# Segmentation and Classification of Fish Species Using Deep Learning

by

Faiaz Ibnee Rahman
21101151
Nahiyan Rahman Talukdar
21101273
Zaion Abrar Akhand
21101315

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
February 2025

# Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

**Student's Full Name & Signature:**

Faiaz Ibnee Rahman
21101151

Nahiyan Rahman Talukder
21101273

Zaion Abrar Akhand
21101315

# Approval

The thesis titled "Segmentation and Classification of Fish Species Using Deep Learning" submitted by

1. Faiaz Ibnee Rahman (21101151)
2. Nahiyan Rahman Talukder (21101273)
3. Zaion Abrar Akhand (21101315)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on February 3, 2025.

**Examining Committee:**

Supervisor:
(Member)

Md. Tanzim Reza
Senior Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

# Abstract

Fish Species Classification using deep learning surfaced as a formidable tool for automating and enhancing species identification in aquatic ecosystems. Moreover, leveraging Convolutional Neural Networks (CNNs), this approach presents us with an efficient and accurate way of identifying and categorizing fish under different species based on image data, providing us with a notable advancement for aquatic biodiversity surveillance and fisheries management. The switch towards deep learning addresses the constraints of traditional techniques such as manual labor and morphological analysis, which can be time consuming, require specialist knowledge and are inclined to human error. The primary objective of our study reviews recent advances in the field of deep learning, which focus on Convolutional Neural Networks and their application in classifying fish species as well as segmenting them through images of fishes overwater and maintaining a white background. We analyzed various approaches adopted in recent research, from CNN-based models like ResNet-50, Inception-v3, YOLOv11 etc. to innovative image preprocessing techniques, highlighting the evolution of methodologies from rudimentary to more sophisticated automated systems. We introduced a custom dataset of 7100 images which were captured during daylight as well as night time for better picture quality keeping in mind the reduction of computational intensity for real-time applications. In addition, through this research our aim is not only to refine the accuracy of fish species classification and segmentation but also to contribute significantly to the protection of marine ecosystems, aiding in the detection of rare species and assisting in sustainable fishing practices. To conclude, this study stands at the forefront of technological advancements in ecological conservation and offers in industrial use.

Key Words: Deep Learning, Classification, Segmentation, CNN, Pre-Trained Models, ResNet-50, Inception-v3, YOLO Models, Custom Dataset.

# Table of Contents

# Chapter 1

# Introduction

## 1.1 Introduction

Fish classification is the process through which a large scale of fishes are categorized into their respective species of fish called The Animal Kingdom. In addition, during this life cycle of fish classification, there are numerous types of fishes available in the ecosystem, one of the classes of fish is the Chordata phylum, which encompasses animals with a notochord, dorsal nerve cord, and other specific features during their life cycle. Moreover, the Animal Kingdoms are further divided into different classes for example Agnatha for primitive, jawless fish like lampreys and hagfish; Chondrichthyes for leathery species such as sharks, rays, and skates and Osteichthyes, which includes the majority of fish species with bony skeletons. Although, these classes are subdivided into various orders, families, and genera based on increasingly specific shared traits culminating in the species level, which identifies groups of individuals capable of interbreeding and producing fertile offspring this hierarchical structure helps in understanding the relationships and evolutionary history of fish, allowing for effective identification, study, and conservation. However, this classification system is dynamic and evolving as new research and genetic analysis provide fresh insights.

Fish species found in the world's waters, like oceans, rivers, and lakes, represent a wide range of aquatic creatures. Moreover, the range from tiny fish is barely visible to the naked eye to the largest sea mammals. Even though, the diversity looks incredible according to their size and roles in their natural habitats. In addition, the fish are an essential part of the ecosystems, acting as hunters and hunted and helping to keep the balance in both the saltwater and freshwater environments. According to the fish type, the environment is suitable even if the situation is in the bright coral reefs or in the deep dark ocean. Moreover, the fishes have different gills, fins, and senses to survive in these varied places, thus proving the suitable environment the fishes are living in. Although these underwater inhabitants are not just important for the health of the planet, they are also very crucial for people and the whole world for food, jobs, cultural heritage, and research purposes. Most importantly, understanding and properly categorizing these fish species into the right environment is the key to protecting them as this will help us keep track of how the fishes are doing in order to survive the environment, saving species from risk, and ensure the fish responsibly safe, especially as our environment changes.

The main goal of our research topic is to understand the classification, segmentation, and conservation fish species and offer industrial usage. In addition, the Animal Kingdom can categorize a vast array of fish into their respective species by studying their roles in ecosystems and recognizing their importance to both the environment and human societies. Moreover, the research focuses on monitoring the health of the fish populations by protecting endangered species and maintaining sustainable fishing practices. This is achieved by examining the diverse traits and adaptations of fish in various aquatic environments from coral reefs to deep oceans. Additionally, the research is dynamic and evolves with new insights from ongoing studies and genetic analysis, highlighting the importance of adapting conservation efforts in response to changing environmental conditions.

## 1.2 Problem Statement

This research field is filled with lots of opportunities to improve as well as many problems and limitations. For this classification and segmentation problem, using deep learning faces several challenges.

- Limited high-quality sample size of the dataset
- Different water colors
- Illumination underwater
- Obstacles underwater like coral, reef etc.
- Difficulties in image processing
- Non-diverse dataset causes over fitting
- Complexity for real-time processing in practical applications for complex models
- Limited uses of updated models like YOLOv11
- Less accuracy on complex dataset
- Limited dataset with lack of diversity in species and high quality image on Bangladeshi fishes

Fish species identification is an essential part of our biological environment. Many researchers have already worked hard to make this research automated. It was a challenging issue due to some of the above-mentioned cases. Larger datasets are rare for this classification. The model could not properly train so sometimes accuracy fell though the approach was pretty good. Again in some research image quality was poor which significantly hindered the model's ability to learn diverse and comprehensive features. Some research was based on around 71,28,000 images and quality was not top-notch in some cases.

In this classification most data is underwater. We know the situation underwater is not the same for all places. So it is very tough to train models with the same fish in different types of watercolors. The color of water varies widely depending on its depth, geographical location, presence of algae or plankton, and time of day. These differences can significantly alter the appearance of fish in photographs, adding difficulty to the categorization process. Some authors went back from these difficulties and chose a plain background to capture fish images which is not effective in real-life applications. Though to overcome this they choose image preprocessing techniques which could hamper the accuracy.

Illumination underwater is also a destruction for collecting images. Due to sunlight, it creates many reflections inside water. Underwater, light behaves strangely than on the surface. Water absorbs and scatters light, creating unique lighting issues. The deeper you go, the less light enters, having a substantial impact on visibility and image quality. This issue is also responsible for low accuracy.

The undersea environment is packed with various natural structures including coral reefs, boulders, and aquatic plants. These features can impede views of fish, resulting in incomplete photos or photographs with the fish not separated from the backdrop. This makes it challenging for a deep-learning model to distinguish and target the fish. To remove this obstacle researchers had to handle this issue.

Processing underwater photographs presents a unique set of issues. These photos and videos frequently exhibit motion blur caused by the movement of water and fish, as well as low contrast, color deterioration, and noise. The author had to convert them into negative, gray images though It has the potential to reduce accuracy.

Non-diverse dataset is also a matter of concern. In many situations, the available datasets lack diversity in terms of species, ambient circumstances, and photographic viewpoints. This lack of variety might cause deep learning models to overfit the training data, resulting in high performance on previously seen data but poor performance on fresh, unseen photos. This is a critical issue in real-world applications because the model will be exposed to a wide range of environments and species.

Deep learning model's computational intensity makes them difficult to implement in real-time applications such as underwater robots or automated monitoring systems. More complex model needs more time sometimes which can't do the real-time processing. Usages of the updated models, and activation functions can help to make this thing faster with good performance. These are some of the problems from the literature review of different authors. Higher Accuracy can be achieved if this problem can be solved.

## 1.3 Research Objective

In today's world fish species classification and segmentation using deep learning has great significance because of the implications for biodiversity conservation, sustainable fisheries management, and environmental preservation. Human activities in the environment are becoming more dangerous for marine ecosystems. Monitoring and classifying fish species is becoming much more difficult. Here, the water pollution plays a big negative role in the ecosystem. As a result, the existence of many rare species is going out of our reach. Rare species need to be found and separated immediately with the help of deep learning. Again, it makes it easier for fishermen to precisely sort their catch and minimize overfishing to guarantee that conservation restrictions are followed and support the recovery of vulnerable species. Besides, automated fish sorting without human involvement can also play a vital role in the industry. We plan to use deep learning here to overcome these problems and ensure a balanced biodiversity for underwater lives.

# Chapter 2

# Literature Review

## 2.1 Literature Review

Salman et al. [8] proposed a two-step process in order to detect and classify fish. In their research, they extracted images of different fishes from a video captured off of the coastal area of Western Australia. Stereo BRUVs are used in order to capture the underwater footage from which the images are extracted. A total of 16 species of fish were collected during the analysis of the footage. Other than that they used another species class namely "Others" for the objects that are not of interest. Their images are cropped down to a size of 224 X 224 pixels for processing. Here the two steps that they followed are using a Convolutional Neural Network (CNN) for object detection and a support vector Machine (SVM) for classification. Three models namely AlexNet, VGGNet, and ResNet of CNN are used in order to test and train the dataset. An accuracy of 94.3% for classification was recorded by excluding the "Others" species class, which is on par with the identification rate of human experts while on the other hand including the same class had a classification accuracy of 89%. The model of ResNet, from layer 4b15 and 4b20 cross-pooling activations, with a local region size of 3 X 3 and training an SVM on the cross-pooled features, achieved an accuracy of 89.0% in terms of classification with the "Others" species class present. In comparison, the AlexNet and VGGNet models produced remarkably degenerated results having 65.8% and 78.1%. The proposed method of CNN as well as SVM had an accuracy of 89% on the dataset collected by Salman et al. .[8] while the dataset by LifeClef'15 had an accuracy of 96.73%. Through the process of cutting the size of the sum of features from several layers in CNN, we may analyze more detailed portions of a picture without using too much memory.

Knausgaard et al. [26], in their research paper, used a different type of approach in order to classify and detect temperate fishes underwater. They extracted images from video footage as well as tried to use live video in order to detect fish and classify them. Their concept is to detect the fish, and then discard the background of the fish in order to classify the fish species using a CNN with Support Vector Machine (SVM). The datasets that were used in this paper are from Fish4Knowledge which consists of 27230 images from these 23 different species exists. YOLO (You Only Look Once) is used for the detection of fish and the SE (Squeeze and Excitation) architecture of CNN is used for the classification of the fish species. The models are trained separately in order to detect and classify fishes. YOLOv3 is used for fish detection whereas CNN - SENet is used for classification. The obtained accuracy is 83.68% for the temperate fish. Fish detection has an accuracy of 86.96% whereas the CNN SENet architecture without any data augmentation as well as without using any image pre-processing, achieves an accuracy of 99.27% which is a great success.

The research paper by Vaneeda et al. [9] focuses on training a convolutional neural network (CNN) using synthetic images in order to fish species. Three types of species namely herring, mackerel, and blue whiting are focused here, each species having a thousand images from surveys but due to splitting the dataset into training, validation, and tests left the researchers with a few hundred images which is low for training a CNN. The researchers proposed creating large amounts of artificial fish images by using small amounts of actual fish images of an individual fish. All the images had to undergo different orientations such as flipping, rotations, etc. TensorFlow deep learning framework with the Keras library was used in this research. Keras library is used with the help of which is pre-trained on the ImageNet classification dataset. In order to discard some parameters as well as to prevent overfitting problems they introduced a global pooling layer and a dropout layer. The inception layer consists of 1000 nodes and it is fully connected where each node represents a class in the dataset of ImageNet. 133 different training datasets were used based on the configuration of total fish images that were cropped and the total trained images. They also compared the training of artificial data with the training of real images, From each species around 70 images were selected for training by CNN with these images as training datasets. Some techniques which include different orientations of a picture like shearing, translation, rotation, flipping, zooming, etc implemented on the final images that were both real and artificial. After training, the network was tested on a test dataset which consisted of 2400 actual images where 800 images from each species, using the saved model that corresponds to a little validation loss. The test accuracy for this was 67.2%. Misclassification is also observed while using the most accurate classifier.

This research paper by Hridayami et al. .[10] presents a fish classification method using one of the CNN architectures, a pre-trained VGG16 using transfer learning methods on ImageNet. The dataset of fish in this paper includes 50 species, there are 15 images for every species. Of these 15 images, 10 were used for training and the remaining were used for testing the model. In this paper, Arthur used 4 different datasets for training the model. Their datasets are color space images with RGB, canny filter images, blending images, and mixed RGB images with blending images. The trained model blending image mixed with RGB image has a big accuracy of 96.4%. The lowest accuracy was 75.6% of the blending image. On the other hand, the GAR of the canny filter model was 80.4%, and for RGB color space was 80.4%. From this paper, it is concluded that having a limited number of images in datasets can be overcome by image enhancements and using pre-trained models on ImageNet. Which can bring great accuracy in this kind of challenging dataset.

