritional choices (one/many, max/min)
Select fish species as per purchaser's nutritional need from the knowledge base
Fish species selected?
Yes
No
Localize and Identify the selected fish species on the uploaded image using the learned model
No fish found to meet the nutritional needs provided as input
Locate fish species by bounding box with name and nutritional informations
Purchaser select the fish that best suits the nutritional need
Want to use the application again
Yes
No
End

(b) Flow diagram of the proposed mobile application

Figure 5.2: Flow diagram of the proposed approach and mobile application developed for the purchasers ́



192.168.0.139:5000

NutriCARP-YOLO

Upload an image and click submit

Choose file No file chosen

Select Which Nutrition You Need

Protein:
○ Maximum ○ Minimum

Fat:
○ Maximum ○ Minimum

Calories:
○ Maximum ○ Minimum

Calcium:
○ Maximum ○ Minimum

Iron:
○ Maximum ○ Minimum

DHA:
○ Maximum ○ Minimum

Omega-3:
○ Maximum ○ Minimum

Submit

Mrigal: Protein-16.19 gm Calcium-174.22 mg DHA- 0.04 mg (per 100 gm)

Figure 5.3: Functional flow diagram of the mobile application for the proposed study

are trained on the above two datasets individually and the predictions from individuals are saved. Ensemble of two models (2-way), and three models (3-way) is performed on these trained models in the second experiment protocol. Here, each model is assigned equal weight during the ensembling process. In the third experiment protocol, a Weighted Box Fusion (WBF) is used to ensemble individual predictions using an experimented weight assignment on the normal models and TTA-based models. After the fusion phase, Non Maximum Suppression (NMS) is used to remove duplicate predictions if present. The entire workflow of the proposed method is shown in Figure 5.2(a).

5.4.1 Mobile Interface:

In the second stage of the proposed work, a mobile interface was developed workable on Android (or another mobile) device to identify fish species in some cluttered environment based on the purchaser's nutrition requirements. The application works properly on the Android device and enables images captured by the rear camera. Upon image collection, the purchaser provides different nutritional needs such as protein, fat, calories, calcium, iron, DHA, and Omega-3 in two modes, either maximum or minimum. The captured image and the different nutritional needs are sent to the Web server, where the saved model localizes and identifies different fish species in the image and then selects fish species that meet the purchaser's nutritional needs. In the end, the mobile application shows the bounding box around those fish species that meet the nutritional needs of the purchaser and also shows the nutritional contents of each fish detected by the model. The entire workflow of the development of the mobile application is presented in Figure 5.2(b). In Figure 5.3 the functional flow diagram of the mobile application is shown. Also, the different stages in the mobile application are presented in the same figure. The mobile application was designed using Flask web server and REST-full web API. The communication between the front-end and the back-end web server is maintained by the REST-full API through the .json data format. The mobile device and the web server should be connected through the same network for the proper functioning of the application. Efficient deployment of deep learning models and scalability needs appropriate handling of large deep learning models. Uploading large deep learning models in the back end is pure overhead and generates delays in the application. Instead of loading the entire deep learning model, lazy loading is applied, where the model is loaded only when it is required after the flask server starts up. The model is loaded into the system only once and remains in the memory globally, and all the functions that need to use the model can easily refer to the already uploaded model instead of loading it separately. This process reduces the overhead of the model loading and handles the large model loading efficiently.

The primary objective of this study is to propose an automatic technique to locate and identify fish species in a cluttered fish market based on consumer nutritional needs. It is challenging to remember different nutritional facts of several fish species and screen them according to specific needs before buying them in the fish market. The proposed method enables the consumer to buy a fish species that meets certain nutritional needs

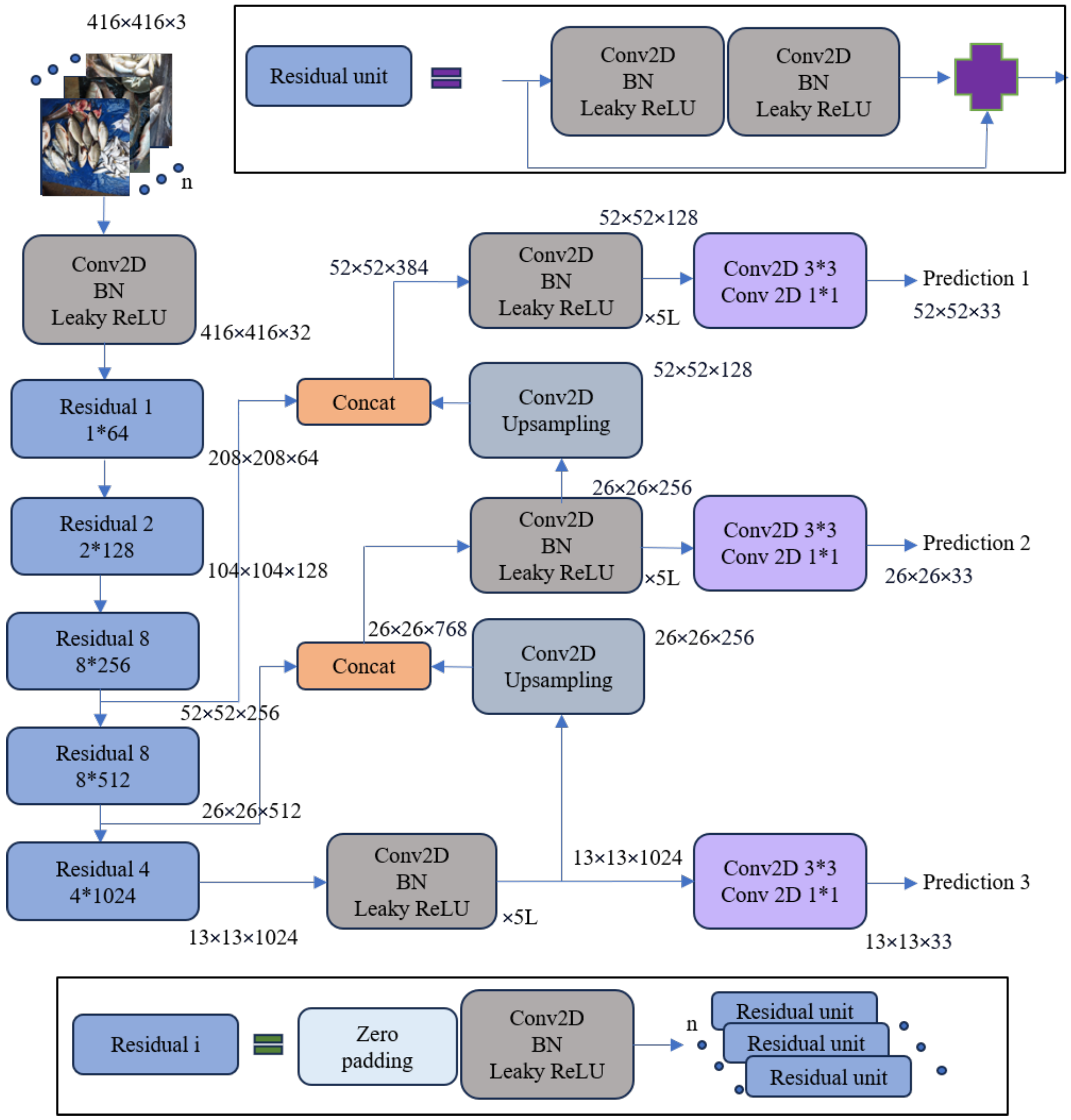


Figure 5.4: Architecture of the YOLOV3 network

specified in Table 5.1. Head annotations and whole body annotations developed in this study were used during the experiments using the proposed localization and identification techniques. Two different popular and stable object detection modules based on deep learning, YoLov3, and YoLov5 were employed in this study because of their excellent performance and considerable computing power. These two models are described in detail here.

5.4.2 Architecture of YOLOv3 and YOLOv5

YOLOv3

You Only Look Once (YOLO) is a family of one-stage deep-learning object detection models. Due to its fast speed and higher precision, YOLO has been the best choice to employ in object detection tasks. YOLOv3 uses refined network architecture DarkNet-53,

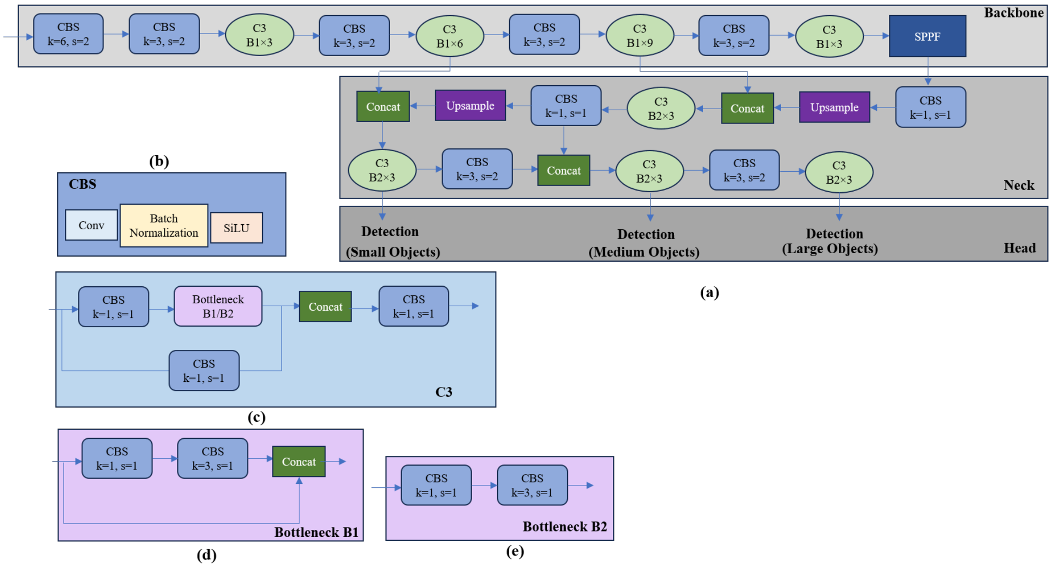


Figure 5.5: Architecture of the YOLOV5 network- (a) Overall architecture (b) CBS unit architecture (c) C3 layer architecture (d) Bottleneck B1 architecture (e) Bottleneck B2 architecture

with 53 layers stacked together, making 106 layers. YOLOv3 applies 1 × 1 kernel on three different size feature maps to achieve a multiscale detection objective. CBL block in YOLOv3 block consists of Conv2D, Batch Normalization, and Leaky Relu as shown in Figure 5.5. With the CBL block, the multiple residual block is present in this network architecture and plays a crucial role. Each residual block consists of zero padding, a CBL block, and several residual units. To achieve feature maps at different levels, feature maps are concatenated. In YOLOv3, pooling layers are not used, instead, a convolutional layer with stride two is used to downsample the feature maps. It helps to avoid the loss of low-level features. Detections are made in three scales 13 × 13, 26 × 26, and 52 × 52. It predicts a total of $((52 \times 52) + (26 \times 26) + 13 \times 13))x3 = 10647$ bounding boxes. As there are six categories to be recognized, the number of channels in the final prediction is $(4 + 1 + 6) \times 3 = 33$. Based on the objectness score and Non-maximum Suppression (NMS), the multiple detection problem is solved. In Figure 5.5, the overall architecture of YOLOv3 is demonstrated.