In this paper, the researcher Mohanty et al. [28] basically used machine learning methods. In addition, the models used by the researchers PyTorch and TensorFlow. Moreover, PyTorch was particularly noted for its more accurate predictions due to its incorporation of neural networks with more layers. However, Mohanty et al. [5]

used a dataset from a Kaggle Competition, consisting of 18,000 fish images with a file size of 2.76 KB. The data was split into training tests and validation sets. However, the TensorFlow split was 5040 training images, 2700 test images, and 1260 validation images and for PyTorch, it was 7290 training images, 900 test images, and 810 validation images. They developed separate Python codes using PyTorch and TensorFlow and compared their performance. It was found that PyTorch provided a more accurate prediction, with training and validation accuracies reaching 99.75%, compared to TensorFlow's 87.22%. The increased accuracy was attributed to the more complex neural network structure in the PyTorch model, which included more layers. Even Though, the model was modified by resizing, augmentation, and normalization.

Rathi et al. [3] proposed CNN and deep learning methods for underwater fish species classification. The dataset consists of 27142 images of 23 fish species. In their proposal, they introduced otsu thresholding and erosion for image preprocessing. They produced a black-white image for training. They have used three different activation functions. These are ReLu, tanh, Softmax using CNN. Three layers have been used here. They apply convolutional operations to the input image, extracting features by using filters or kernels. Then pooling layers are used to reduce the spatial dimensions. At the last fully connected layers are used, where every input is connected to every output of the previous layer. For the accuracy part, a best of 96.29% was achieved by ReLu. However, Tanh achieved 72.62% which is much less compared to ReLu. Softmax has achieved only 61.91% which is the lowest. However, this model has used more sample data than previous research. This research faced difficulties due to water obstacles and the illumination underwater. Additionally, Making image negation can lower the accuracy rate of the classification. Instead of applying negative image transfer learning could help in accuracy more. so the author assured us to work further for image enhancement.

Khalifa et al. [5] completed their research on aquarium family fish species identification using deep neural networks. In the dataset used from QUT Robotics fish, a camera was set up beside the aquarium when a fish passes it captures pictures. Their dataset consists of 8 fish species with 191 subspecies. Arthur used a raw dataset directly as input. The model was developed with 4 layers including two fully connected layers and two convolution layers. The proposed model is a simplified version of AlexNet which is designed to reduce computational complexity and memory requirements while maintaining the high accuracy percentage. The first layer is for input, the second is used for convolution, the third is for ReLu as an activation function and again the fourth layer is convolution. The fully connected layer has 256 neurons and at last, also used softmax to obtain class membership as illustrated. The accuracy of the proposed model was 85.59% which is a little low compared with other models like AlexNet, Vgg16, and Vgg19. Arthur claimed that the accuracy was low because the proposed model has fewer trained images, less memory, and less computational complexity in the training, validation, and testing phases. The problem with this model is fewer sample images and low accuracy. Grid search could help to find the optimal set of parameters. Updated architectures like DenseNet, and ResNet could create an impact on accuracy. In real life, there will be illumination in water. This type of data is neglected here which is not real-life-based research.

A different approach to fish species classification has been shown by the team of Olsvik et al. [11] They have applied Squeeze-and-Excitation architecture to discard image pre-filtering. They claim that this filter hampers the ability to detect species accurately. For the pre-training part, the Input size is 200*200*3 and the dataset consists of 19149 pictures with batch size 50. Before applying the squeeze they completed batch normalization. The result with batch normalization of CNN-SENet is higher which is 99.27% and the time one epoch is 197s. Besides SE they also compared the accuracy with Inception-V3, ResNet-50, and Inception-ResNet-V2. Now for the post training, only 712 images of 4 species have been used and the batch size was 8. Weight from the pre-training was applied to this post and the accuracy was evaluated using the weights from the final epoch. Post training accuracy 83.68% with CNN-SENet Squeeze-and-Excitation. The accuracy is not expected as the benefits of the architecture. However, the problem in this paper is very deep. The transfer learning process is good until there aren't many differences in terms of quantity and environment between pre and post-training. The accuracy is lower because of this huge variation in datasets. Maybe post-training could give a good accuracy if the dataset was not much different.

This is the research paper of Kaya et al. [32] where they implemented a model of deep-learning fish species classification. The dataset consists of 8000 images per 1000 images per species. These images have been converted into grayscale. Here ResNet50, ResNet101, and VGG16 models, a new model called IsVoNet8 was proposed. 50, 101, 41, and 18 layers have been used respectively. Here they have used convolution, maximum pooling, dropout, ReLU, fully connected, and classification layers. In training Adamax optimized was used here including 20 epochs and 32 batch size. VoNet8 is the winner here in terms of accuracy with 98.62% accuracy. ResNet50 has 91.37%, 86.12% is the accuracy for resNet101. VGG has 98.62% accuracy. VoNet8 has fewer parameters and the cost is low. In terms of problems, the quality and variety of images in the dataset are noticeable. Due to the complexity, there is a chance of overfitting. The regularization technique could be applied here. The usage of ensemble methods, by this overall accuracy of multiple models, could be combined.

According to the paper of Kratzert et al. [6] showed research on fish species classification in underwater video monitoring using CNN. The dataset was used of 10 fish species with 8099 individuals. For image preprocessing Author used FishNet software which automatically extracts images from video and crops also rotates images to ensure horizontal image. 80% for training, 5% for the validation set, and 15% for the test set have been used. Four different network configurations have been used these are Vgg16, Vgg16+ fish length, Vgg+ migration date, Vgg16+ fish length, and Migration date. They used two different methods for final accuracy: Mode and Max-Prob. For accuracy, Vgg+both got the highest accuracy of 93.3% in Max-Prob. The lowest accuracy was 89.7 for Vgg16. So the author was successful by taking two more features for classifications. But the problem in this paper was they used a clear white background image with sufficient light which isn't related to real-life scenarios. Data is imbalanced and variability is problematic here. Implementing L1/L2, dropout, and regularization can help prevent overfitting.

In this paper, researchers Yang et al.[20] have basically focused on Convolutional

Neural Networks (CNNs) for the identification of fish classification. In addition, the researchers chose this method as 29 papers have already been reviewed ( CNNs) out of 41 which shows a greater percentage of 71% thus the researchers used the CNNs model. Moreover, other models that have been used other than CNNs are Generative Adversarial Networks (GAN), Recurrent Neural Networks (RNN), You Only Look Once ( YOLO), Deep Belief Networks (DBN), Long Short-Term Memory (LSTM) networks. Although some of the CNNs models are combined with SVM and Softmax or even with Softmax classifiers. Moreover, the accuracy of the models was increased by 15% and 10% by the models SVM and Softmax which was reported by Qin et al. [20]. Therefore, the average time taken for DL models for capturing a lionfish is 0.097 seconds per frame. Moreover, another study showed that the accuracy for capturing jellyfish is 80% by using the model of CNN with UAV systems. On the other hand, another model took 115 images within 6 seconds as per Meng et al.[20]. The quality and process of obtaining a large number of samples is a big challenge for this model reported by Ahmed et al. [20] and others. However, the CNNs model uses high-level features that are built on low-level ones where the initial layers learn low-level features while the latter layer learns the high-level features.

The authors Francis Jesmar Perez and Montalbo and Alexander A. Hernandez [36] describe the detailed process of Deep Learning (DL) in fish classification. Moreover, in Verde Island, the model VGG16 Deep Convolutional Neural Network (DCNN) is used for classifying the fish. The model VGG16 is improved by fine-tuning, optimized, and retained in order to increase the accuracy of the model in Verde Island while carrying out the classification of fish. In Addition, the accuracy of the VGG16 model is 99% while classifying three different fish species. This percentage overall describes the effectiveness of the model VGG16. In addition, the problem faced by the model while carrying out the process is the limitation of data and training only a few data sets. On the other hand, Montalbo and Hernandez utilized traditional transformation techniques of image transformation and color modification as part of their data augmentation process to increase the number of training images. The dataset used by Montalbo and Hernandez [36] consisted of images of three fish species.it is indicated that the model underwent training for ten epochs. The VGG16 model, as developed by K. Simonyan and A. Zisserman [36], is composed of layers with an input layer of [224 x 224 RGB] and having only a [3 x 3] convolutional layer throughout the whole model. The depth of the model was considered a significant factor in improving model performance.

The researchers Musulin et al [14] came up with a proposal for using the Convolutional Neural Network (CNN) for the classification of different types of fish. In addition, an image of 12859 in total. In addition, the model achieved an accuracy of 99.40% and this accuracy was obtained when the Identity function was applied to the hidden layers and RMSprop was used as a solver with a rate of 0.001 and a decay rate of 1e-5. Moreover, to balance the researchers Musulin et al [14] carried out only four clusters due to an imbalance of data sets. However, the paper mentions that all input images were rescaled into pixels of 100 x 100 before being applied to the first convolutional layer, which is a standard preprocessing step in CNNs to normalize the input size of the images. The CNN architecture described in the paper includes three convolutional layers with 32 and 64 feature maps respectively, and three pooling layers, two maximal pooling layers, and one average pooling layer which is followed by a fully-connected output layer. This setup indicates a multi-layered structure typical of CNNs, designed to extract and learn features at various levels of abstraction. Thus illustrating the effectiveness of CNNs in classifying fish species from images and showcasing the potential of machine learning in the field of marine biology and conservation.

In this paper, the researchers B. S. Rekha, G. N. Srinivasan, Sravan Kumar Reddy, Divyanshu Kakwani, and Niraj Bhattad [13] discussed the technique of detecting fish and their classification easily increased to an accuracy of (90-92)%. The researchers Rekha, Srinivasan, Reddy, Kakwani, and Bhattad [13], employed CNNs for both the detection and classification phases in the system. In Addition, the researchers used different architectures of CNNs to extract and analyze features from images captured through boat cameras. Specifically, they used the VGG-16 network for the classification phase. Moreover, due to an imbalance of datasets, the researchers faced problems while carrying out the plan. On the other hand, the quality and quantity of the picture was improved by data augmentation. In addition, Simple affine transforms and preprocessing techniques were applied to the initial dataset, which contained 3777 images. The augmentation process resulted in a total of 16000 images, creating a balanced dataset across all categories. Even Though the detection and classification networks were trained using the Adam optimizer, with a learning rate of 1e-3, the objective of categorical cross-entropy, and for 40 epochs and the classification network included several convolutional layers followed by fully connected (FC) layers.

The paper written by researcher Guangxu Wang, Akhter Muhammad, Chang Liu, Ling Du, and Daoliang Li [19], presents a novel approach to recognizing fish behavior using Deep Learning (DL). In addition, the DSC3D model achieved an accuracy of 95.79% for distinguishing the five species of fish. Moreover, the researchers also developed a dual-stream 3D convolutional neural network (DSC3D) to recognize the five fish classifications. The network combines spatiotemporal features and motion features using FlowNet2 and 3D convolutional neural networks. Moreover, the researchers employed methods such as grayscale geometric transformation and image enhancement. The frame rate of all input videos was adjusted to 16 frames, and the image size was adjusted to pixels 112 x 112. Moreover, the original C3D network structure used in this study consists of convolutional layers of eight, pooling layers of five, two fully connected layers, and finally one softmax layer. The convolution layers use a kernel of $3 \times 3 \times 3$ with convolution kernels number in each layer being 64, 128, 256, and 512, respectively.

Majumder, Rajbongshi, Rahman, and Biswas [16] explore various methods for fish recognition, noting that traditional approaches often struggle with the complexity of fish shapes and colors in their research paper. Alsmadi et al. [16] used neural networks to recognize fish with an accuracy of 86% while Storbeck and Daan [16] achieved 98% accuracy using computer vision, though the dataset details in their paper were lacking. Montalbo and Hernandez[16] used VGG16, and DCNN, achieving 98.67% accuracy with synthetic data. Rahman et al. [16] also focused on local bird recognition in Bangladesh, achieving high accuracy with MobileNet and Inception-V3. Deep and Dash [16] combined CNN with SVM and k-NN for 98.79% accuracy using standard datasets. Qin et al. [16] employed PCA and SPP in a deep-learning architecture for live underwater fish recognition, achieving 98.64% accuracy. Alsmadi et al. [16] combined ANN and Decision Tree for 96.4% accuracy. Hridayami et al. [16] used VGG16 DCNN for fish recognition across different datasets on the other hand Khalifa et al. [16] applied DCNN for aquarium fish species identification. Sharmin et al.[16] used machine vision with SVM, k-NN, and Ensemble classifiers for local fish recognition, achieving 94.2% accuracy. In conclusion, the diverse body of work highlights the evolution and effectiveness of deep learning and neural network approaches in fish classification.

The research paper of Mathur, Vasudev, Sahoo, Jain, and Goel [12] used various deep learning approaches for fish species classification thus highlighting the inefficiencies of traditional methods due to their time-consuming, and expensive nature. Alsmadi et al. [12] achieved 86% accuracy using neural networks for fish recognition by combining size and shape features. On the other hand, Storbeck and Daan [12] obtained 98% accuracy using computer vision but lacked in obtaining the dataset for more details. Moreover, Montalbo and Hernandez [12] reached 98.67% accuracy with VGG16 DCNN and synthetic data whereas Rahman et al. (2020) recognized local birds in Bangladesh with high accuracy using MobileNet and Inception-V3. Deep and Dash [12] combined CNN with SVM and k-NN for 98.79% accuracy on standard datasets.The last but not the least, Qin et al. [12]employed PCA and SPP in deep learning for live underwater fish recognition, achieving 98.64% accuracy. Alsmadi et al. [12] combined ANN and Decision Tree with 96.4% accuracy. Hridayami et al.[12] used VGG16 DCNN across various datasets for fish recognition. Khalifa et al. [12] applied DCNN for aquarium fish species identification. Sharmin et al.[12] used machine vision with SVM, k-NN, and Ensemble classifiers for local fish recognition, achieving 94.2% accuracy. In conclusion, the work demonstrates the effectiveness of deep learning and neural networks in improving fish species classification, particularly in challenging underwater conditions.

In the research paper, Aziz, Desai, and Baluch [30] proposed a novel Deep Learning Artificial Neural Network model that is optimized for fish image classification and addressing the inefficiencies of traditional methods that are tedious and slow. The study highlights the importance of identifying and classifying fish species due to their nutritional value and the need to recognize symptoms of sickness and decay. The proposed DLANN model incorporates a Genetic Algorithm into the exploration phase of the Cuckoo Search algorithm to enhance optimization. Moreover, the extensive experiments comparing this GA-CS optimized DLANN with popular deep learning techniques such as EfficientNet, Inception V3, ResNet150 V2, VGG-19, DenseNet 121, LSTM Model and a personalized Convolutional Neural Network model demonstrate superior performance in classification accuracy, recall, precision, standard deviation and F1 scores making it an efficient tool for fish image classification.

Ma, Y., Zhang, P., Tang, Y. [7]. In addition, the application of transfer learning in fish image classification using convolutional neural networks. Moreover, the traditional process of fish image classification is complex and lacks accuracy. However, convolutional neural networks can improve accuracy but the process requires large datasets, which presents the main challenges in this research paper. The study proposes using transfer learning with a CNN model to classify fish images when sample sizes are small. Initially, image data is normalized and enhanced for better classification outcomes. The ResNet50 network's final layer is modified into a six-category fully connected layer, utilizing pre-trained weights from the ImageNet dataset. Moreover, the experiment examines the impact of frozen layers in ResNet50 and the number of training samples on classification accuracy and overfitting. Results show that the ResNet50 model, with transfer learning, has strong feature extraction and generalization capabilities and resists overfitting, achieving up to 97.19% accuracy in fish image classification and demonstrating a significant time advantage.

Alaba, Nabi, Wallace, Ball, Shah, and Moorhead [21] address the importance of accurate fish species recognition for ecosystem monitoring, production management, and identifying endangered species in this research paper. In addition, the scientists put out an object detection model to address the problem of categorizing several fish in one picture. The model combines an SSD-detecting head with the MobileNetv3-large and VGG16 backbone networks. They add a class-aware loss function that gives species with fewer instances more weight in order to address the dataset's class imbalance. Consequently, this method improves the model's performance on a variety of datasets. To sum up, the SEAMAPD21 reef fish dataset experimental findings show that the class-aware loss function can increase model accuracy up to 79.7% over the original loss. Additionally, the model outperforms the original SSD object detection model according to tests conducted on the Pascal VOC dataset.

Shammi, Das, Hasan, and Noori [18] explored the application of both traditional machine learning algorithms and Convolutional Neural Networks for fish classification to assist individuals in identifying fish species and understanding their nutritional value. Moreover, the study mainly aimed to address the common knowledge gaps regarding fish classification, which can help individuals to make informed dietary choices and provide crucial support to those involved in the marine fishing industry. In addition, the authors experimented with six types of fish and applied traditional algorithms such as Support Vector Machine (SVM) alongside CNN to evaluate their effectiveness in classification tasks. Although, among the two models, SVM demonstrated the highest accuracy of 63.93%, but the results highlighted limitations in handling complex image data. In contrast, the CNN model achieved a significantly higher accuracy of 88.96%, showcasing its superior capability for feature extraction and classification in visual datasets. Even Though, the study emphasizes CNN's suitability for fish classification due to its ability to learn intricate patterns and deliver a strong performance compared to traditional methods. Shammi et al. [18] underscored the practical applications of their work in promoting nutritional awareness and supporting sustainable fishing practices. In conclusion their findings contribute to the growing body of research validating CNN as a reliable approach for image based fish classification while providing insights into the relative strengths of machine learning algorithms.

Dey, Hassan, Rana and Hena [15] proposed a CNN based system to classify traditional indigenous fish species in Bangladesh addressing a significant cultural and nutritional need. In addition,fish is a vital component of Bangladeshi cuisine and the primary protein source for many rural households. However, numerous indigenous fish species are either extinct or endangered and the younger generation struggles to recognize them as well. Moreover, the authors emphasized that an automated fish classification system could play a crucial role in identifying, preserving and promoting these fish species will be benefiting both conservation efforts and the fish industry, which supports thousands of livelihoods in the country. Although,their study utilized a dataset of eight classes of indigenous fish, comprising 8,000 images created through eight data augmentation techniques, such as flipping, rotating and scaling, to enhance the training process. However,the authors evaluated the dataset using pre trained models like VGG16, Inception V3 and MobileNet thus modifying the output layers for better adaptation. Furthermore, Dey et al. [15] introduced a custom 5 layer CNN model named FishNet, which incorporated "Adam" optimization, ReLU and SoftMax activation functions. Moreover, their results demonstrated that FishNet outperformed VGG16 across various performance metrics and delivered competitive results against Inception V3 and MobileNet. In conclusion, the study also highlighted the importance of combining pre trained models with custom architectures to achieve optimal accuracy and also underscored the potential of AI in preserving in Bangladesh's aquatic heritage.

Tamou, Benzinou and Nasreddine [22] addressed the challenges of classifying live reef fish species in dynamic underwater environments and also focusing on the biodiversity of reef ecosystems. They emphasized that underwater video systems are non invasive and generate significant visual data for marine ecology but the analysis of such data is hindered by poor image quality and challenging aquatic conditions. In addition,to tackle these issues the researchers proposed a deep Convolutional Neural Network combined with an incremental learning strategy. Moreover,this novel approach began by training the network to focus on classifying the most challenging fish species before incrementally introducing new species thus utilizing knowledge distillation loss in order to maintain high accuracy on previously learned species. Tamou et al. [22] methodology demonstrated the effectiveness of progressive learning and achieved an accuracy of 81.83% on the LifeClef 2015 Fish benchmark dataset. Therefore, this study also highlighted the potential of CNN model to adaptively learn and classify diverse species in unconstrained environments, offering a pathway for efficient and scalable biodiversity monitoring. In conclusion, the authors contributed significantly to advancing the field of automatic fish species classification and bridging gaps between ecological conservation needs and technological capabilities.

Ismail, Ayob, Muslim and Zulkifli [17] explored fish image classification by comparing the performance of several convolutional neural network (CNN) models, emphasizing their potential to solve complex image classification problems with high accuracy. Their study employed AlexNet, GoogLeNet, and ResNet-50, leveraging transfer learning with data augmentation techniques to enhance model performance. The dataset, comprising 18,000 fish images across 18 categories, was divided into 12,600 images for training (70%) and 5,400 images for validation (30%). The results highlighted the effectiveness of these CNN models, with AlexNet achieving the highest classification accuracy of 99.85%, followed by ResNet-50 at 99.51%, and GoogLeNet at 96.39%. To assess their frameworks, the authors utilized various evaluation metrics, including learning curves, validation tests, confusion matrices, and top-five predictions. Despite the visually similar features among certain fish species, these models demonstrated exceptional precision, showcasing their robustness in identifying even subtle differences. The research underscores the potential of CNN architectures in developing state-of-the-art tools for fish classification, contributing significantly to the fields of aquaculture and ichthyology. This study serves as a benchmark for leveraging deep learning in fish species recognition, addressing the challenges of visually indistinct datasets with innovative neural network applications.

Deka, Laskar and Baklial [23] proposed an innovative framework for classifying indigenous freshwater fish species using deep learning and traditional machine learning techniques. Moreover, the study shows a utilized pretrained ResNet-50 architecture, specifically leveraging the top three layers to extract deep features from fish images. In addition,these features were then used as inputs to a "one-for-all" Support Vector Machine (SVM) classifier and enabling precise classification of eleven indigenous fish species from India. Deka et al. [23] evaluated the framework on their custom dataset and the Fish-Pak dataset thus achieving remarkable results this classification framework yielded an unprecedented accuracy, precision and recall of 100% on their custom dataset, while on the Fish-Pak dataset, it achieved a maximum accuracy of 98.74%, setting a new benchmark in fish classification tasks. Although, this hybrid approach highlights the strength of combining feature extraction from CNNs with traditional classifiers to overcome dataset variability and enhance generalization. In addition, the research underscores the potential of deep learning in ecological studies, providing a strong solution for fish species identification that integrates state of the art feature extraction and scalable classification. In conclusion, the study's exceptional accuracy across datasets reinforces its applicability in biodiversity monitoring and conservation efforts.

Zhuang, Xing, Liu, Guo, and Qiao [4] contributed significantly to the SEACLEF 2017 task by developing advanced deep learning models for marine animal detection and recognition. Their research focused on three subtasks, including automatic fish identification, frame-level salmon detection, and marine animal recognition. Moreover,for the fish identification and species recognition task they utilized frameworks to detect proposal boxes around the fish in images and fine tuned a pre-trained neural network for classification. In the salmon identification task, the team implemented the BN-Inception model to determine the presence of salmon in video frames. Additionally, for marine animal recognition, they examined various neural networks to classify images with weak labels, showcasing the potential of deep learning in scenarios with limited annotations. However, Zhuang et al. [4] methods achieved strong results, particularly in the fish identification and marine animal recognition subtasks, demonstrating the effectiveness of leveraging pretrained networks and task specific tuning. In conclusion, this research highlights the adaptability of deep learning architectures to diverse tasks and datasets, underscoring their potential in tackling real world challenges in species classification y addressing these complexities, the study contributes to advancing biodiversity monitoring and conservation through precise and automated techniques.

Auliasari et al. [29] researched a study that explored the use of deep learning models, specifically a general deep learning model and a specialized VGG16-based architecture, for fish classification tasks. Moreover, their work demonstrated the general model's promising capability to improve both accuracy and validation accuracy over successive training epochs. However, they noted that higher training accuracy compared to validation accuracy might indicate potential overfitting, which could hinder the model's generalizability and to address this phase they emphasized the importance of monitoring validation accuracy and implementing early stopping mechanisms when training accuracy level. The VGG16 model, on the other hand, showcased exceptional performance, achieving precision, recall and F1-scores exceeding 0.97, 0.96 and 0.98, respectively, across all fish species. Therefore, these metrics highlight the model's strong ability to differentiate among diverse species effectively. In addition, the study also recognized the potential applications of these models in fisheries management, aquaculture and ecological research, emphasizing their role in promoting sustainable fishing practices and accurate fish population monitoring. For future improvements, the researchers proposed expanding datasets, incorporating other emerging technologies and exploring explainable AI methods to enhance transparency and performance. In conclusion, their findings contribute to a growing body of evidence supporting the practical utility of deep learning in ecological and marine studies.

Deka, Laskar, and Baklial [31] explored the classification of freshwater fish species using deep learning techniques, addressing the challenges posed by the resemblance among species and the limitations of manual identification. Moreover, the authors utilized AlexNet and ResNet-50, two widely recognized deep learning architectures in order to classify 20 indigenous freshwater fish species from North Eastern India and by fine tuning these models, they evaluated their performance on both a custom dataset and the Fish Pak benchmark dataset. However,their analysis revealed that the ResNet-50 model achieved a remarkable 100% classification accuracy, precision and recall at an optimal learning rate of 0.001, outperforming AlexNet. The study highlighted the role of hyperparameter tuning and demonstrated that increasing the Weight and Bias learning rate led to higher validation losses. Moreover, this empirical insight emphasizes the need for careful optimization in deep learning tasks and by achieving exceptional accuracy, Deka et al. [31] also showcased the potential of pre-trained models in biodiversity conservation and ecological studies. Also their work underscores the importance of advanced computational approaches in addressing real-world challenges, such as distinguishing closely related species and establishes a strong framework for automated freshwater fish classification. In conclusion, the research offers significant contributions to the field, providing insights into scalable and accurate species identification.