YOLOv5

YOLOv5 is the PyTorch implementation of YOLOv4, including some major architecture revisions. It uses CSPDarkNet53 as the backbone instead of DarkeNet53. In DarNet53, the residual and dense blocks help flow into the network's deep layers. The major problem due to this is redundant gradients. CSPDarkNet53 discards this gradient flow and helps to remove the redundant gradient flow. In the neck part of the network, the Path Aggregation Network (PANet), which uses the concept of Feature Pyramid Network (FPN) is used. PANet is used to warp the information flow using FPN, by employing different top-down and bottom-up paths. YOLO head is used in the YOLOv5 head part, generating three different feature maps for small, medium, and large objects. The Focus unit, found in the previous versions, is replaced in YOLOv5 by a 6×6 Conv2D structure. The SPP unit is replaced by SPP-Fast (SPPF) unit, simplifying the pooling process and avoiding the repetitive operation in SPP. SPPF is roughly two times faster than SPP structure. The CBS structure consists of convolution, Batch Normalization, and SiLu activation function as shown in Figure 5.5. The C3 Layer in Figure 5.5 is a Cross Stage Partial (CSP) bottleneck with 3 CBS structures. It does a Convolution on the input tensor and concat the result to the same tensor passed through a CBS and a series of bottleneck layers. Finally, the concatenated tensor is passed again through a CBS structure. The architecture of CBS unit, C3 unit, Bottleneck B1, and Bottleneck B2 are presented in Figure 5.5(b), (c), (d), and (e). The loss in YOLOv5 is calculated as a combination of Binary Cross-Entropy (BCE) Loss, Objectness Loss (BCE), and Location Loss(CIoU). The first Binary Cross-Entropy loss measures the classification loss and the second BCE loss measures the error in detecting whether an object is present inside a bounding box. The CIoU loss measures the localization error within a grid cell. Using a weight vector of $[4.0, 1.0, 4.0]$, the objectness loss is computed on small, medium, and large object parts. The overall neat architecture of the YOLOv5 network is shown in Figure 5.5.

5.4.3 Non Max Suppression (NMS)

YOLO models propose multiple bounding boxes for each object of interest in an image. Non-maximum suppression (NMS) is the technique to find the best bounding box proposal based on the Intersection of Union (IoU) and confidence score. YOLO model generates the bounding box proposals in the format $(x1, y1, x2, y2, class, confidence)$. All the bounding boxes in NMS are sorted according to the confidence score. A confidence threshold $conf_{th}$ is selected, and based on that, all the bounding boxes with less confidence score compared to $conf_{th}$ are deleted. Next, an IoU threshold IoU_{th} is selected, and then for each bounding box proposal, all the remaining bounding box proposals are checked for the overlapped region based on $Conf_{th}$. For example, if two boxes are found with high IoU (see Eqn. 5.1), they are treated with the same class, the bounding box with the highest confidence score is selected, and the rest are discarded. The filtering of the bounding box proposals in NMS is sensitive to the selection of the IoU threshold. If multiple objects are present side by side, applying NMS leads to deleting bounding boxes around objects.

$$IoU = \frac{Intersection(BBox1, BBox2))}{Union(BBox1, BBox2)} \tag{5.1}$$

In object detection tasks, the learned models predict bounding box proposals, object labels, and class confidence scores for each box. A single model may predict multiple boxes for a single object in an image. In the ensembling process, multiple bounding boxes per object come into play because different models' predictions are added up. In NMS, a box belongs to an object if the intersection-over-union (IoU) is higher than a set threshold, and it suppresses all other bounding boxes. This means NMS depends on a single IoU threshold, and selecting this threshold value is a very tricky job. If the threshold is high, some valid boxes may be removed because of low confidence score. Again, setting a low threshold value retains many overlapping boxes, increasing false positives. This problem is somehow handled in Soft-NMS technique, by modifying confidence scores of overlapping bounding boxes gradually. This modification is usually done in a way that reduces the confidence score overlapping bounding boxes, making them less likely to be selected as the top bounding box during the suppression process. But, NMS and Soft-NMS both remove redundant bounding boxes, and thus getting averaged prediction from all proposals is not possible.

5.4.4 Weighted Box Fusion

The NMS process is not efficient in generating an average prediction of bounding boxes from different detection modules. When multiple detection models are ensembled, the Weighted Box Fusion (WBF) technique is important in producing an average prediction. To understand the process of WBF, consider two bounding box proposals from two models M_1 and M_2, $bboxM_1$:(M_1x_1,M_1y_1,M_1x_2,M_1y_2, $class_{M_1}$,CS_{M_1}), and $bboxM_2$:(M_2x_1,M_2y_1,M_2x_2,M_2y_2,$class_{M_2}$,CS_{M_2}). The M_1x_1,M_1y_1 represent the upper left corner coordinates and M_1x_2,M_1y_2 represent the lower right corner coordinates. The $class_{M_1}$, CS_{M_1} represents the predicted class label and the confidence score of the bounding box generated by model M_1. The same representation holds for the bounding box $bboxM_2$. Using the WBF technique, a combined bounding box $bbox_{fused}$ is produced by fusing $bboxM_1$ and $bboxM_1$. The different parameters of $bbox_{fused}$ are calculated using the formula given in Equation 5.2 to Equation 5.6.

$$fusedx_1 = \frac{M_1x_1 \times CS_{M_1} + M_2x_1 \times CS_{M_2}}{CS_{M_1} + CS_{M_2}} \tag{5.2}$$

$$fusedy_1 = \frac{M_1y_1 \times CS_{M_1} + M_2y_1 \times CS_{M_2}}{CS_{M_1} + CS_{M_2}} \tag{5.3}$$

$$fusedx_2 = \frac{M_1x_2 \times CS_{M_1} + M_2x_2 \times CS_{M_2}}{CS_{M_1} + CS_{M_2}} \tag{5.4}$$

$$fusedy_2 = \frac{M_1y_2 \times CS_{M_1} + M_2y_2 \times CS_{M_2}}{CS_{M_1} + CS_{M_2}} \tag{5.5}$$

$$CS_{fused} = \frac{CS_{M_1} + CS_{M_2}}{2} \tag{5.6}$$

Weighted Box ensemble of YOLO models

In this study, different variations of the YOLOv3 and YOLOv5 models, YOLOv3, YOLOv3-SPP, YOLOv5l, YOLOv5m, and YOLOv5s were applied in the experiments. These two YOLO versions were selected in this study because of their excellent performance, speed, and lightweight nature. Each predicted box from each model under the ensemble process was included in a list and sorted in descending order based on the confidence score. A cluster of the bounding boxes was prepared after checking the similarity using the IoU. If the IoU is higher than a threshold, then the similarity is established. From this cluster, the final confidence score and coordinates of the bounding box were calculated using equations 5.2 to 5.6. Finally, the NMS was applied to remove any further redundant bounding box. The grid search method was used to find the optimal weight for the different models in the ensemble process. The ensemble of models produces a robust and generalized detection model to localize and detect fish species according to the consumer's nutritional needs. Different ensembles were experimented with, and the results proved that the proposed approach effectively improved the performance compared to the single YOLO model.

5.4.5 Testing Time Augmentation (TTA)

Testing Time Augmentation (TTA) is a technique used for evaluating machine learning model performance. In TTA, predictions for test data are obtained not only on the original data but also on multiple versions of the data obtained through transformations. These transformations are typically simple operations like flipping, rotation, scaling, cropping, or colour variations for images. The key idea behind TTA is to obtain multiple predictions for each test sample by applying different transformations and aggregating these predictions to produce a final result. Aggregation can be done using various methods such as averaging, voting, or taking the maximum probability, depending on the task and the model being used.

TTA helps to evaluate the model's performance under various conditions and transformations, which can improve its robustness against variations in the test data. By obtaining multiple predictions for each test sample and aggregating them, TTA can provide a measure of confidence or uncertainty in the predictions. TTA can help the model generalize better to unseen variations in the data distribution by exposing it to a diverse set of transformations during evaluation. Overall, TTA is a valuable technique for more robust evaluation of machine learning models, providing insights into their generalization capabilities and helping to identify potential weaknesses.

5.5 Experiment Protocols and Results

After the dataset preparation and augmentation were done, two YOLOv3 models: YOLOv3 and YOLOv3-SPP, and three YOLOv5 models, YOLOv5l, YOLOv5m, and YOLOv5s were applied on the training part of the **JUDVLP-WBUAFS**: Fishdb-Detection.v1 and **JUDVLP-WBUAFS**: Fishdb-Detection.v2 dataset. All the models with pre-trained weight matrices trained on the ImageNet dataset were utilized in all the experiments. As the dataset size is too small to apply these models from scratch, the transfer learning approach was our prime choice to handle the issue and then the models were fine-tuned on the proposed datasets. All the experiments were done on the two different annotation schemes, head annotation and whole body annotation.

The trained YOLO models were ensembled using a weighted box fusion approach, with the experimentally selected weights on different models. Also, models were ensembled without any assigned weights for the comparative analysis. The NMS technique was used to remove the bounding boxes with IoU less than the fixed threshold. Ensembling of the YOLO models was proposed to achieve better recognition performance and robustness in comparison to a single model.