Mittal et al. [27] conducted a comprehensive survey on the application of deep learning techniques for underwater image classification and emphasizing its growing significance in diverse domains. Moreover, the authors explored the classification of underwater objects, including fish, plankton, coral reefs, and even sea diver gestures, underscoring its importance in assessing aquatic health and protecting endangered species. However, their review highlighted the potential of deep learning to address challenges in oceanography, marine economy, environmental protection and underwater exploration by further extending its utility to human robot collaborative tasks and defense applications and by analyzing various deep learning methods, the study also illuminated both the similarities and distinctions among approaches, providing researchers with valuable insights of techniques. Mittal et al. [27] also emphasized the critical role of underwater image classification as a definitive application for evaluating the success of deep learning methods in real world contexts.

In addition, their work acts as both a guide for researchers and a call to advance the field by addressing existing challenges. In conclusion, this survey not only underscores the transformative potential of deep learning in underwater ecosystems but also highlights the necessity for innovation to push the frontiers of aquatic image classification forward, paving the way for improved ecological preservation and technological integration.

Desai [24] explored the pressing issue of overfishing and its impact on marine ecosystems therefore proposing an innovative deep learning based method to address the challenge. Moreover, the study focuses on classifying nine distinct fish species using a dataset of 430 images, leveraging two popular models, MobileNetV2 and VGG16. MobileNetV2 emerged as the better performing model by achieving a test loss of 0.12303 and an accuracy of 96.51%, while VGG16 recorded a test loss of 0.39825 and an accuracy of 88.37%.On the other hand,to enhance model performance the data augmentation and fine tuning strategies were applied by demonstrating the potential for achieving high accuracy even with a limited dataset. Desai [24] also highlighted the importance of preserving natural habitats by enabling the identification and selective retention of fish, thus reducing the bycatch of unwanted species. However, this study also emphasizes the challenges posed by the scarcity of high quality data sources and demonstrates MobileNetV2's capability to maintain reliable accuracy under such constraints and by integrating deep learning techniques into fisheries management the research contributes to sustainable fishing practices by aiming to balance ecological conservation with industry demands. In conclusion,this work exemplifies how advanced computational approaches can be applied to address real-world environmental challenges effectively.

Chuang, Hwang, and Williams [2] addressed the challenges of live fish recognition in fisheries surveys by emphasizing issues like poor underwater image quality, uncontrolled environments and the difficulty in acquiring representative samples. Moreover,to overcome these obstacles, they proposed a fully unsupervised feature learning technique paired with an error resilient classifier by forming a strong framework for underwater fish recognition. In addition,the approach begins with object part initialization by using saliency detection and relaxation labeling thus ensuring accurate part matching even in uncertain conditions. However, a non rigid part model is then developed, guided by fitness, separation and discrimination criteria by allowing the system to handle variable and complex fish morphologies effectively. Even Though,the framework introduces a binary class hierarchy for classification, where each node acts as an independent classifier and developed using an unsupervised clustering method. On the other hand, to deal with ambiguous images, Chuang et al. [2] introduced the concept of partial classification, which assigns coarse labels by optimizing the classifier's "benefit" of indecision. The system demonstrated high accuracy on both public and self collected datasets even when faced with class imbalance and uncertainty. In conclusion,this study highlights the potential of unsupervised learning and strong classification methods in automating fisheries surveys by providing a scalable solution for identifying fish species in challenging underwater environments.

Burger et al. [1] proposed an innovative fish classification system which aimed at assisting marine biologists in analyzing fish behavior in natural underwater environments. Moreover, the system integrates two types of features texture and shape to accurately classify fish species. In addition, texture features are derived through statistical moments of the gray level histogram, spatial Gabor filtering and properties of the co-occurrence matrix. Meanwhile, shape features are extracted using the Curvature Scale Space transform and the histogram of Fourier descriptors of the fish's boundaries. On the other hand, to further enhance the accuracy of feature extraction, Burger et al. [1] employed an affine transformation to represent the fish in 3D from multiple views. Although, the system was tested on a dataset of 360 images from ten different species and achieving an impressive classification accuracy rate of 92%. In addition to classification, the authors also integrated a tracking system to analyze fish trajectories, providing valuable insights into unusual fish behaviors. In conclusion, the combination of classification and tracking allows for the detection of anomalies, which can be used for further investigation by marine biologists.

Cheng et al. [35] presented an innovative approach to underwater target recognition by leveraging an improved YOLOv11 algorithm which is implemented on the Jetson Xavier NX platform, their system addresses common challenges such as low accuracy, high energy consumption and limited automation in existing underwater recognition methods. In addition, employing an industrial camera to capture images and integrating hardware software optimization, the proposed system enhances clarity through adaptive lighting adjustments. Moreover, the improved YOLOv11 model, termed DD-YOLOv11, outperforms its predecessor by achieving an accuracy of 87.5% on the ROUD dataset thus demonstrating a 5% reduction in parameter size and a 0.3G decrease in floating point operations. Notably, the system's lightweight design ensures compatibility with resource constrained environments while maintaining high precision, making it suitable for real time applications. Additionally, experiments on the URPC2020 dataset validate the model's versatility across different conditions. In conclusion, the integration of a buoyant ball with positioning capabilities further illustrates the system's automation and communication potential and ensures seamless operation even in dynamic underwater settings. By offering a highly automated and energy efficient solution, this study underscores the significant benefits of YOLOv11 in underwater segmentation, including its capability to improve accuracy by optimizing resource usage and also adapt to diverse aquatic environments.

He, Cao, Xu, Luo, Chen, Song and Huo [34] addressed the challenges in measuring aquatic animals' morphology such as scale variability and feature extraction complexities, by proposing an innovative Point Cloud Measurement Model combined with an improved YOLOv8 instance segmentation algorithm. Moreover, their method integrates a depth camera to reconstruct 3D point clouds and process aquatic animal images by offering a precise solution for capturing length, width and height measurements with remarkable accuracy. In addition, the improved YOLOv8 model demonstrated significant advancements, achieving a mAP@0.5 increases of 97.5% on the custom AASM dataset, 3.5% on the UIIS dataset and 1.5% on the USIS10K dataset compared to baseline models. Furthermore, the absolute error for dimensions measured, such as the clam's height was reduced to less than 1 mm thus showcasing the precision of their approach. In conclusion, He et al. [34] highlighted the versatility of YOLOv8 in underwater segmentation tasks, benefiting from its ability to handle diverse aquatic datasets effectively and these improvements made

YOLOv8 highly efficient tools for underwater segmentation and offering unparalleled accuracy, adaptability and also computational efficiency in dynamic aquatic environments.

Macías et al. [33] explored the use of deep learning models for instance segmentation of sea lions in video frames captured in the Galápagos Islands. Moreover,their study highlighted the challenges of segmenting aquatic species, including the proximity of individuals during group activities, variable lighting conditions and inconsistent image quality. In addition,the research compared several models of YOLOv8 and YOLOv9 outperformed others, such as Detectron2 with a Mask R-CNN backbone. On the other hand, YOLOv8 achieved a remarkable AP50 rate of 97.90% while YOLOv9 surpassed it with a precision rate of 98.2% demonstrating exceptional accuracy in identifying and delineating sea lions. However, the authors emphasize the practicality of YOLOv9 in underwater segmentation, as it excels in handling complex scenarios thus including overlapping objects and dynamic environmental changes also YOLOv9's enhanced precision and strength to make it an invaluable tool for wildlife monitoring in diverse natural habitats. In conclusion, enabling accurate tracking and segmentation it supports conservation efforts by offering insights into behavioral patterns and population dynamics thus the significant potential of advanced YOLO models to address critical ecological challenges, facilitating more effective biodiversity monitoring and management in underwater environments.

Garcia et al. [25] tackled the pressing issue of marine biodiversity loss and its impact on fisheries by developing an automated monitoring system for fish markets. Moreover, recognizing the fishing sector's struggle with outdated practices and unreliable data on fish catches, the authors also utilized YOLACT models to identify and classify fish species directly from in-tray images. In addition, their approach also influences the newly introduced DeepFish dataset, enabling accurate species recognition in real-time scenarios. However, this innovative system not only provides detailed insights into fish populations but also supports stakeholders in assessing the overall health of fisheries and managing species sustainability and by experimenting with various network backbones, the study demonstrated YOLACT's capability to achieve reliable performance in complex environments like fish markets. In addition,the use of YOLO-based segmentation models, including YOLACT, offers significant benefits in underwater and market segmentation tasks. On the other hand, these models are highly efficient, capable of handling overlapping instances and adaptable to diverse datasets, making them particularly effective for recognizing multiple species under dynamic conditions.In conclusion, the integration of YOLO technology in such systems bridges the gap between traditional fishing practices and modern conservation efforts, fostering sustainable fisheries management through precise and automated data collection.

In conclusion, the reviewed research papers showcase the diversity and progression of methodologies in fish species classification using machine learning. Advanced architectures like AlexNet, VGGNet, ResNet, and their modifications (e.g., CNN-SENet and VoNet8) have been effectively utilized. Preprocessing techniques such as erosion and Otsu's threshold further demonstrate the range of methods applied to improve accuracy. Studies encompass small and large underwater datasets, enabling comprehensive accuracy comparisons.The literature identifies challenges like underwater illumination, varying watercolors, and the complexity of capturing clear images due to natural underwater structures. Researchers have adopted innovative solutions, such as unsupervised learning, synthetic data generation, and incremental learning strategies, to address these issues. Techniques like transfer learning, hybrid CNN-SVM approaches, and custom architectures (e.g., YOLO models and FishNet) have achieved remarkable results in fish classification and segmentation tasks, even under constrained conditions.The research has bridged gaps between lab environments and real-world applications by addressing data diversity and environmental complexities. For example, leveraging YOLO-based segmentation models demonstrated the capability to handle overlapping fish instances, whereas ResNet-50-based classifiers achieved unparalleled precision in diverse datasets. Additionally, models like MobileNet and InceptionResNetV2 have proven their adaptability and efficiency for real-time applications.Finally, the studies highlight the importance of sustainable fisheries management and biodiversity conservation, emphasizing automated fish behavior recognition, real-time monitoring, and the use of AI for ecological preservation. These advancements inspire future research directions and provide a robust foundation for extending the practical applications of machine learning in aquatic environments.However, these selected papers address the gaps between laboratory research and real-world applicability, which is a key consideration for our project. By examining how different models perform under real-life conditions, we can better understand the practical challenges and limitations we might face and develop strategies to address them. In summary, the reason why these papers were chosen is because they provide a rich and varied perspective on the challenges and opportunities in the field of fish species classification using deep learning as their insights and findings have significantly shaped our approach and methodology, guiding us towards a more comprehensive and informed research strategy.

# Chapter 3

# Methodology

## 3.1 Work Plan

Firstly, we have built a custom dataset consisting of 10 classes of fishes and a combination of fishes in sets of 2, 4, 6 and 8 as well as overlapping fishes ensuring good-quality images. Our total dataset consists of 7100 images. Acquiring a better dataset played a vital role in achieving accuracy. For better accuracy, we used image augmentation like rotation, flip, zooming, shearing etc. The dataset has been split into training and testing. We used 80% for training and 20% for testing. Moreover, for every class, 400 images were selected for training and 100 for testing, totaling up to 4000 for training and 1000 for testing in case of classification. For segmentation, the combination of fishes in sets were combined into 2000 images and then split into 1600 images for training and 400 images for testing. For overlapped segmentation, the 100 images were split into 80 images for training and 10 images for testing. Different models were employed for comparison to identify the best one in terms of accuracy. We used a basic CNN architecture, then applied the VGG-16 architecture along with ResNet-50 and Inception-v3 with the help of our literature and collected knowledge about models in terms of accuracy and efficiency for classification. The models used for segmentation were YOLOv8 followed by YOLOv9 and then YOLOv11 for both overlapping and non-overlapping images. The fundamental step in deep learning is model training, which teaches a neural network how to infer or decide from provided data. In model architecture, loss function, optimized algorithm, backpropagation, epochs, and batch size are some key elements in model training. Here we used Relu and Softmax as an activation function in classification. Basic CNN is a multi-class classification so we encoded the classes from 0 to 9. In case of segmentation, we annotated the 2100 images manually using the Roboflow annotation tool. For performance evaluation, we used multiple models to understand which model is working best in both classification and segmentation.