Before starting the training process, the configuration file (data.yaml) was prepared by setting the directory path of the training, validation, and test part of the dataset. The number of classes (nc) was set to 6 and names of the species were assigned as ['Catla','Mrigal','Rohu','Bata','Silver carp','Grass carp']. All the images were resized to a dimension of 416 × 416. The training process was done for 16 epochs with a batch size of 32. If no improvement in mAP was recorded, the training process was stopped, and the best weight was saved. Different hyperparameters involved in the training process were fine-tuned on the proposed datasets. To evaluate the performance of the proposed approach, mAP@0.5, precision, and recall were used. In IoU, the relation between test set predictions and the ground truth is evaluated. The true positive (TP) and false positive (FP) were calculated using a fixed IoU threshold. The ensembling of the trained models (ETM) and testing time augmentation (TTA) were applied during the testing time. ETM averages the predicted bounding boxes of different models and all the hyperparameters. The ensemble models provide better generalized and robust predictions than single model performance. In TTA, different augmented (flipped and resolution change) versions of the base images were created and predictions were obtained. Finally, all the predicted boxes were averaged, and using NMS the duplicate bounding boxes were removed based on some fixed IoU threshold. Also, ETM and TTA were applied to minimize the generalization error and get better recognition performance.

All the experiments were done as per three experiment protocols. All these experiment protocols are described here, with the results.

5.5.1 Experiment Protocol 1:

In this protocol, the previously mentioned YOLO models were used in the proposed datasets individually. Also, the TTA was done on these models and compared with the original one. The results are tabulated in Table 5.2. Two YOLOv3 models, YOLOv3, and YOLOv3-SPP achieved the mAP@0.5 of 0.69 and 0.65 on full box annotations and mAP@0.5 of 0.81 and 0.84 on head annotations. TTA improved the performance significantly by taking into account all the predictions of augmented images and the original image as well. When TTA was done on YOLOv3 and YOLOv3-SPP, mAP@0.5 of 0.76 and 0.69 was achieved on full box annotations, and mAP@0.5 of 0.84 and 0.87 was achieved on head annotations. The YOLOv5l, YOLOv5m, and YOLOv5s achieved mAP@0.5 of 0.65, 0.68, and 0.63 on full box annotations and a mAP@0.5 of 0.78, 0.81, and 0.77 were achieved on head annotations. Among all the YOLOv5 models, applying TTA, a maximum mAP@0.5 of 0.72 was achieved using YOLOv5l on full box

Table 5.2: Performance of the single YOLO models and TTA YOLO models

Models	mAP@0.5		mAp@0.5-0.95	
	Full box annotation	**Head Annotation**	**Full box annotation**	**Head Annotation**
YOLOv3	0.69	0.81	0.37	0.50
YOLOv3-SPP	0.65	0.84	0.37	0.52
YOLOv5l	0.65	0.78	0.40	0.52
YOLOv5m	0.68	0.81	0.36	0.51
YOLOv5s	0.63	0.77	0.34	0.46
YOLOv3-TTA	**0.76**	0.84	0.42	0.53
YOLOv3-SPP-TTA	0.69	**0.87**	0.40	**0.54**
YOLOv5l-TTA	0.72	0.81	**0.45**	0.53
YOLOv5m-TTA	0.68	0.83	0.37	0.53
YOLOv5s-TTA	0.66	0.77	0.35	0.48

annotations. However, on head annotations, YOLOv5m achieved a mAP@0.5 of 0.83 on head annotations using TTA. In full box annotations, YOLOv3 achieved the best mAP@0.5 of **0.76** and YOLOv3-SPP, achieved the best mAP@0.5 of **0.87** on head annotations when TTA was done at the testing time.

Table 5.3: Performance of ensemble of models without using any weights

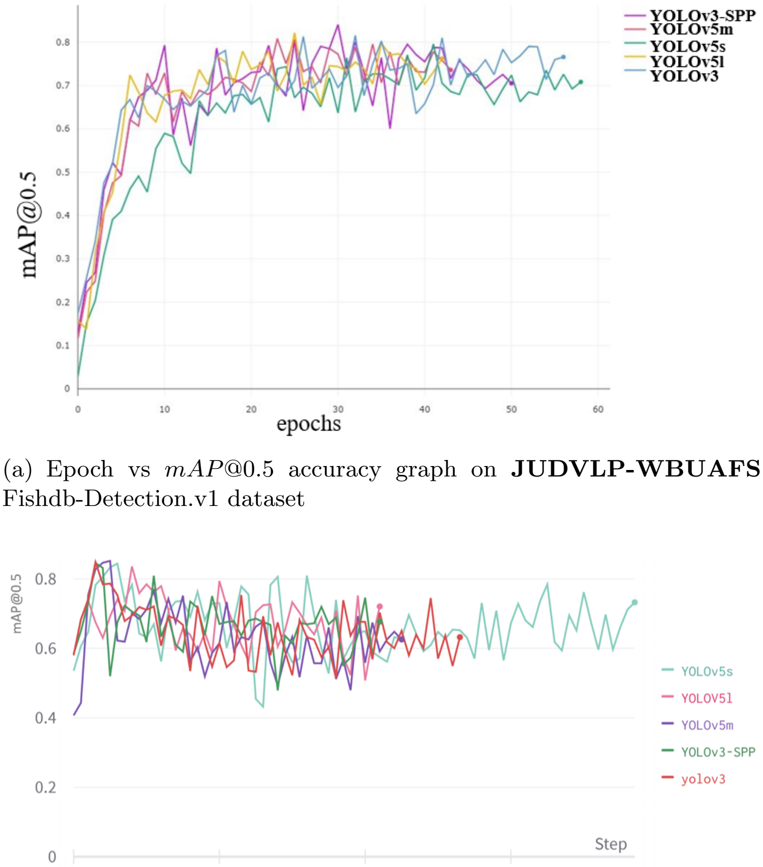
Ensemble Versions	mAp@0.5		mAP@0.5-0.95	
	Full box annotation	Head annotation	Full box annotation	Head annotation
YOLOV3 and YOLOv3-SPP	0.74	0.88	0.40	0.54
YOLOv3 and YOLOv3-SPP-TTA	0.76	0.88	0.44	0.56
YOLOv5l and YOLOv5m	0.73	0.83	0.43	0.54
YOLOv5l and YOLOv5m-TTA	0.75	0.84	0.45	0.56
YOLOv5l and YOLOv5s	0.70	0.81	0.42	0.52
YOLOV5l and YOLOv5s-TTA	0.73	0.82	0.44	0.56
YOLOv5m and YOLOv5s	0.70	0.84	0.38	0.51
YOLOv5m and YOLOv5s-TTA	0.71	0.82	0.39	0.52
YOLOv5l and YOLOv5m and YOLOv5s	0.73	0.84	0.42	0.54
YOLOv5l and YOLOv5m and YOLOv5s-TTA	0.75	0.84	0.44	0.56
YOLOv3 and YOLOv3-SPP and YOLOv5l-TTA	**0.782**	0.88	**0.46**	0.56
YOLOv3 and YOLOv3-SPP and YOLOv5m -TTA	0.76	**0.89**	0.42	**0.57**
YOLOv3 and YOLOv3-SPP and YOLOv5s-TTA	0.77	0.87	0.44	0.552

5.5.2 Experiment Protocol 2:

In this protocol, the promising YOLO models were ensembled using combinations of two models and combinations of three models. The ensemble of trained models (ETM) was

Table 5.4: Performance of ensemble of models using Weighted Box Fusion

Ensemble Versions	mAP@0.5		mAP@0.5-0.95	
	Full Box Annotations	Head Annotations	Full Box Annotations	Head Annotations
YOLOv3 and YOLOv3-SPP-TTA	0.76	0.89	0.44	0.58
YOLOv5l and YOLOv5m-TTA	0.75	0.84	0.46	0.56
YOLOv5l and YOLOv5s-TTA	0.73	0.83	0.44	0.56
YOLOv5m and YOLOv5s-TTA	0.70	0.85	0.39	0.54
YOLOv5l-YOLOv5m and YOLOv5s-TTA	0.75	0.85	0.46	0.57
YOLOv3 and YOLOv3-SPP and YOLOv5l-TTA	**0.79**	0.89	**0.49**	**0.60**
YOLOv3 and YOLOv3-SPP and YOLOv5m-TTA	0.78	**0.91**	0.46	0.59
YOLOv3-YOLOv3-SPP and YOLOv5s-TTA	0.78	0.89	0.46	0.59

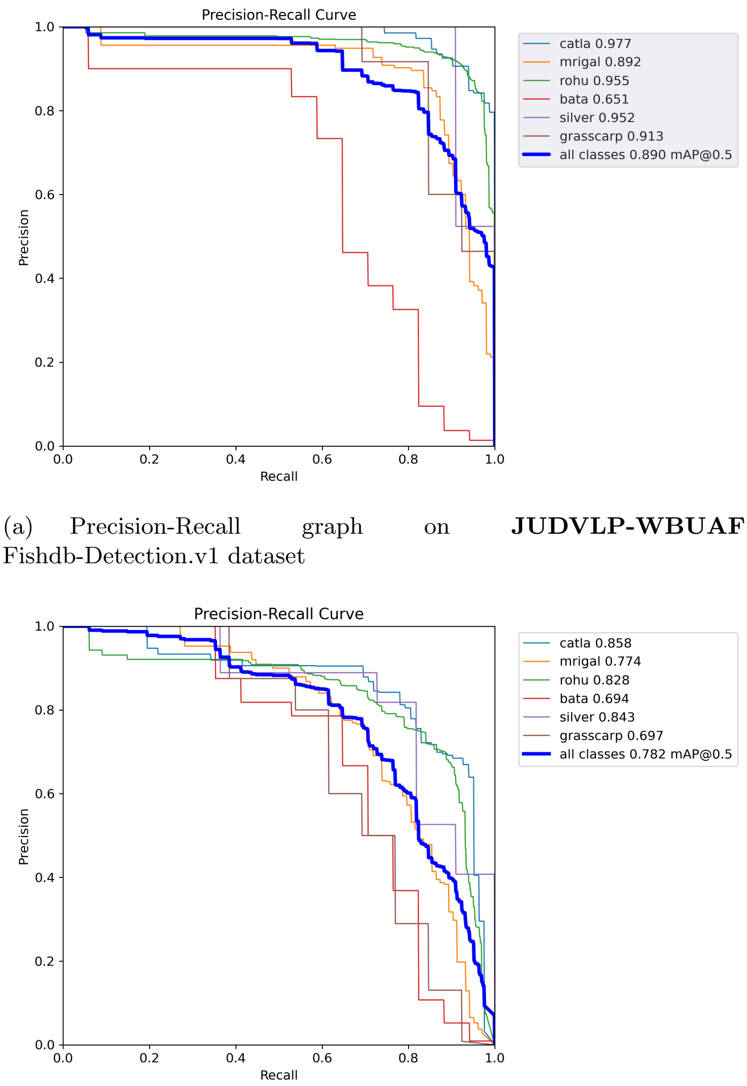


(a) Epoch vs mAP@0.5 accuracy graph on **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset

(b) Epoch vs mAP@0.5 accuracy graph on **JUDVLP-WBUAFS**: Fishdb-Detection.v2 dataset

Figure 5.6: Epoch vs mAP@0.5 accuracy graph after training the YOLO models for 200 epochs

done normally without using any weights on any model. Also, to gain more effective performance, TTA was applied to these ensemble models. The ensemble of YOLOv3 and YOLOv3-SPP achieved a mAP@0.5 of 0.74 on whole body annotation and mAP@0.5 of 0.88 achieved on head annotations. Among the 2-way ensemble of the YOLOv5 models, the ensemble of YOLOv5l and YOLOv5m achieved a mAP@0.5 of 0.73 on whole body annotations and the ensemble of YOLOv5m and YOLOv5s achieved a mAP@0.5 of 0.84 on the head annotations. Based on the TTA ensemble of YOLOv3 and YOLOv3-SPP, mAP@0.5 of 0.76 was achieved on whole body annotations and a mAP@0.5 of 0.88 was achieved on head annotations. Among the 2-way ensemble of YOLOv5 models based on TTA, the ensemble of YOLOv5l and YOLOv5m achieved a mAP@0.5 0f 0.75 on whole body annotations and both the ensemble of YOLOv5l with YOLOv5m and YOLOv5s achieved a mAP@0.5 of 0.84 on the head annotations. All three YOLOv5 models were ensembled, but no improvement in mAP@0.5 performance was recorded. As the performance of the ensemble of YOLOv3 and YOLOv3-SPP gained better mAP@0.5, this combination was selected for the 3-way ensemble. In whole body annotations, the ensemble of YOLOv3, YOLOv3-SPP, and YOLOv5l using TTA gained the mAP@0.5



(a) Precision-Recall graph on **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset

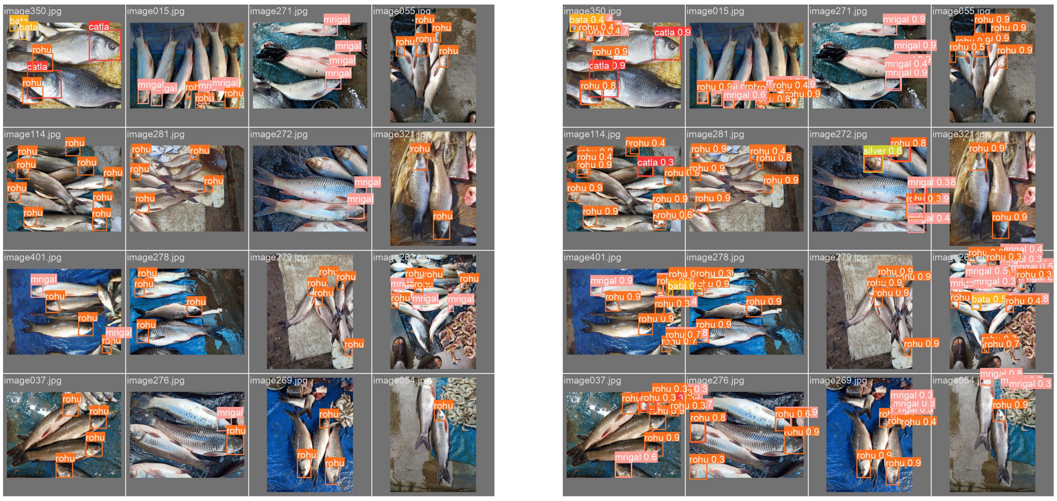
(b) Precision-Recall graph on **JUDVLP-WBUAFS**: Fishdb-Detection.v2 dataset

Figure 5.7: Precision-Recall graph after training the YOLO models for 200 epochs

of 0.782 and outperformed all the other ensemble versions. However, the ensemble of YOLOv3, YOLOV3-SPP, and YOLOv5m using TTA achieved a mAP@0.5 of 0.89 on the head annotations and outperformed the other ensembles. The detailed performance with mAP@0.5-0.95 is tabulated in Table 5.3.

5.5.3 Experiment Protocol 3:

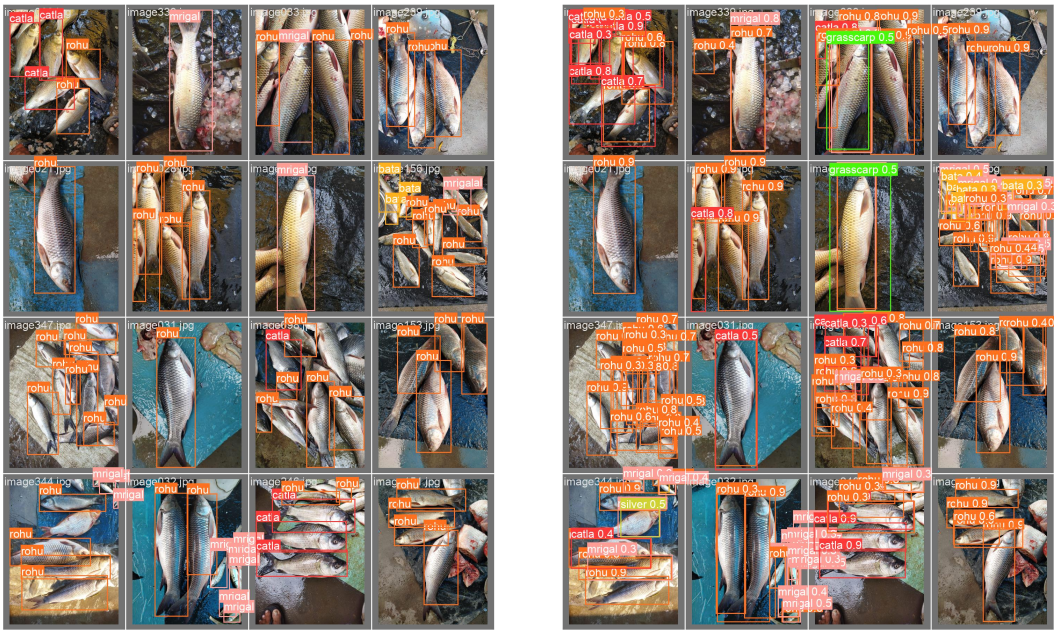
In this protocol, the WBF technique was used to ensemble the predictions from the trained YOLO models. As the performance of the ensemble using the TTA-enabled ETM technique was good enough compared to normal ensembling, the results of the TTA-enabled ETM models were used in this experiment protocol. The NMS technique used in ETM removes some of the valuable occluded bounding boxes, leading to a decrease



(a) Labels of validation batch 1 dataset

(b) Prediction label of batch 1 dataset

Figure 5.8: Ground truth labels and prediction labels of validation batch 1 of **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset

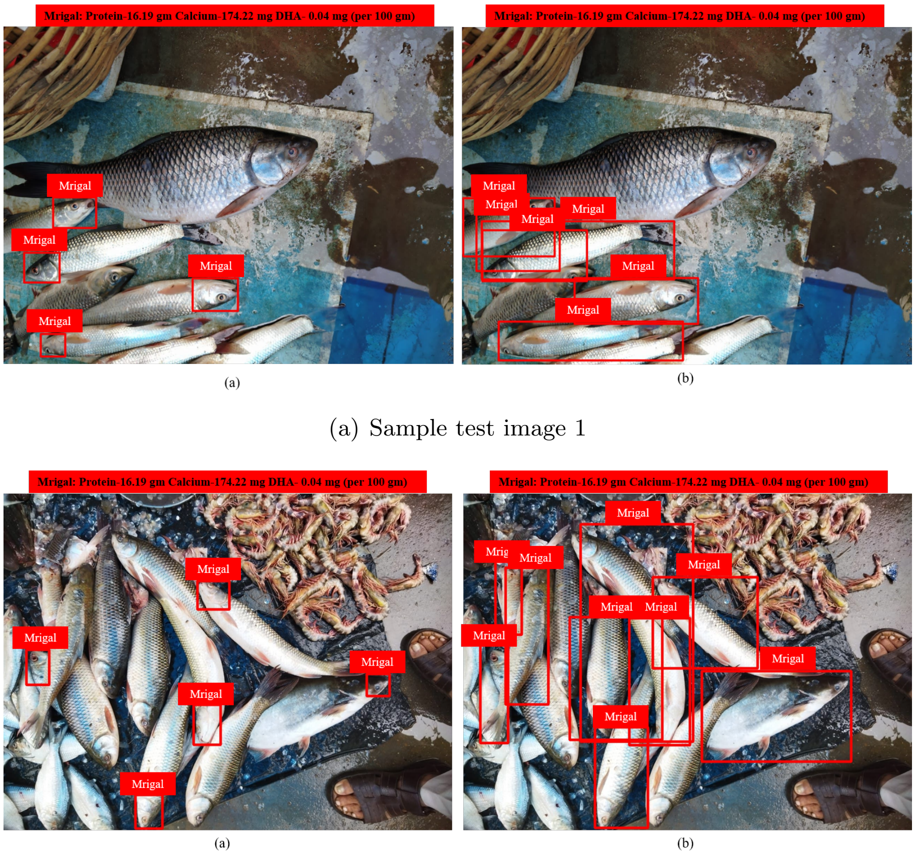


(a) Labels of validation batch 1 dataset

(b) Prediction label of batch 1 dataset

Figure 5.9: Ground truth labels and prediction labels of validation batch 2 of **JUDVLP-WBUAFS**: Fishdb-Detection.v2 dataset

in the model performances. The ensemble of YOLOV3 and YOLOv3-SPP using WBF and TTA achieved the mAP@0.5 of 0.76 and 0.89 on whole body annotations and head annotations. Among all the 2-way ensembles of YOLOv5 models using WBF and TTA, the ensemble of YOLOv5m and YOLOv5s achieved mAP@0.5 0.85 on head annotations, whereas, the ensemble of YOLOv5l and YOLOv5m achieved a mAP@0.5 of 0.75 on whole body annotations. In most cases, using WBF, the performance of the models was improved significantly. Like experiment protocol 2, YOLOv3, and YOLOv3-spp

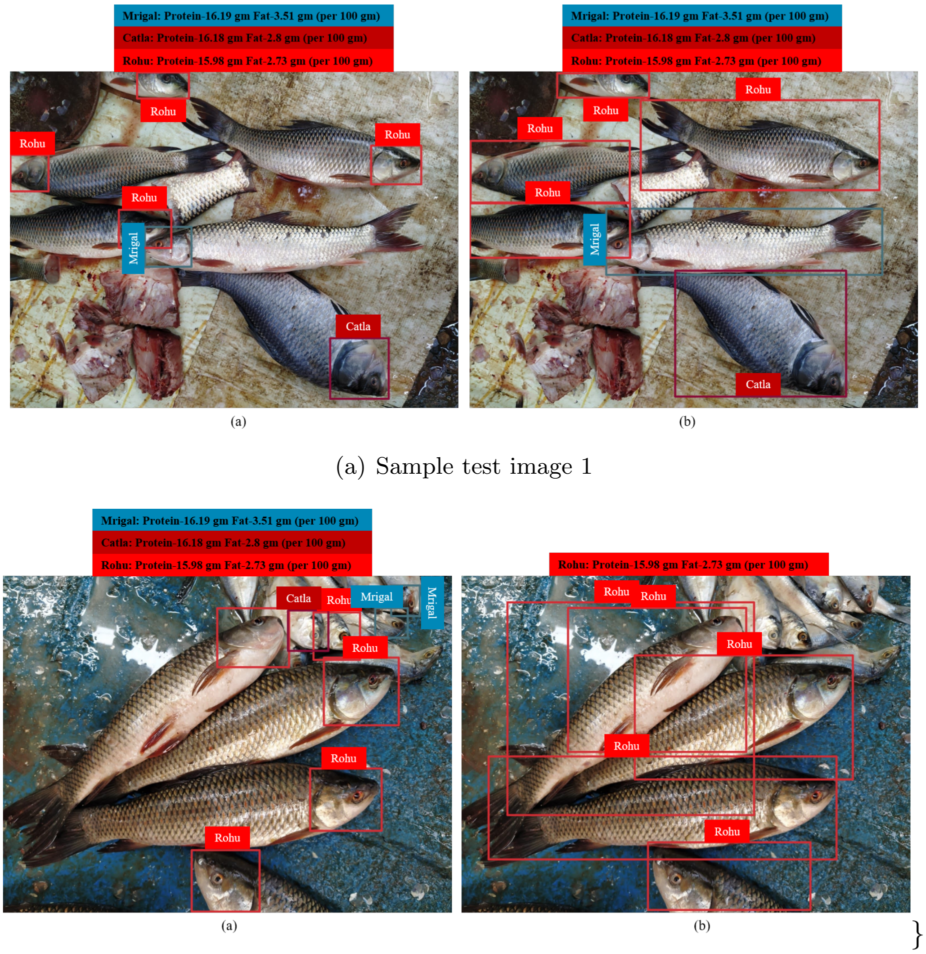

Mrigal: Protein-16.19 gm Calcium-174.22 mg DHA- 0.04 mg (per 100 gm)
Mrigal
(a)
(b)

(a) Sample test image 1

(b) sample test image 2

Figure 5.10: Output of the proposed approach for sample test images 1 and 2 (Nutritional facts-(*Protein-maximum, Calcium-maximum, DHA-maximum*)): (a) Head annotations (b) Full box annotations

were chosen as the candidate to make the 3-way ensemble of models. The ensemble of YOLOv3, YOLOv3-SPP, and YOLOv5l using WBF and TTA achieved the best mAP@0.5 of 0.79 on the whole body annotations and surpassed all the previous ensembles. In head annotations, the ensembles of YOLOv3, YOLOv3-SPP, and YOLOv5m using WBF and TTA achieved the best mAP@0.5 of 0.91, and surpassed the performance of all the other ensembles experimented in this study. Most interestingly, using the proposed ensemble technique using WBF and TTA, the ensemble of YOLOv3, YOLOv3-SPP, and YOLOv5l achieved a mAP@0.5-0.95 of 0.60 on the head annotations. This study also beat the state-of-the-art performance mAP@0.5 of 0.84 by Banerjee et al. [3] and achieved a state-of-the-art performance mAP@0.5 of 0.91 and mAP@0.5-0.95 of 0.60. The detailed results are tabulated in 5.4. All the experiments were done on **JUDVLP-WBUAFS**: Fishdb-Detection.v1 (head annotations) and **JUDVLP-WBUAFS**: Fishdb-Detection.v2 (whole body annotations) dataset individually using the above experiment protocols with an image size of 416 × 416 and batch size 32. All the other hyperparameters remained the same and fine-tuned on the proposed dataset. The epoch vs mAP@0.5 accuracy graph of all the used YOLO models on the two datasets is presented in Figure 5.6. The ETM approach and TTA used in this study improved the mAP@0.5 score significantly.


Mrigal: Protein-16.19 gm Fat-3.51 gm (per 100 gm)
Catla: Protein-16.18 gm Fat-2.8 gm (per 100 gm)
Rohu: Protein-15.98 gm Fat-2.73 gm (per 100 gm)
Rohu
Mrigal
Catla
(a)
(b)

(a) Sample test image 1

}

(b) sample test image 2

Figure 5.11: Output of the proposed approach for sample test images 1 and 2 (Nutritional facts-(*Protein-maximum, Fat-minimum*)): (a) Head annotations (b) Full box annotations

Precision and recall are important metrics in object detection studies. The precision-recall graph of the two best-performing models, ETM and TTA enabled ensemble of YOLOv3, YOLOv3-SPP, and YOLOv5m, and ETM and TTA enabled ensemble of YOLOv3, YOLOv3-SPP, and YOLOv5l on head annotations and full box annotations is presented in Figure 5.7. In head annotations, average precision and recall of 0.815 and 0.809 were achieved. Looking into fish species-wise, in the Catla, Rohu, and Mrigal categories, precision and recall are both high, which means that maximum ground truth objects were predicted, and the maximum predicted box labels were also correct. Major confusion occurred between the small Mrigal fish and Bata categories. It is the reason for low precision compared to recall in the Mrigal category. In the Bata (Precision-0.824, Recall-0.529), and Grass carp (Precision-1, Recall-0.615) categories, high precision and low recall were seen, which means that all the predicted boxes were correct, but most ground truth objects were missed at prediction time. Low precision and high recall were achieved in the 'Silver carp' category (Precision-0.586, recall-0.909), due to many duplicate boxes being generated around the fish head.

In full box annotations, average precision and recall of 0.662 and 0.745 were achieved. Low precision and high recall were the observations found in the Catla, Rohu, Mrigal, and Silver carp categories. The major reason is that the models in full box annotations predicted many duplicate bounding boxes due to the highly cluttered environment found in the image. Many boxes were predicted, but the maximum number of boxes was wrongly recognized compared to the ground truth label. In the Bata (precision-0.752, recall-0.529), and Grass carp (precision-0.852, recall-0.444) categories, precision was high compared to recall. However, not enough ground truth objects were predicted by the model for these two categories. The ground truth labels and predicted labels of two different validation batches on the head annotations dataset and full box annotation dataset are shown in Figure 5.8 and Figure 5.9.

Applying WBF in the ensemble stage of the TTA-enabled model predictions improved the mAP@0.5 score as well as the prediction label refinement. Based on the WBF technique, the TTA-enabled ensemble of YOLOv3, YOLOv3-SPP, and YOLOv5l achieved an improved mAP@0.5 of 0.79 on the full box annotations dataset. And on the head annotations dataset, using WBF the TTA-enabled ensemble of YOLOv3, YOLOv3-SPP, and YOLOv5m achieved an improved mAP@0.5 of 0.91. In this study, achieving high precision is more important compared to achieving a high recall. Average top precision of 0.98 (without WBF-0.84), 0.91 (without WBF-0.79), and 0.97 (without WBF-0.82) were achieved in Catla, Mrigal, and Rohu categories. Also, the average precision of the 'Silver carp' category was improved from 0.586 to 0.94, whereas, the average precision of Bata and 'Silver carp' got down by some percent. The same happened in the TTA-enabled ensemble of YOLOv3, YOLOv3-SPP, and YOLOv5l in the full box annotation dataset. In Catla, Rohu, and Mrigal categories, top average precision of 0.88 (without WBF-0.64),0.78 (without WBF-0.52), and 0.81 (without WBF-0.62) were achieved. Average precision in the Silver carp category was significantly improved from 0.599 to 0.88. As with the head annotation dataset, the average precision in Bata and Grass carp was down by some percentage.

To keep in mind the accessibility of purchasers, the development of a mobile application is very essential nowadays. In this study, the ensemble of YOLOv3, YOLOv3-SPP, and YOLOv5m, based on WBF and TTA is deployed in a mobile application based on a Flask web server. The system needs a knowledge base where the nutritional information of the fish species available in the dataset is kept. In the back end, the image captured by the purchaser and nutritional needs are taken as input. First, the fish species that meet nutritional requirements are selected automatically from the knowledge base. Next, the deep learning model localizes and identifies only the selected fish species in the image captured by the purchaser. The purchaser can see the bounding boxes and the class label with the nutritional information of each fish species. The purchaser can select the fish that best fits the nutritional needs of many fish detected in the image.