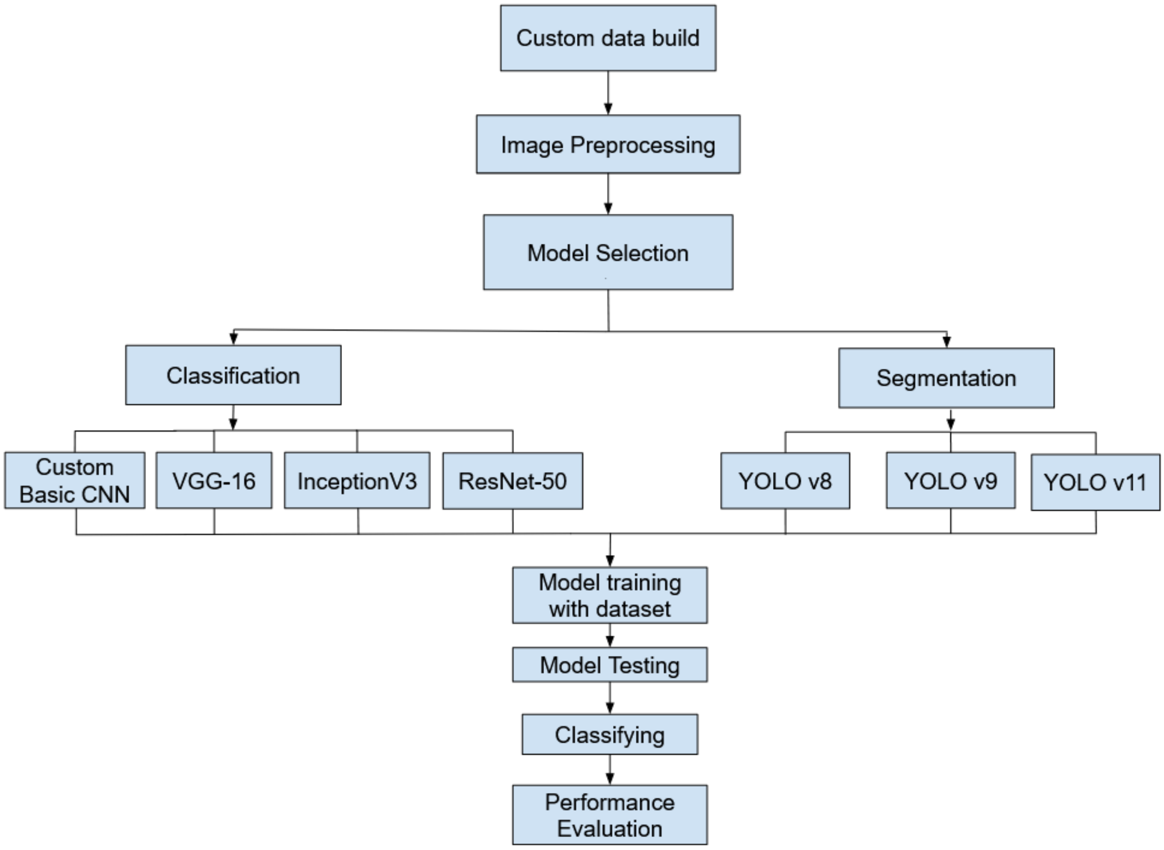


Figure 3.1: Work Plan

## 3.2 Custom Dataset

### 3.2.1 Dataset Introduction

The dataset for this thesis comprises 7,100 custom-captured fish images, all taken using an iPhone and a camera (Canon RP) to ensure consistent quality and high resolution. There are a total of 10 classes of fish each containing 500 images totaling up to 5000 images for classification. Other than that, our dataset also consists of 2100 more images from which 2000 images consist of different combinations of non-overlapping segmentation fishes and 100 images consist of overlapping of fishes for segmentation. Moreover, a critical feature of this dataset is the use of a plain white background in every image which simplifies the detection, classification and segmentation by minimizing distractions and focusing the deep learning models solely on the fish. In addition, this choice allows the model to identify key features such as shape, texture, and fin patterns with greater precision, enhancing accuracy in classification and segmentation tasks. The uniform background also contributes to a more controlled dataset, reducing variability that could confuse the model during training and improving its generalization across different fish species by eliminating unnecessary complexity in the image space, this dataset optimizes conditions for the deep learning models to perform better, making it a valuable asset for achieving higher accuracy in fish classification in the context of this research.

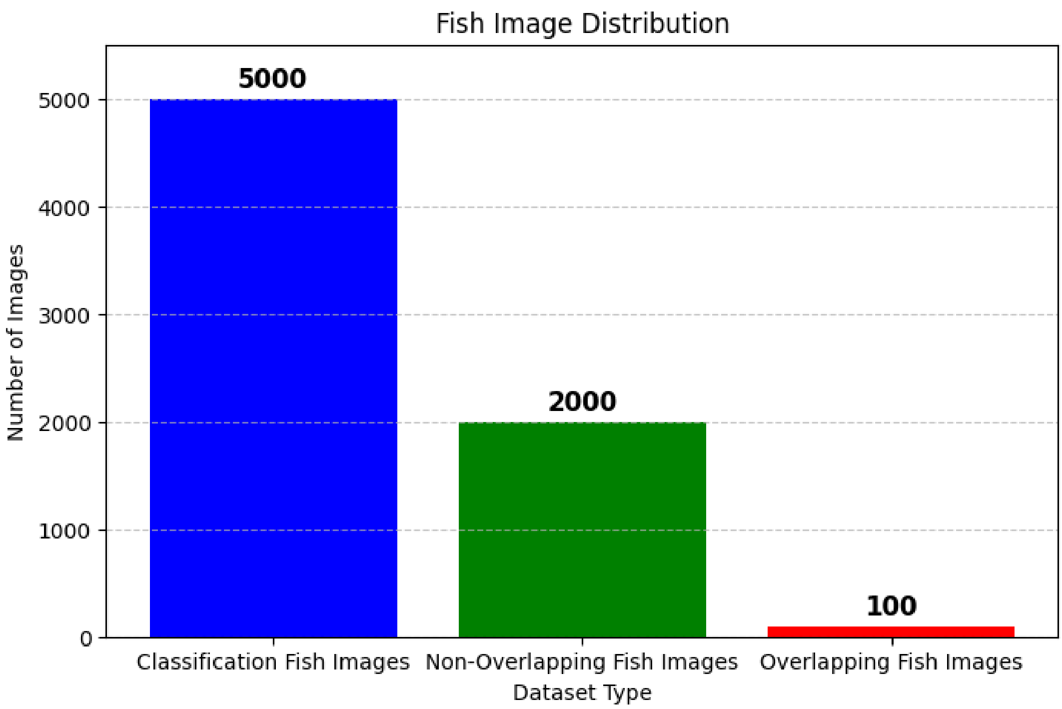


Figure 3.2: Dataset Splitting

### 3.2.2 Classification Dataset Construction

In this dataset, the fish samples were captured from various angles and sizes to enhance the model's ability to generalize across different perspectives and angles. Moreover, images were taken at varying angles, providing a diverse range of viewpoints for each fish and this approach ensures that the deep learning models used in this research can effectively handle rotational variations, improving the quality in real-world applications where fish may not always appear from the same angle. Additionally, the images were captured with varying zoom levels, ensuring both close-up and distant shots of the fish. This variability in size allows the model to learn to detect and classify fish regardless of how prominently they appear in the frame. Importantly, despite these variations in angles and zooming the uniform white background was consistently maintained throughout the dataset to ensure model's focus on the fish. Furthermore, all the images were taken under natural daylight as well as nighttime conditions, which is crucial in maintaining the quality of the dataset. The use of daylight standardizes the lighting conditions, offering consistent illumination across the entire dataset, which is beneficial for training deep learning models to perform well in various real-world lighting scenarios. On the other hand, artificial lighting during night-time provides low lighting and shadows, creating stark contrasts between the bright parts of the fish and the surrounding background darkness. The combination of varied angles, zoom levels, and a consistent white background ensures that the dataset not only captures the fish in a detailed and diverse manner but also provides optimal conditions for the models to learn effectively. In conclusion, this approach to data collection enhances the potentiality of deep learning models to achieve high accuracy and robustness in fish classification, making the dataset an invaluable resource for this research.

a) Magur

b) Bain

c) Kachki

d) Hilsha

e) Aar

f) Shing

g) Shrimp

h) Rui

i) Pabda

j) Tengra

Figure 3.3: 10 Raw Image Of Single Fishes

The fish samples for this thesis were initially captured using an iPhone, where the images were stored in HEIC format, which is the default for iPhones to save space while maintaining high image quality. However, for better compatibility and ease of use in machine learning workflows, the HEIC files were later converted to JPEG format, which is more universally supported across various platforms and tools. This conversion ensured that the images retained their quality while becoming easier to integrate into the deep-learning model pipeline. After the conversion, the 5,000 JPEG images were uploaded to cloud storage, specifically on a drive, to facilitate easy access and efficient management of the dataset, thus storing the images in the cloud ensures that they are backed up and accessible from multiple devices, providing flexibility during the research process.

### 3.2.3 Segmentation Dataset Construction

#### 3.2.3.1 Non-Overlapping Dataset

In this dataset, fish samples were grouped and captured in sets of 2, 4, 6, and 8 fish per image, with each set containing 500 images. For example, in the set of 2, any two fish were randomly selected and combined for each of the 500 images, ensuring diverse pairings across the dataset. Although, the same process was applied to the sets of 4, 6, and 8 fish. The combination set of fish samples was captured at varying angles, to enhance the model's ability to recognize fish from different perspectives. A total of 2,000 combination images were captured, ensuring high-resolution quality. These images, like the rest of the dataset, were initially stored in HEIC format and later converted to JPEG for easier processing and compatibility with machine learning tools.

Figure 3.4: Combination of 2 Fishes

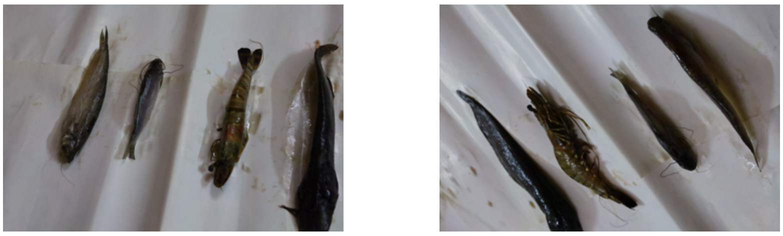

Figure 3.5: Combination of 4 Fishes

Figure 3.6: Combination of 6 Fishes

Figure 3.7: Combination of 8 Fishes

The main purpose of this segmentation dataset is to predict each fish individually from the sets of 2, 4, 6, and 8, fishes using segmentation models.

#### 3.2.3.2 Overlapping Dataset

In this dataset, the fish samples were grouped into 6 fishes. It consists of 100 images where the fishes are overlapped on top of each other and the images were captured following different patterns. For example, the fishes were put on top of each other and then after capturing 5 images from different angles, the position of the fishes was changed and then again another set of 5 images were captured in different angles. In this way a total of 100 images were captured. The reason for taking the images in such a way is to ensure diversity in the dataset by exposing it to different angled backgrounds so that the models may train properly and give appropriate accuracy.

Figure 3.8: Overlapping Fishes

The images were captured using a Canon RP camera. The format of the images were set to JPEG by default from the camera.

### 3.2.4 Dataset Splitting

#### 3.2.4.1 Classification Dataset Splitting

In supervised learning approaches, which are used for different types of classification and regression models, it is imperative to divide the dataset into training and testing portions to assess the performance of a machine learning model. For our dataset of 5,000 fish images, we split our dataset manually into 80:20 ratio, allocating 80% of the image from each class which sums up to 4,000 images for training, and reserving the rest 20% of the image from each class which sums up to 1,000 images for testing. For each class 400 images for training and 100 images for validation.