In this chapter, the main aim was to propose a system that can recognize the fish species in a cluttered environment as per the consumer's nutritional needs. The end application asks the user to enter their preferred nutritional facts. Each of the nutritional facts was categorized as 'Maximum' and 'Minimum'. According to the consumer's needs, the end application selects the top three fish species, and then localizes and identifies them. All the other fish species in the image were not considered and treated as background objects. In Figure 5.10, the final output for the nutritional facts *(*Protein-maximum, Calcium-maximum, DHA-maximum) is given. For the sample nutritional facts *(*Protein-maximum, Fat-minimum), the final output of some images is depicted in Figure 5.11. In full box annotations, many duplicate boxes were predicted even after using WBF to ensemble the predictions. However, in head annotations, a very small number of duplicate boxes were predicted by the ensemble model. In a pure cluttered environment, head annotations performed very well compared to the full box annotations. But in some cases, Indian sardines were wrongly recognized as Mrigal. The head part of the Indian oil sardine and Mrigal looks very similar, and in our proposed dataset very few images of Indian oil sardines were present. Sometimes the small size Mrigal and Bata fish heads look quite the same and were wrongly predicted by the proposed approach in some cases. Overall, in the head annotations dataset, the proposed ensemble model based on TTA and WBF performed very well with minimum wrong recognitions and maximum mAP@0.5 accuracy.

When the weighted ensemble was applied to the results obtained from YOLOv3 and YOLOv5, the head and whole body annotations produced descent mAP@0.5 values of 0.91 and 0.79, respectively. The main goal of this study is to develop an application that enables users to choose the type of fish that best suits their specific dietary requirements. In this sense, the most crucial stage in the workflow is detecting and classifying the fish species. Given the cluttered environment, the detection performance is adequate, and the outcomes enable individuals to choose the right fish item on a personal level. When additional fish species are added to the dataset in the future, buyers could choose fish species from the range of species offered on the market.

5.6 Conclusion

In this study, an automatic approach is proposed to localize and identify some fish species in a cluttered environment based on the nutritional needs of the consumer. It is tough for the consumer to remember all the nutritional facts of each species found generally in the Indian fish markets. This study helps the consumer identify different fish in cluttered environments which meet their nutritional needs. The entire work is done using two parts: in the first part, a technique for the localization and identification of fish species in some cluttered environment is developed using an ensemble of different deep-learning object detection algorithms, and in the second part a mobile application for Android (or other) device is developed for the purchasers to choose their nutritional needs and to identify the fish species that meets the nutritional needs. Six different fish species, *Labeo catla* (Catla), *Labeo rohita* (Rohu), *Cirrhinus mrigala* (Mrigal), *Labeo bata* (Bata), *Hypophthalmichthys molitrix* (Silver carp), and *Ctenopharyngodon idella* (Grass carp) is considered in this study. Two different datasets **JUDVLP-WBUAFS**: Fishdb-Detection.v1 (Head annotations) and **JUDVLP-WBUAFS**: Fishdb-Detection.v2 (Full box annotations) were developed and used in this study. Using Weighted Box Fusion (WBF) different versions of YOLOv3, and YOLOv5 were ensembled and Testing Time Augmentation (TTA) was done to achieve more generalized prediction. Experimentally WBF and TTA-enabled ensemble of YOLOv3, YOLOv3-SPP, and YOLOv5m achieved the state-of-art mAP@0.5 of 0.91 on the head annotations dataset and ensemble of YOLOv3, YOLOv3-SPP, and YOLOv5l achieved the state-of-art mAP@0.5 of 0.79 was achieved in full box annotations. Though the results are promising, some challenges need to be solved in future. In the future, the major contribution would be increasing the dataset size by increasing the number of fish species and introducing other fish species. Most of the incorrect recognition of fish species was observed in the Bata and Silver carp categories, because of fewer instances of these categories. Further, the different variations of fish size could be introduced to make a more challenging and generalized dataset. However, this study would open a new wing of research where major contributions would likely come to address the broad problem area.

Chapter 6

Conclusions and Future Work

THIS chapter briefs the entire work done in this thesis, shortcomings, and future scope in the fish species recognition field.

6.1 Conclusion and Future Scope

The work presented in this Thesis has successfully addressed various aspects of automatic carp species recognition using different machine learning and deep learning algorithms under non-cluttered and cluttered environments. For this, four different benchmark datasets, two for non-cluttered environments, i.e. a) **JUDVLP-WBUAFS**: Fishdb-IMC.v1 and b) **JUDVLP-WBUAFS**: Fishdb-EC.v1, and two for cluttered environments, i.e. c) **JUDVLP-WBUAFS:** Fishdb-Detection.v1 and d) **JUDVLP-WBUAFS**: Fishdb-Segmentation.v1 have been developed. The **JUDVLP-WBUAFS**: Fishdb-IMC.v1 dataset consists of three popular major carp in India- Rohu*Labeo rohita*, Catla*Labeo catla*, and Mrigal*Cirrhinus mrigala* in non-cluttered environment, and **JUDVLP-WBUAFS**: Fishdb-EC.v1 is the dataset created for the recognition of three popular exotic carp in India: Common carp*Cyprinus carpio*), Grass carp (*Ctenopharyngodon idella*), and Silver carp(*Hypophthalmichthys molitrix.* In cluttered environment, **JUDVLP-WBUAFS:** Fishdb-Detection.v1 is the dataset developed for the recognition of six different fish species, Catla(*Labeo catla*), Rohu(*Labeo rohita*), Mrigal (*Cirrhinus mrigala*), Bata(*Labeo bata*), Silver carp (*Hypophthalmichthys molitrix*), and Grass carp (*Ctenopharyngodon idella*) by employing object localization and identification approach, and **JUDVLP-WBUAFS**: Fishdb-Segmentation.v1 is the dataset developed for recognizing fish in cluttered environment using a semantic segmentation approach. The development of the dataset is one of the major contributions of the present Thesis. Various techniques are developed for the recognition and segmentation of carp species, depending upon the requirements of the dataset. Among different works, latent representations of the simple autoencoder, deep autoencoder, and deep convolutional autoencoder are extracted as features which are used further for the recognition of three freshwater major carp present in the **JUDVLP-WBUAFS**: Fishdb-IMC.v1 dataset using several machine learning algorithms such as Logistic Regression, Naive Bayes, K-Nearest Neighbor, Support Vector Machine, and Random Forest. Different conventional feature descriptors such as Hu moments, Haralick texture, WLD, and HOG are used to compare the performance with the autoencoder latent features. The deep convolutional autoencoder (DCAE) with Fully connected network

(FCN) achieved maximum 97.33% accuracy and beat the other conventional feature descriptors as well as some state-of-the-art deep learning algorithms like InceptionV3, InceptionResNetV2, MobileNet, VGG16, and VGG19. The FCN classifier gives better performance compared to the best-performing multiclass SVM classifier. Another work for identifying the three exotic carp present in **JUDVLP-WBUAFS**: Fishdb-EC.v1 dataset is performed in this thesis using some of the state-of-the-art deep learning algorithms. More specifically, VGG16, VGG19, InceptionV3, MobileNetV2, and InceptionResNetV2 deep learning networks are applied with the transfer learning approach. Using MobileNetV2, and VGG16 network 99.18% accuracy is obtained on the recognition of the three exotic carp. The effects of different hyperparameters such as momentum, learning rate, batch size, and epochs are also studied during the experiments. In the non-cluttered environment, these two works demonstrated a good performance in automatic fish species recognition and helped various fields as stated in the introduction part. In cluttered environments, typically found in live fish markets, consumers face challenges in recognizing fish. An object localization and identification approach on the **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset using different versions of YOLO algorithms is proposed. Only the fish heads are considered for the recognition of the species, because of their full visibility in the cluttered environments. The YOLOv3-SPP network achieved a mAP@0.5 of 0.84 and demonstrated a state-of-the-art result in this category. An approach based on semantic segmentation is studied to recognize different fish species in some cluttered environment where multiple fish species are present and generally the maximum portion of the body is hidden by the other fish. Using two popular deep learning methods, U-Net, and PSPNet on the **JUDVLP-WBUAFS**: FISHDB-Segmentation.v1 dataset, a mean IoU of 0.76 is achieved using PSPNet with ResNet34 as the backbone. Consumers in the fish markets face problems in recognising fish species that meet their nutritional needs. In the view of consumers need to recognize fish species with the required nutrition to have a balanced diet, different YOLO algorithms are ensembled using the Weighted box fusion (WBF) technique. The optimal weights are chosen experimentally, and Testing time augmentation (TTA) is utilized to improve the performance score. Here, the ensemble of YOLOv3, YOLOv3-SPP, and YOLOv5m using WBF based on the experimentally selected weights and TTA achieved a mAP@0.5 of 0.91 on the head annotation dataset **JUDVLP-WBUAFS**: Fishdb-Detection.v1. To compare the result with the visible body part of the fish, another annotated version of **JUDVLP-WBUAFS**: Fishdb-Detection.v1 is developed named as **JUDVLP-WBUAFS**: Fishdb-Detection.v2. Here, the visible portion of the fish body is annotated using a bounding box, and the same experiments are done. The results on the **JUDVLP-WBUAFS**: Fishdb-Detection.v1 dataset is better compared to the **JUDVLP-WBUAFS**: Fishdb-Detection.v2 dataset, because the head part does not carry redundant information like the whole body given cluttered environments.

While the results of the various studies conducted in this thesis are promising and beneficial for the fishery industry and everyday consumers in the fish market, there are instances of incorrect recognition of certain fish specimens. The misclassification occurs due to some of the inherent challenging conditions like lighting variety in fish markets, placement of fish in a cluttered way, minimum visibility of fish bodies, changes in camera angles, illumination conditions etc. Another limitation in this case is the limited size of the dataset, which is addressed successfully by employing some augmentation techniques to generate a greater variety of samples for the experiments. The presence of these issues facilitated the opportunity to broaden the extent of the investigation concerning various forms of data augmentation utilizing state-of-the-art Generative Adversarial Networks (GANs), Diffusion models, and other related techniques. More fish species could be incorporated into the dataset to enhance the diversity of fish species represented in the dataset. From a technical perspective, various deep learning networks for object localization, segmentation, classification and their ensemble can produce an improved recognition performance. Some fuzzy techniques or bio-inspired techniques can also be applied in the ensemble stage to achieve better performance. Above all, the field of automatic fish species recognition in cluttered environments still has lots of opportunities for further advancements and exploration by researchers. The works presented in this thesis provide a foundation for future researchers to further explore the aforementioned possibilities and beyond.