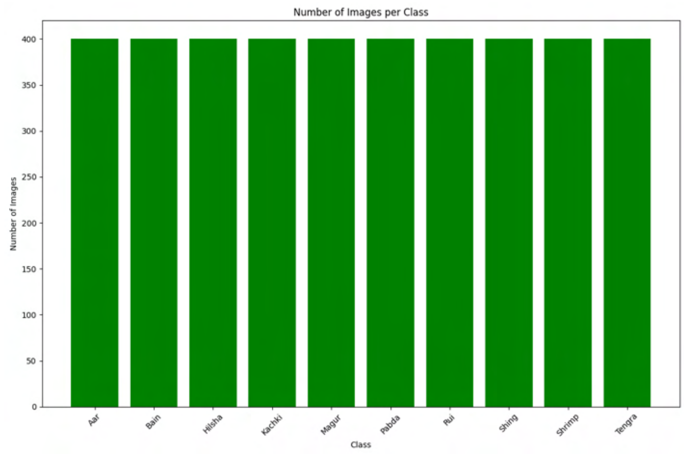


Figure 3.9: Graph of Training Dataset

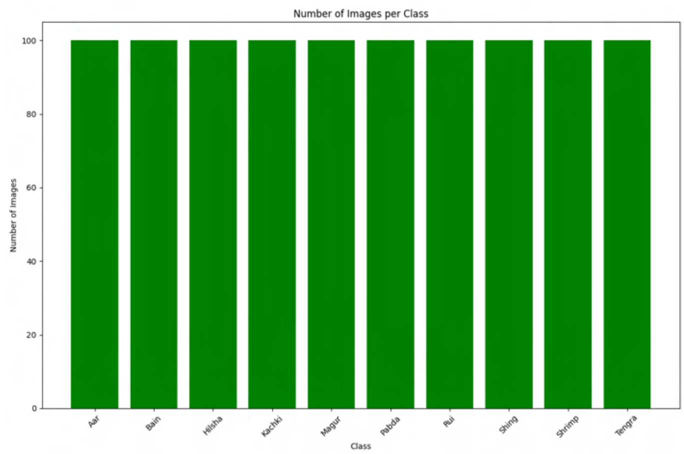


Figure 3.10: Graph of Testing Dataset

#### 3.2.4.2 Non-Overlapping Segmentation Dataset Splitting

The fish segmentation task consists of 1600 training images and 400 testing images, making a total of 2000 images. The splitting of the dataset was done manually the same way as classification dataset with 80:20 ratio. After annotation, it was observed that the number of instances for each fish category is not equal, resulting in an unbalanced dataset. The highest number of fish is shrimp and the lowest number of fish is Hilsha in this dataset of 2000 images.

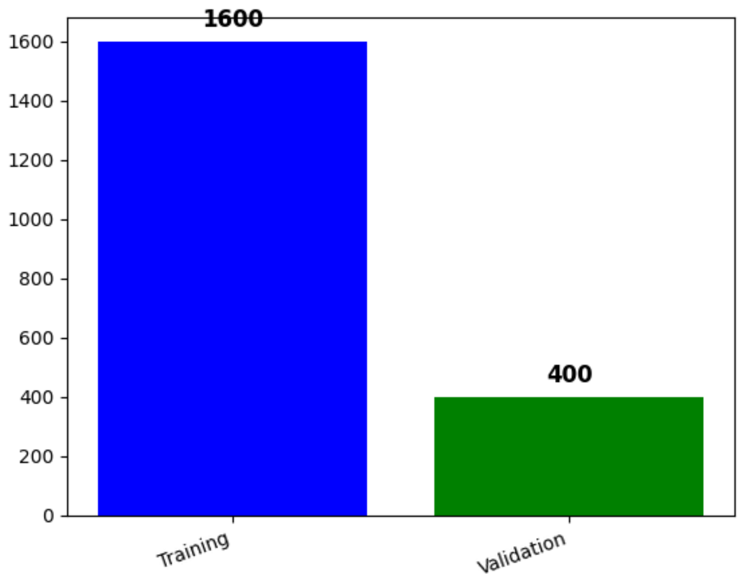


Figure 3.11: Non-Overlapping Dataset Splitting

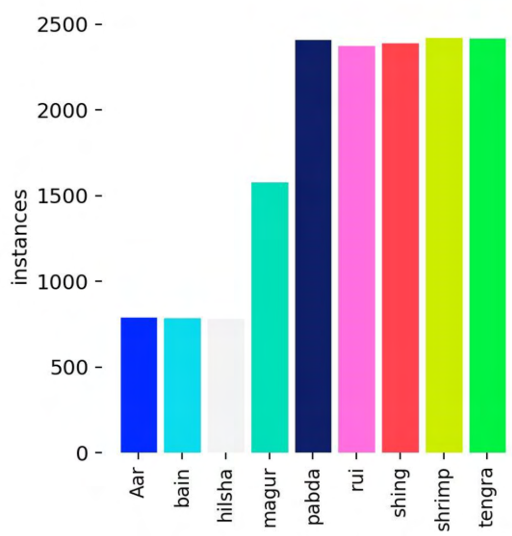


Figure 3.12: Class-Wise Instances

#### 3.2.4.3 Overlapping Segmentation Dataset Splitting

In this task of segmentation, the dataset consists of 100 images which was split into 20:80 ratio like the rest of the datasets. 80 images were kept for training and the rest of the 20 images were used for testing. The images were annotated before splitting.

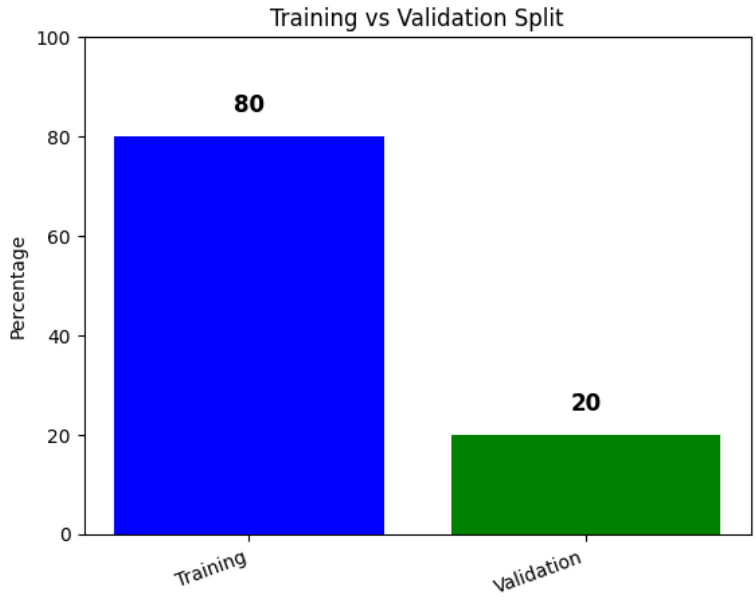


Figure 3.13: Dataset Splitting

### 3.2.5 Data Pre-Processing

#### 3.2.5.1 Classification Data Preprocessing

Data pre-processing is a critical step in preparing the fish dataset for deep learning model training. In this case, the 5,000 single fish samples were carefully modified to a resolution of 230x230 pixels to ensure consistency, compatibility, and speedup model training. Resizing the images is also an important step since it guarantees consistency throughout the dataset, which helps the model process images quickly and extract useful features without being influenced by different image sizes. In addition, the pre-processing of 5,000 images was first gathered from the dataset, which was captured using an iPhone and later converted from HEIC to JPEG format. Each image was then resized from its original dimensions to a fixed size of 230x230 pixels. On the other hand, low resolution could lead to a loss of important features necessary for accurate classification. Therefore, 230x230 pixels is an optimal size for this dataset. The resized images were then saved permanently in two dedicated folders which are training resized and testing resized. After resizing, the images were saved in class-specific folders. The dataset consists of 10 distinct classes of fish in the same folder as the train and test models the samples are resized with an even distribution of images across these classes and each image within these classes underwent the same resizing process to ensure that all samples were uniform in size and this folder acted as the input for our model and this step is important because deep learning models typically expect input images of the same dimensions, and having consistent image sizes ensures that the convolutional layers in the model can process the data without any errors and increases speed of the training process. Proper organization of these folders is critical for model training, as it allows the deep learning framework to correctly map each image to its respective label during the training phase. These resized images will now be used for tasks such as data augmentation, where further transformations like rotation, flipping, shearing, shifting and scaling can be applied. This pre-processing step ensures that the dataset is ready for training a high-quality and accurate fish classification model, with each of the 10 fish classes being represented uniformly in a standardized format.

For each of the four models we did pre-processing by resizing the original image as well as using augmentations like rotation, re-scaling, height shifting,width shifting, shearing, horizontal flips and zooming. These augmentations help to convert the images for the model to predict perfectly, having more variation of images in the dataset.

- Image Size: 230x230
- Batch Size: 32
- Rescale: 1./255
- Width Shift Range: 0.2
- Height Shift Range: 0.2
- Shear Range: 0.2
- Zoom Range: 0.2
- Horizontal Flip: True

#### 3.2.5.2 Non-Overlapping Segmentation Data Preprocessing

All fish were annotated using the Roboflow annotation platform. We have used instance segmentation for each image and labeled each class carefully. Several data augmentation approaches were used to enrich the dataset for better model performance. The dataset initially included 2000 photos that had been scaled to a consistent 640x640 pixel size. Each training example produced two enhanced outputs, increasing the dataset size to 3600 pictures through augmentation. Shear transformations of $\pm 10°$ in both horizontal and vertical axes, 90-degree rotations in both clockwise and counterclockwise directions, and horizontal and vertical flips were among the augmentation approaches. Additionally, to provide color variations and improve robustness, saturation modifications were made within the range of -20% to +20%. In order to improve the model's ability to generalize to new data, these augmentation techniques sought to broaden the dataset's variety.

- Flip: Horizontal, Vertical
- 90° Rotation: Clockwise, Counter-Clockwise
- Shear: $\pm 10°$ Horizontal, $\pm 10°$ Vertical

- Saturation: Between -20% and +20%

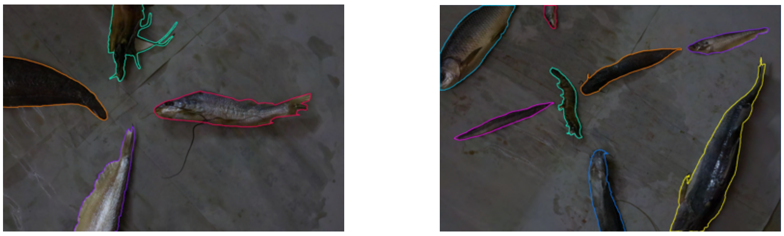

Figure 3.14: Non-Overlapping Fish Annotation

#### 3.2.5.3 Overlapping Segmentation Data Preprocessing

Roboflow was used for data augmentation techniques to improve the dataset and annotation data using instance segmentation. After the initial dataset of 100 photos was scaled to 640x640 pixels, 260 images were added using augmentation. Shear transformations of ±10° in both horizontal and vertical directions, 90-degree rotations (clockwise and counterclockwise), horizontal and vertical flips, and saturation adjustments between -20% and +20% were among the methods used. The model was better able to generalize to different settings thanks to these augmentations, which increased dataset variety. Roboflow expedited the procedure, guaranteeing effective augmentation for enhanced segmentation performance and accuracy.

- Flip: Horizontal, Vertical
- 90° Rotation: Clockwise, Counter-Clockwise
- Shear: ±10° Horizontal, ±10° Vertical
- Saturation: Between -20% and +20%

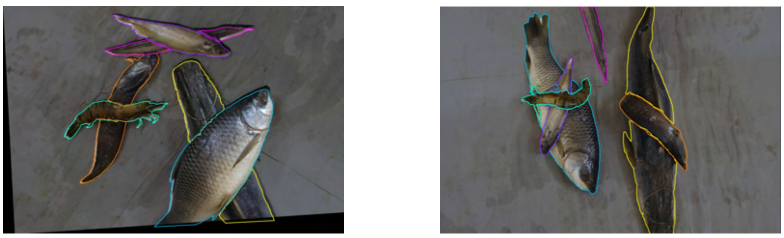

Figure 3.15: Overlapping Fish Annotation

## 3.3 Model Implementation

### 3.3.1 Introduction To Model Implementation

CNN is a class of deep learning designed primarily for image recognition, classification and segmentation tasks by mimicking the human visual cortex's way of processing visual information. CNN excels at recognizing patterns in images through their hierarchical architecture, with convolutional layers extracting features, pooling layers reducing dimensionality, and fully connected layers performing final classification. Collectively, the CNN models represent significant milestones in the evolution of deep learning for image processing, each introducing innovations in architecture that address the challenges of training deep neural networks, improving accuracy, and managing computational complexity. Although each model varies in depth, design principles, and optimization strategies, they all contribute to pushing the boundaries of what CNNs can achieve in visual recognition tasks.

### 3.3.2 Fish Classification

The models that have been implemented in this sector are Basic CNN, Inception-v3, VGG16 and ResNet-50. Among the notable advancements of CNNs is the Inception v3 model, a deep learning network known for its complex architecture that uses inception modules, which allow the model to capture information at various scales simultaneously. In addition to employing multiple convolution filters of different sizes within the same layer, Inception v3 can efficiently handle image variability and complexity. Similarly, VGG16, a highly popular deep learning model, is known for its simplicity yet effectiveness. It consists of 16 layers of convolution and fully connected layers, leveraging small filters (3x3) to systematically increase the depth of the network while retaining high accuracy in image classification. The architecture emphasizes the depth of the network over the width, a significant shift from earlier models. On the other hand, another important model is ResNet, specifically ResNet-50, which introduces the concept of residual learning. ResNet addresses the vanishing gradient problem by adding shortcut connections or skip connections, allowing the model to skip layers, thus making it easier to train deeper networks without losing accuracy which enables models to be significantly deeper than previous architectures while still maintaining efficiency.