References

[1] Arnab Banerjee, Arijit Das, Samarendra Behra, Debotosh Bhattacharjee, Nagesh Talagunda Srinivasan, Mita Nasipuri, and Nibaran Das. Carp-dcae: Deep convolutional autoencoder for carp fish classification. *Computers and Electronics in Agriculture*, 196:106810, 2022. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2022.106810. URL https://www.sciencedirect.com/science/article/pii/S0168169922001272.

[2] Arnab Banerjee, Roopsia Chakraborty, Samarendra Behra, Nagesh Talagunda Srinivasan, Debotosh Bhattacharjee, and Nibaran Das. Deep learning based identification of three exotic carps. In Asit Kumar Das, Janmenjoy Nayak, Bighnaraj Naik, S. Vimal, and Danilo Pelusi, editors, *Computational Intelligence in Pattern Recognition*, pages 416–426, Singapore, 2022. Springer Nature Singapore. ISBN 978-981-19-3089-8.

[3] Arnab Banerjee, Debotosh Bhattacharjee, Nibaran Das, Samarendra Behra, and Nagesh Talagunda Srinivasan. Carp-yolo: A detection framework for recognising and counting fish species in a cluttered environment. In *2023 4th International Conference for Emerging Technology (INCET)*, pages 1–7, 2023. doi: 10.1109/INCET57972.2023.10170475.

[4] Arnab Banerjee, Debotosh Bhattacharjee, Nagesh Talagunda Srinivasan, Samarendra Behra, and Nibaran Das. Segfishhead: A semantic segmentation approach for the identification of fish species in a cluttered environment. In *2023 International Conference on Computer, Electronics & Electrical Engineering their Applications (IC2E3)*, pages 1–6, 2023. doi: 10.1109/IC2E357697.2023.10262432.

[5] Nathalie Castignolles, Michel Cattoen, and M. Larinier. Identification and counting of live fish by image analysis. In Sarah A. Rajala and Robert L. Stevenson, editors, *Image and Video Processing II*, volume 2182, pages 200 – 209. International Society for Optics and Photonics, SPIE, 1994. doi: 10.1117/12.171067. URL https://doi.org/10.1117/12.171067.

[6] B. Zion, A. Shklyar, and I. Karplus. In-vivo fish sorting by computer vision. *Aquacultural Engineering*, 22(3):165–179, 2000. ISSN 0144-8609. doi: https://doi.org/10.1016/S0144-8609(99)00037-0. URL https://www.sciencedirect.com/science/article/pii/S0144860999000370.

[7] D.J. Lee, S. Redd, R. Schoenberger, Xiaoqian Xu, and Pengcheng Zhan. An automated fish species classification and migration monitoring system. In *IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society (IEEE Cat.*

No.03CH37468), volume 2, pages 1080–1085 Vol.2, 2003. doi: 10.1109/IECON.2003.1280195.

[8] M.S. Nery, A.M. Machado, M.F.M. Campos, F.L.C. Padua, R. Carceroni, and J.P. Queiroz-Neto. Determining the appropriate feature set for fish classification tasks. In *XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)*, pages 173–180, 2005. doi: 10.1109/SIBGRAPI.2005.25.

[9] Dah-Jye Lee, James K. Archibald, Robert B. Schoenberger, Aaron W. Dennis, and Dennis K. Shiozawa. *Contour Matching for Fish Species Recognition and Migration Monitoring*, pages 183–207. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-78534-7. doi: 10.1007/978-3-540-78534-7_8. URL https://doi.org/10.1007/978-3-540-78534-7_8.

[10] Rasmus Larsen, Hildur Olafsdottir, and Bjarne Kjær Ersbøll. Shape and texture based classification of fish species. In Arnt-Børre Salberg, Jon Yngve Hardeberg, and Robert Jenssen, editors, *Image Analysis*, pages 745–749, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-02230-2.

[11] Mutasem Alsmadi, Khairuddin Omar, Shahrul Azman Mohd Noah, and Ibrahim Almarashdeh. Fish recognition based on robust features extraction from size and shape measurements using neural network. *Journal of Computer Science*, 6, 12 2010. doi: 10.3844/jcssp.2010.1088.1094.

[12] Jing Hu, Daoliang Li, Qingling Duan, Yueqi Han, Guifen Chen, and Xiuli Si. Fish species classification by color, texture and multi-class support vector machine using computer vision. *Computers and Electronics in Agriculture*, 88:133–140, 2012. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2012.07.008. URL https://www.sciencedirect.com/science/article/pii/S0168169912001937.

[13] Mohamed Mostafa M. Fouad, Hossam M. Zawbaa, Nashwa El-Bendary, and Aboul Ella Hassanien. Automatic nile tilapia fish classification approach using machine learning techniques. In *13th International Conference on Hybrid Intelligent Systems (HIS 2013)*, pages 173–178, 2013. doi: 10.1109/HIS.2013.6920477.

[14] Lian Li and Jinqi Hong. Identification of fish species based on image processing and statistical analysis research. In *2014 IEEE International Conference on Mechatronics and Automation*, pages 1155–1160, 2014. doi: 10.1109/ICMA.2014.6885861.

[15] Takeshi Saitoh, T. Shibata, and Tsubasa Miyazono. Feature points based fish image recognition. *International Journal of Computer Information Systems and Industrial Management Applications*, 8:12–22, 03 2016.

[16] Francesco Rossi, Alfredo Benso, Stefano Di Carlo, Gianfranco Politano, Alessandro Savino, and Pier Luigi Acutis. Fishapp: A mobile app to detect fish falsification through image processing and machine learning techniques. In *2016 IEEE*

International Conference on Automation, Quality and Testing, Robotics (AQTR), pages 1–6, 2016. doi: 10.1109/AQTR.2016.7501348.

[17] Muhammad Naufal Rachmatullah and Iping Supriana. Low resolution image fish classification using convolutional neural network. In *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, pages 78–83, 2018. doi: 10.1109/ICAICTA.2018.8541313.

[18] Alexis Joly, Hervé Goëau, Hervé Glotin, Concetto Spampinato, Pierre Bonnet, Willem-Pier Vellinga, Robert Planqué, Andreas Rauber, Simone Palazzo, Bob Fisher, and Henning Müller. Lifeclef 2015: Multimedia life species identification challenges. In Josanne Mothe, Jacques Savoy, Jaap Kamps, Karen Pinel-Sauvagnat, Gareth Jones, Eric San Juan, Linda Capellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 462–483, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24027-5.

[19] Alaa Tharwat, Ahmed Abdelmonem Hemedan, Aboul Ella Hassanien, and Thomas Gabel. A biometric-based model for fish species classification. *Fisheries Research*, 204:324–336, 2018. ISSN 0165-7836. doi: https://doi.org/10.1016/j.fishres.2018.03.008. URL https://www.sciencedirect.com/science/article/pii/S0165783618300821.

[20] Muhammad Ather Iqbal Hussain, Zhi-Jie Wang, Zain Ali, and Shazia Riaz. Automatic fish species classification using deep convolutional neural networks. *Wireless Personal Communications*, 116, 01 2019. doi: 10.1007/s11277-019-06634-1.

[21] Kaneswaran Anantharajah, ZongYuan Ge, Chris McCool, Simon Denman, Clinton Fookes, Peter Corke, Dian Tjondronegoro, and Sridha Sridharan. Local inter-session variability modelling for object classification. In *IEEE Winter Conference on Applications of Computer Vision*, pages 309–316, 2014. doi: 10.1109/WACV.2014.6836084.

[22] Alexis Joly, Hervé Goëau, Hervé Glotin, Concetto Spampinato, Pierre Bonnet, Willem-Pier Vellinga, Robert Planqué, Andreas Rauber, Simone Palazzo, Bob Fisher, and Henning Müller. Lifeclef 2015: Multimedia life species identification challenges. In Josanne Mothe, Jacques Savoy, Jaap Kamps, Karen Pinel-Sauvagnat, Gareth Jones, Eric San Juan, Linda Capellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 462–483, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24027-5.

[23] Francis Jesmar P. Montalbo and Alexander A. Hernandez. Classification of fish species with augmented data using deep convolutional neural network. In *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, pages 396–401, 2019. doi: 10.1109/ICSEngT.2019.8906433.

[24] Hafiz Tayyab Rauf, M. Ikram Ullah Lali, Saliha Zahoor, Syed Zakir Hussain Shah, Abd Ur Rehman, and Syed Ahmad Chan Bukhari. Visual features based automated identification of fish species using deep convolutional neural networks. *Computers and Electronics in Agriculture*, 167:105075, 2019. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2019.105075.

[25] Phoenix X. Huang, Bastiaan J. Boom, and Robert B. Fisher. Underwater live fish recognition using a balance-guaranteed optimized tree. In Kyoung Mu Lee, Yasuyuki Matsushita, James M. Rehg, and Zhanyi Hu, editors, *Computer Vision – ACCV 2012*, pages 422–433, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-37331-2.

[26] Concetto Spampinato, D. Giordano, Roberto Di Salvo, Jessica Chen-Burger, Robert Fisher, and Gayathri Nadarajan. Automatic fish classification for underwater species behavior understanding. *Analysis and Retrieval of Tracked Events and Motion in Imagery Streams*, 10 2010. doi: 10.1145/1877868.1877881.

[27] Jorge Cabrera-Gámez, Modesto Castrillón Santana, Antonio Carlos Domínguez-Brito, Daniel Hernández-Sosa, Josep Isern-Gonzalez, and Javier Lorenzo-Navarro. Exploring the use of local descriptors for fish recognition in lifeclef 2015. In Linda Cappellato, Nicola Ferro, Gareth J. F. Jones, and Eric SanJuan, editors, *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015*, volume 1391 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015. URL https://ceur-ws.org/Vol-1391/49-CR.pdf.