#### 3.3.2.1 Inception-v3

By creating a probability distribution across several fish classes, the architecture outlined uses a complex picture classification model that uses the Softmax function in its final output layer to enable multi-class classification. In this model, the dense layer sizes are structured with a fully connected layer comprising 512 neurons, followed by another fully connected layer with 256 neurons, both utilizing ReLU activation functions along with L2 regularization to combat overfitting and enhance model robustness. The input layer size is defined as (230, 230, 3), indicating that the images are resized to dimensions of 230x230 pixels, with three color channels representing the RGB format. Moreover, it is important to note that the Inception-v3 base model inherently includes these layers within its architecture, designed to effectively downsample the spatial dimensions of the feature maps generated during convolutional operations. Multiple inception modules that leverage varying filter sizes (1x1, 3x3, and 5x5) to capture intricate features from the input images. Additionally, the model employs factorized convolutions, which streamline the computational load by substituting larger convolutional filters with smaller, more efficient ones, thereby maintaining the ability to extract detailed visual features without excessive resource consumption. The pre-trained Inception v3 base, which forms the basis for feature extraction, is the first layer in this architecture. A Global Average Pooling 2D layer follows, which minimizes the spatial dimensions of the feature maps, ensuring that the following fully connected layers receive a condensed representation of the extracted features. In summary, the architecture integrates multiple key components, including two dense layers with 512 and 256 neurons respectively, again dropout layers set to a dropout of 0.5 to enhance generalization, L2 regularization applied to the dense layers, and a concluding output layer with the softmax activation function that facilitates classification into various fish categories, effectively optimizing the model's performance in image classification tasks.

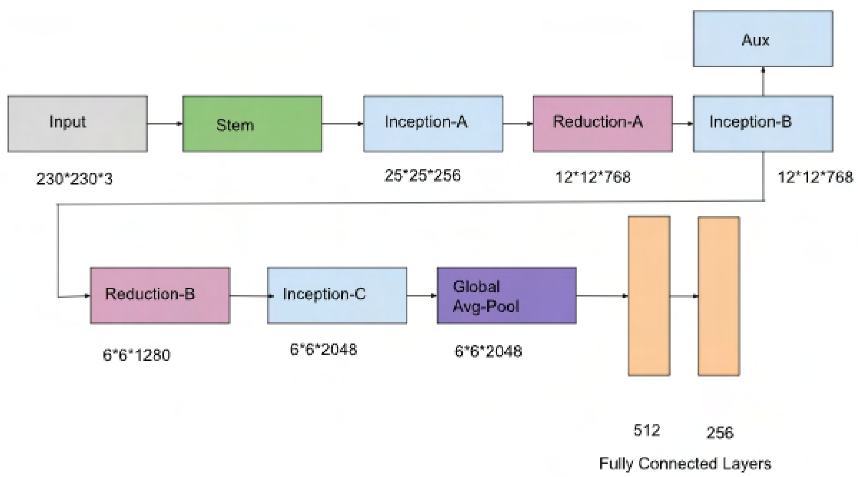


Figure 3.16: Architecture Of Inception-v3

In Inception-v3, the stem is a critical initial section of the architecture designed to efficiently process input images and extract low-level features. Moreover, the stem consists of a series of convolutional layers followed by max pooling operations that help reduce the spatial dimensions while preserving essential information. In addition, specifically, the stem begins with a 3x3 convolution with a stride of 2, utilizing 32 filters which serves to increase the depth of the feature maps while reducing the spatial resolution. Later on, this is followed by a second 3x3 convolution with a stride of 2 again but this time using 32 filters and further downsampling the feature maps. On the other hand, a 3x3 convolution with a stride of 2 and 64 filters is applied for transitioning to deeper feature extraction. After these convolutions, a 3x3 max pooling layer is utilized to downsample the feature maps followed by a 1x1 convolution that increases the dimensionality, using 80 filters. In conclusion, the stem effectively prepares the input for the subsequent inception modules, optimizing the model's ability to capture a wide range of features across different spatial resolutions while maintaining a computationally efficient architecture suitable for complex image classification tasks.

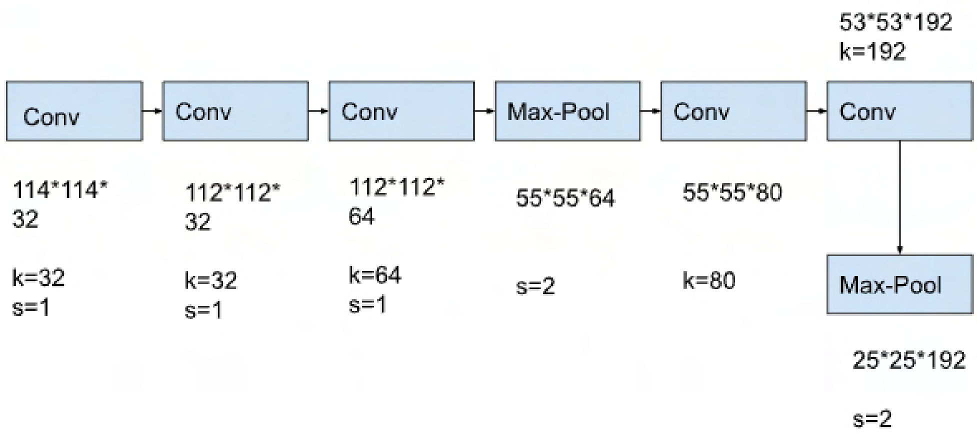


Figure 3.17: Architecture Of Inception-v3 Inner Stem

#### 3.3.2.2 VGG-16

A well-known deep convolutional neural network known for its efficiency and ease of use in image classification applications is the VGG16 architecture. Even though, VGG-16 operates on input images resized to (230, 230, 3) pixels, which reflects a resolution of 230x230 with three color channels (RGB). The architecture is characterized by its sequential arrangement of layers, primarily consisting of a total of 13 convolutional layers followed by a total of 5 max pooling layers. In this case, each convolutional layer employs small filters of size 3x3, which allows the network to learn intricate features while maintaining a manageable number of parameters. The max-pooling layers, utilizing 2x2 windows with a stride of 2, are important for downsampling the feature maps. Initially, the convolutional layers detect basic features such as edges and textures. As the data flows through the network, deeper layers learn to identify more complex patterns, shapes, and even high-level representations like objects and scenes. There are two dense layers in the architecture, with 512 and 256 neurons, respectively, following the convolutional and pooling layers. The ReLU activation function is used by both dense layers, which contributes to the model's non-linearity and facilitates effective learning and overfitting. The architecture incorporates dropout layers with a rate of 0.6 between the dense layers. This technique randomly drops a fraction of neurons during training, encouraging the model to generalize better. Lastly, a dense layer with a softmax activation function makes up the output layer.

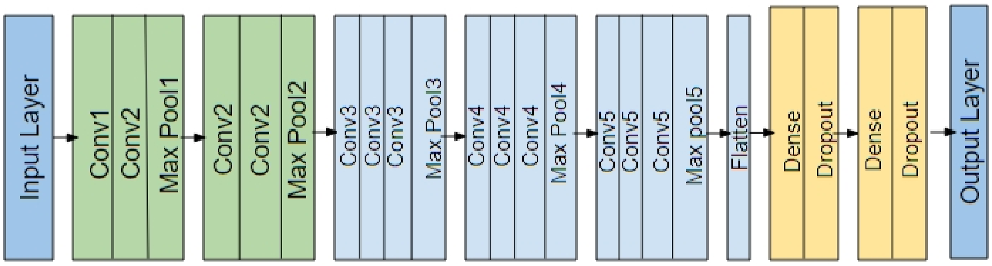


Figure 3.18: Architecture Of VGG-16

#### 3.3.2.3 ResNet-50

The ResNet-50 architecture is a powerful deep-learning model for image classification. The model accepts input images resized to (230, 230, 3) pixels, ensuring compatibility with various image datasets. ResNet-50 comprises 49 convolutional layers and 1 global average pooling layer. These convolutional layers utilize filters of varying sizes, primarily 3x3 convolutions, applied in a series of blocks that allow for the extraction of hierarchical features. Each block includes skip connections that add the input of a block to its output, facilitating the learning of residuals. After the convolutional layers, a Global Average Pooling 2D layer reduces the spatial dimensions, providing a more manageable input for the subsequent dense layers. The architecture includes a fully connected dense layer with 1024 neurons, utilizing ReLU activation, followed by a dropout layer with a of 0.5 to prevent overfitting. The final output layer consists of a dense layer with 10 neurons and a softmax activation function, which provides class probabilities for the classification task. This design, integrating convolutional layers with residual connections, makes ResNet-50 highly effective for a wide range of computer vision applications.

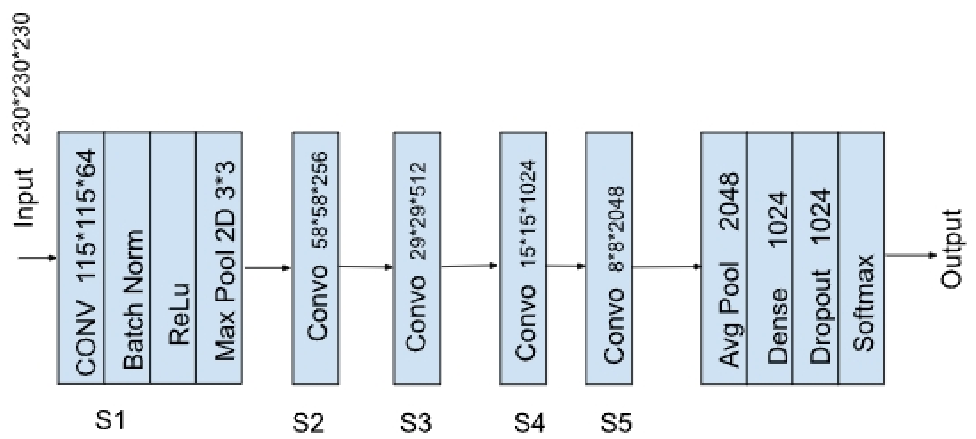


Figure 3.19: Architecture Of ResNet-50

#### 3.3.2.4 Basic CNN

The architecture of the Basic CNN model is a deep machine learning model designed for classification image problems, specifically tailored to classify fish species. We have applied 1 additional convolution and max pooling layer on top of the basic CNN. Moreover, this model begins with an input layer that accepts images of sizes (230, 230, 3), corresponding to the resized input images used for processing the trained model. The model has a series of properties of four convolutional layers, which use 128 and 256 filters in varying configurations, employing 3x3 kernel sizes and the ReLU activation function to extract features from the images each convolutional layer is pursued by a layer called Batch Normalization to secure the learning process and improve convergence speed and to reduce spatial dimensions, a Max Pooling layer with a pool size of (2, 2) is applied after each pair of convolutional layers, effectively down-sampling the feature maps and retaining essential information. After the convolutional blocks, the model transitions into the fully connected layers, beginning with a flattened layer that mutates the 2D featured maps into a 1D vector. The first fully connected dense layer contains 256 neurons with L2 regularization, followed by another Batch Normalization layer and a dropout layer with a rate of 0.3 to mitigate overfitting. This is succeeded by a second dense layer with 128 neurons, also employing L2 regularization, Batch Normalization, and dropout. In conclusion, a dense layer that has several neurons which is equal to the number of classes in the dataset, is equipped with a softmax activation function to provide class probabilities. Overall, this architecture, combining convolutional layers with regularization techniques, is well-suited for robust feature extraction and classification tasks in computer vision applications. The pre-trained model architecture can be built upon this structure, leveraging transfer learning by initializing the convolutional base with weights pre-trained on a large dataset like ImageNet, allowing the model to benefit from prior knowledge of feature extraction. In conclusion, by unfreezing the last few layers for fine-attuning, the model can adapt to the nuances of the specific dataset while retaining the foundational understanding learned from the extensive training on ImageNet.