[28] Kai Hu, Chenghang Weng, Yanwen Zhang, Junlan Jin, and Qingfeng Xia. An overview of underwater vision enhancement: From traditional methods to recent deep learning. *Journal of Marine Science and Engineering*, 10(2), 2022. ISSN 2077-1312. doi: 10.3390/jmse10020241. URL https://www.mdpi.com/2077-1312/10/2/241.

[29] Xiu Li, Min Shang, Hongwei Qin, and Liansheng Chen. Fast accurate fish detection and recognition of underwater images with fast r-cnn. In *OCEANS 2015 - MTS/IEEE Washington*, pages 1–5, 2015. doi: 10.23919/OCEANS.2015.7404464.

[30] Ahsan Jalal, Ahmad Salman, Ajmal Mian, Mark Shortis, and Faisal Shafait. Fish detection and species classification in underwater environments using deep learning with temporal information. *Ecological Informatics*, 57:101088, 2020. ISSN 1574-9541. doi: https://doi.org/10.1016/j.ecoinf.2020.101088. URL https://www.sciencedirect.com/science/article/pii/S1574954120300388.

[31] Jonas Jäger, Erik Rodner, Joachim Denzler, Viviane Wolff, and Klaus Fricke-Neuderth. Seaclef 2016: Object proposal classification for fish detection in underwater videos. In *Conference and Labs of the Evaluation Forum*, 2016. URL https://api.semanticscholar.org/CorpusID:9252325.

[32] Monika Mathur, Diksha Vasudev, Sonalika Sahoo, Divanshi Jain, and Nidhi Gooel. Crosspooled fishnet: transfer learning based fish species classification model. *Multimedia Tools and Applications*, 79, 11 2020. doi: 10.1007/s11042-020-09371-x.

[33] Monika Mathur and Nidhi Gooel. Fishresnet: Automatic fish classification approach in underwater scenario. *SN Computer Science*, 2, 07 2021. doi: 10.1007/s42979-021-00614-8.

[34] Zhixue Zhang, Xiujuan Du, Long Jin, Wang Shuqiao, Lijuan Wang, and Xiuxiu Liu. Large-scale underwater fish recognition via deep adversarial learning. *Knowledge and Information Systems*, 64:1–27, 02 2022. doi: 10.1007/s10115-021-01643-8.

[35] Marius Paraschiv, Ricardo Padrino, Paolo Casari, Eyal Bigal, Aviad Scheinin, Dan Tchernov, and Antonio Fernández Anta. Classification of underwater fish images and videos via very small convolutional neural networks. *Journal of Marine Science and Engineering*, 10(6), 2022. ISSN 2077-1312. doi: 10.3390/jmse10060736. URL https://www.mdpi.com/2077-1312/10/6/736.

[36] Hongwei Qin, Xiu Li, Jian Liang, Yigang Peng, and Changshui Zhang. Deepfish: Accurate underwater live fish recognition with a deep architecture. *Neurocomputing*, 187:49–58, 2016. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2015.10.122. URL https://www.sciencedirect.com/science/article/pii/S0925231215017312. Recent Developments on Deep Big Vision.

[37] Xin Sun, Junyu Shi, Junyu Dong, and Xinhua Wang. Fish recognition from low-resolution underwater images. In *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 471–476, 2016. doi: 10.1109/CISP-BMEI.2016.7852757.

[38] R. Froese and D. Pauly. Editors. 2021. fishbase. world wide web electronic publication. URL www.fishbase.org.

[39] D.J. Lee, S. Redd, R. Schoenberger, Xiaoqian Xu, and Pengcheng Zhan. An automated fish species classification and migration monitoring system. In *IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society (IEEE Cat. No.03CH37468)*, volume 2, pages 1080–1085 Vol.2, 2003. doi: 10.1109/IECON.2003.1280195.

[40] Muharrem Mercimek, Kayhan Gulez, and Tarik Mumcu. Real object recognition using moment invariants. *Sadhana*, 30:765–775, 12 2005. doi: 10.1007/BF02716709.

[41] Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3 (6):610–621, 1973. doi: 10.1109/TSMC.1973.4309314.

[42] Jie Chen, Shiguang Shan, Chu He, Guoying Zhao, Matti Pietikäinen, Xilin Chen, and Wen Gao. Wld: A robust local image descriptor. *IEEE Transactions on Pattern*

Analysis and Machine Intelligence, 32(9):1705–1720, 2010. doi: 10.1109/TPAMI.2009.155.

[43] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005. doi: 10.1109/CVPR.2005.177.

[44] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and ZB Wojna. Rethinking the inception architecture for computer vision. 06 2016. doi: 10.1109/CVPR.2016.308.

[45] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *AAAI Conference on Artificial Intelligence*, 02 2016.

[46] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. pages 4510–4520, 06 2018. doi: 10.1109/CVPR.2018.00474.

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

[48] R. Froese and D. Pauly. Fishbase. world wide web electronic publication, 2024.

[49] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):6999–7019, 2022. doi: 10.1109/TNNLS.2021.3084827.

[50] Rafael Garcia, Ricard Prados, Josep Quintana, Alexander Tempelaar, Nuno Gracias, Shale Rosen, Håvard Vågstøl, and Kristoffer Løvall. Automatic segmentation of fish using deep learning with application to fish size measurement. *ICES Journal of Marine Science*, 77(4):1354–1366, 10 2019. ISSN 1054-3139. doi: 10.1093/icesjms/fsz186. URL https://doi.org/10.1093/icesjms/fsz186.

[51] D. MoFAHD. Handbook of fisheries statistics, 2022.

[52] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018. URL https://arxiv.org/abs/1804.02767.

[53] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, (Zeng Yifu), Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation, November 2022. URL https://doi.org/10.5281/zenodo.7347926.

[54] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001. doi: 10.1109/CVPR.2001.990517.

[55] Yoav Freund and Robert E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In Paul Vitányi, editor, *Computational Learning Theory*, pages 23–37, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg. ISBN 978-3-540-49195-8.

[56] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005. doi: 10.1109/CVPR.2005.177.

[57] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014. URL https://arxiv.org/abs/1311.2524.

[58] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. URL https://arxiv.org/abs/1506.01497.

[59] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018. URL https://arxiv.org/abs/1703.06870.

[60] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection, 2017. URL https://arxiv.org/abs/1712.00726.

[61] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016. URL https://arxiv.org/abs/1506.02640.

[62] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. *SSD: Single Shot MultiBox Detector*, page 21–37. Springer International Publishing, 2016. ISBN 9783319464480. doi: 10.1007/978-3-319-46448-0_2. URL http://dx.doi.org/10.1007/978-3-319-46448-0_2.

[63] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018. URL https://arxiv.org/abs/1708.02002.

[64] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection, 2020. URL https://arxiv.org/abs/1911.09070.

[65] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger, 2016. URL https://arxiv.org/abs/1612.08242.

[66] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation, 2018. URL https://arxiv.org/abs/1803.01534.

[67] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, Atlanta, GA, USA, 2013.

[68] Richard E. Bellman. *Adaptive Control Processes: A Guided Tour.* Princeton University Press, Princeton, NJ, USA, 1972. ISBN 9780691079516.

[69] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, Munich, Germany, 2015. Springer. doi: 10.1007/978-3-319-24574-4_28.

[70] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890. IEEE, 2017. doi: 10.1109/CVPR.2017.308.

[71] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. *Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations*, page 240–248. Springer International Publishing, 2017. ISBN 9783319675589. doi: 10.1007/978-3-319-67558-9_28. URL http://dx.doi.org/10.1007/978-3-319-67558-9_28.

[72] B Zion, A Shklyar, and I Karplus. Sorting fish by computer vision. *Computers and Electronics in Agriculture*, 23(3):175–187, 1999. ISSN 0168-1699. doi: https://doi.org/10.1016/S0168-1699(99)00030-7. URL https://www.sciencedirect.com/science/article/pii/S0168169999000307.

[73] Israt Sharmin, Nuzhat Farzana Islam, Israt Jahan, Tasnem Ahmed Joye, Md. Riazur Rahman, and Md. Tarek Habib. Machine vision based local fish recognition. *SN Applied Sciences*, 1(12):1529, Nov 2019. ISSN 2523-3971. doi: 10.1007/s42452-019-1568-z. URL https://doi.org/10.1007/s42452-019-1568-z.

[74] Harshit Singh Chhabra, Akshay Kumar Srivastava, and Rahul Nijhawan. A hybrid deep learning approach for automatic fish classification. In Pradeep Kumar Singh, Bijaya Ketan Panigrahi, Nagender Kumar Suryadevara, Sudhir Kumar Sharma, and Amit Prakash Singh, editors, *Proceedings of ICETIT 2019*, pages 427–436, Cham, 2020. Springer International Publishing. ISBN 978-3-030-30577-2.

[75] Baidya Nath Paul, Narasimhan Sridhar, Soumen Chanda, GS Saha, and Shiba Shankar Giri. Nutrition information of cirrhinus mrigala (mrigal), 2016. URL http://cifa.nic.in/sites/default/files/Mrigal_Pamplet.pdf.

[76] Baidya Nath Paul, Narasimhan Sridhar, Soumen Chanda, GS Saha, and Shiba Shankar Giri. Nutrition information of labeo rohita (rohu), 2016. URL http://cifa.nic.in/sites/default/files/Rohu_Pamphlet.pdf.

[77] Baidya Nath Paul, Narasimhan Sridhar, Soumen Chanda, GS Saha, and Shiba Shankar Giri. Nutrition information of labeo catla (catla), 2016.

[78] United States DEPARTMENT OF AGRICULTURE. Usda national nutrient database. URL https://fdc.nal.usda.gov/.

[79] Muhammad Ashraf, Asma Zafar, Abdul Rauf, Shahid Mahboob, Naureen, and Naureen Qureshi. Nutritional values of wild and cultivated silver carp (hypophthalmichthys molitrix) and grass carp (ctenopharyngodon idella). *International Journal of Agriculture and Biology*, 13:1560–8530, Mar 2011.

[80] Renata Pyz-Lukasik and Danuta Kowalczyk-Pecka. Fatty acid profile of fat of grass carp, bighead carp, siberian sturgeon, and wels catfish. *Journal of Food Quality*, 2017: 5718125, Dec 2017. doi: 10.1155/2017/5718125. URL https://doi.org/10.1155/2017/5718125.

Arnab Banerjee
30.07.24