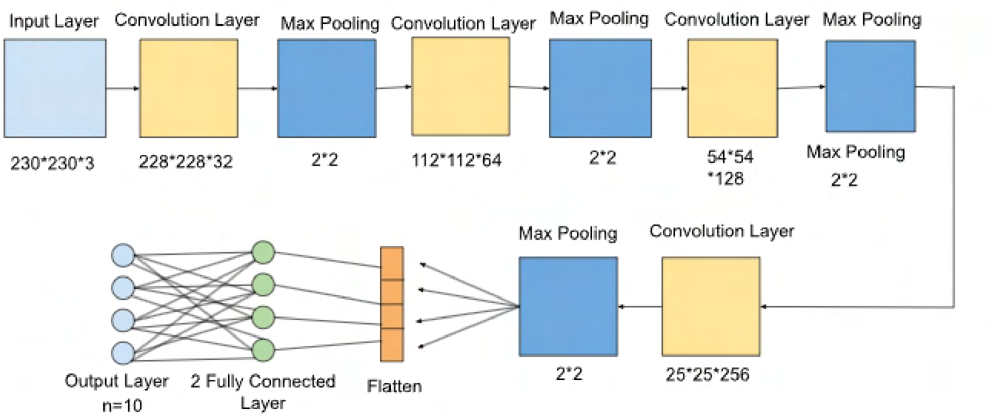


Figure 3.20: Architecture Of Basic CNN

### 3.3.3 Overlapping and Non-Overlapping Fish Segmentation

The models that have been implemented in this sector are YOLOv8, YOLOv9 and YOLOv11 segmentation models. One of the remarkable models among the CNN models are the YOLO segmentation models which are known for their speed and efficiency making them suitable for real-time applications and segmentation. Their strengths in real-time performance and segmentation have contributed to their general adoption in various industries. The YOLOv8 model is known for its high accuracy, real time performance, and adaptability for object detection, segmentation and classification. Followed by that YOLOv9-seg has improved accuracy than YOLOv8 which is suitable for small object detection and segmentation. The YOLOv11 model is known for its fastest inference speed among the YOLO series with strict latency requirements making it a good candidate for real time segmentation.

#### 3.3.3.1 YOLOv8

YOLOv8 is an advanced object detection model known for its lightweight yet powerful architecture, making it ideal for real-time applications on edge devices, where computational resources may be limited. It builds on the YOLO (You Only Look Once) series by optimizing detection speed without sacrificing accuracy, achieving a balance that is crucial for environments where efficiency is paramount. YOLOv8 leverages modern deep learning techniques, such as EfficientNet backbones and attention mechanisms, to improve both performance and adaptability. This makes it capable of processing complex tasks like multi-class object detection and segmentation on devices with limited power and memory. Its design is particularly suited for field research and remote monitoring, where quick, accurate detection is required in real-time by reducing the computational overhead traditionally associated with deep learning models, YOLOv8 can run on low-power devices like mobile phones, embedded systems, and IoT devices without compromising speed or precision. The model's lightweight nature allows it to detect and classify objects with minimal latency, even under challenging conditions like dynamic environments or varying image qualities. Overall, YOLOv8's combination of efficiency, accuracy, and versatility makes it an ideal choice for real-time object detection and segmentation tasks in resource constrained settings, such as fish segmentation studies.

Backbone: The backbone of YOLOv8 is the foundation of the model, responsible for extracting critical features from input images. It employs efficient convolutional layers, which are designed to capture hierarchical spatial information and understand the various components of an image. These convolutional layers focus on optimizing feature extraction while minimizing computational load, a crucial factor when working with resource-constrained devices. YOLOv8's backbone also incorporates advanced attention mechanisms, such as channel and spatial attention, to enhance the model's ability to focus on the most relevant parts of the image, thereby improving feature extraction. These mechanisms help the model differentiate between important and less important features, allowing it to prioritize areas of interest. Additionally, the backbone is built to be adaptable to varying scales and environmental conditions, ensuring that the model can maintain performance across a wide range of object sizes and diverse backgrounds. The backbone's efficiency and ability to adapt to different conditions make it highly effective in real-time object detection and segmentation tasks, such as fish classification in field research, where images may vary in size, quality, and lighting conditions. The backbone's role is critical in ensuring that YOLOv8 achieves its high-speed performance without compromising accuracy.

Neck: The neck in YOLOv8 plays an essential role in refining and aggregating the feature maps produced by the backbone. It integrates specialized modules that focus on merging context from multi-scale spatial features to ensure that the model can handle objects of different sizes and complexities. These modules enhance the model's ability to capture both fine-grained details and broader contextual information. For example, the neck often uses feature pyramid networks (FPNs) or other advanced architectures to combine high-resolution low-level features with low-resolution high-level features. This aggregation helps to preserve critical information at various scales, ensuring that both large and small objects are detected accurately. In the context of fish segmentation, where the fish may appear at different sizes or angles in the image, the neck ensures that all relevant information is maintained and passed on to the next stages of the network. By refining the feature maps, the neck improves the robustness of the model's predictions, allowing it to handle various challenges such as cluttered backgrounds, overlapping objects, or varying lighting conditions. This refinement process is vital in generating accurate and consistent results, especially when the model is applied to real-time tasks in dynamic environments.

SPPF (Spatial Pyramid Pooling-Fast): Spatial Pyramid Pooling-Fast (SPPF) is a specialized module in YOLOv8 that significantly improves the model's ability to detect objects at varying scales. By pooling spatial features at multiple scales, SPPF captures both global context (such as the overall layout of the image) and local context (such as the fine details of individual objects). This dual approach enables the model to effectively recognize and localize objects, even when they vary in size, shape, or orientation. In the case of fish segmentation, SPPF excels at detecting fish in different sizes, from smaller species to larger ones, by pooling information from different spatial levels of the image. It does this by applying different levels of pooling to feature maps, creating representations that capture various levels of granularity. This results in a more versatile model that can detect objects of different sizes and shapes, regardless of the conditions in the input image. The "Fast" aspect of SPPF refers to its optimized design, which allows the pooling operations to occur quickly without adding significant computational overhead. This is critical for real-time detection, as it helps YOLOv8 maintain its speed while enhancing its ability to handle varying object scales, a necessity for applications like fish segmentation in dynamic and diverse environments.

Head: The head of YOLOv8 is responsible for making final predictions based on the refined feature maps passed through the backbone, neck, and other network modules. It is designed to output precise and consistent predictions for object localization, classification, and segmentation. The head is crucial for achieving high detection accuracy, especially when dealing with objects of different sizes and scales. It excels in both small and large object detection tasks by using specialized techniques to refine the bounding boxes and class probabilities generated for each detected object. For example, the head typically uses anchor boxes to predict the locations of objects in the image, ensuring that objects are localized with high accuracy. In the case of fish segmentation, where accurate delineation of fish boundaries is essential, the head's design allows the model to provide precise segmentation maps, ensuring that each fish is accurately identified and separated from the background and other objects. By producing high-quality predictions across various scales, the head improves the overall performance of YOLOv8 in complex environments, such as aquatic ecosystems, where the fish may vary in shape, size, and pose. The head's ability to handle both small and large objects effectively makes it a key component in ensuring the model's robustness and precision in real-time detection and segmentation tasks.

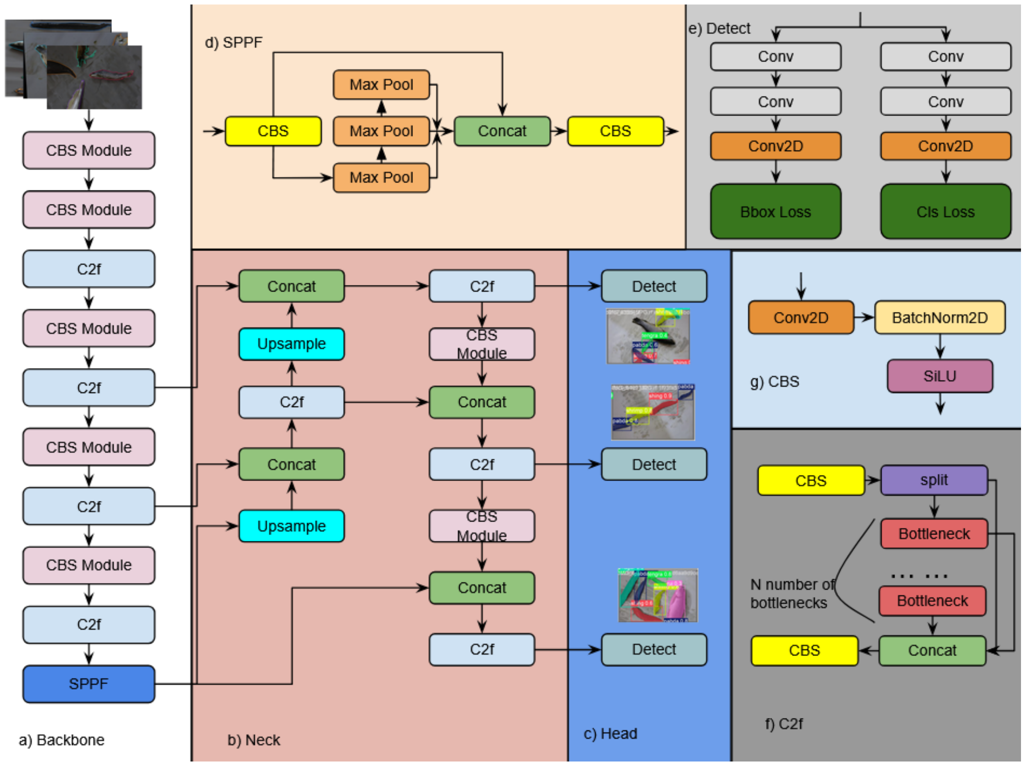


Figure 3.21: Architecture Of YOLOv8

#### 3.3.3.2 YOLOv9

YOLOv9 strikes a balance between computational efficiency and high detection accuracy which makes it ideal for real time applications. Moreover,this version prioritizes a lightweight design and scalability while ensuring strong performance, particularly in multi-object detection scenarios. In addition, YOLOv9's architecture incorporates advanced optimization techniques that streamline feature extraction thus making it suitable for use in environments with limited computational resources. However, the model's ability to process data rapidly without compromising accuracy makes it a perfect choice for tasks requiring real time detection and segmentation. In our fish segmentation experiments, YOLOv9 excelled in handling challenging datasets, such as those with overlapping fish and varying lighting conditions and the backbone employs RepNCSPELAN modules, which enhance feature extraction efficiency by focusing on spatial representation. Meanwhile, the neck's use of concatenation and upsampling layers ensures smooth integration of features from multiple scales, contributing to better accuracy in detecting objects of varying sizes. The detection head is optimized for multi-scale predictions, enabling the model to reliably detect small, occluded, or overlapping objects. In conclusion,combining speed with precision the YOLOv9 proved instrumental in advancing our fish classification studies with delivering consistent performance and scalability across diverse aquatic datasets.

Backbone: The RepNCSPELAN modules in YOLOv9 are designed for efficient feature extraction, focusing on spatial representation and scalability. These modules play a critical role in enhancing the model's adaptability to diverse environments which include aquatic datasets. Moreover, employing lightweight computation techniques, RepNCSPELAN reduces the complexity of feature extraction while preserving rich spatial information. This is particularly useful in fish segmentation, where subtle details such as fin edges and scales must be captured accurately. The design ensures that high resolution spatial features are retained in the early layers while gradually transitioning to more abstract and semantically meaningful representations in deeper layers. However,this hierarchical extraction process not only improves the model's generalization across varying scales and lighting conditions but also ensures rapid inference. Additionally, the modules are optimized to handle noisy and cluttered environments, such as overlapping fish, by enhancing spatial encoding and reducing redundancies in the feature maps. By focusing on efficiency without compromising detail, the backbone ensures the model is strong, adaptable and capable of handling real world aquatic datasets effectively.

Neck: The neck employs concatenation, upsampling layers and CBFuse modules, seamlessly integrating multi-scale features for better detection of objects of various sizes and complexities. Moreover,this component is crucial for merging features from different levels of the backbone to produce a cohesive and informative feature map which combines fine grained details from lower layers with broader contextual features from higher layers, the neck ensures that the model captures the diverse characteristics of fish, such as varying body shapes and intricate patterns. CBFuse, in particular, enhances the integration of features by maintaining consistency across scales and reducing information loss during fusion. In conclusion,this multi scale approach enables the model to excel in detecting overlapping or partially occluded objects, a common challenge in fish segmentation datasets. The upsampling layers further ensure that critical spatial details are preserved and amplified, improving overall detection accuracy. The neck's design contributes significantly to the model's robustness in handling cluttered environments and diverse aquatic scenarios. This multi-scale fusion ensures that the model captures diverse characteristics of the fish, from their body shapes to finer patterns, making it effective in scenarios with cluttered or overlapping objects.

Head: The detection head focuses on multi-scale predictions, optimizing detection accuracy for small, occluded, or overlapping objects. This functionality is essential for precise segmentation in real-world aquatic environments. The head employs advanced mechanisms to generate bounding boxes and classification scores with high precision, even in challenging conditions. By utilizing multi-scale prediction layers, the head can adapt to objects of varying sizes, ensuring accurate localization of small fish alongside larger ones. Additionally, the head incorporates confidence scoring and non-maximum suppression to eliminate redundant or false-positive detections, enhancing overall segmentation reliability. This is particularly important in fish datasets, where objects often overlap or exhibit similar features. The design also allows for refinement of predictions, ensuring that even subtle details, like the edges of fins or the separation between closely packed fish, are accurately captured. By balancing precision and scalability, the detection head elevates the performance of YOLOv9, making it an indispensable tool for biodiversity monitoring and fish classification.

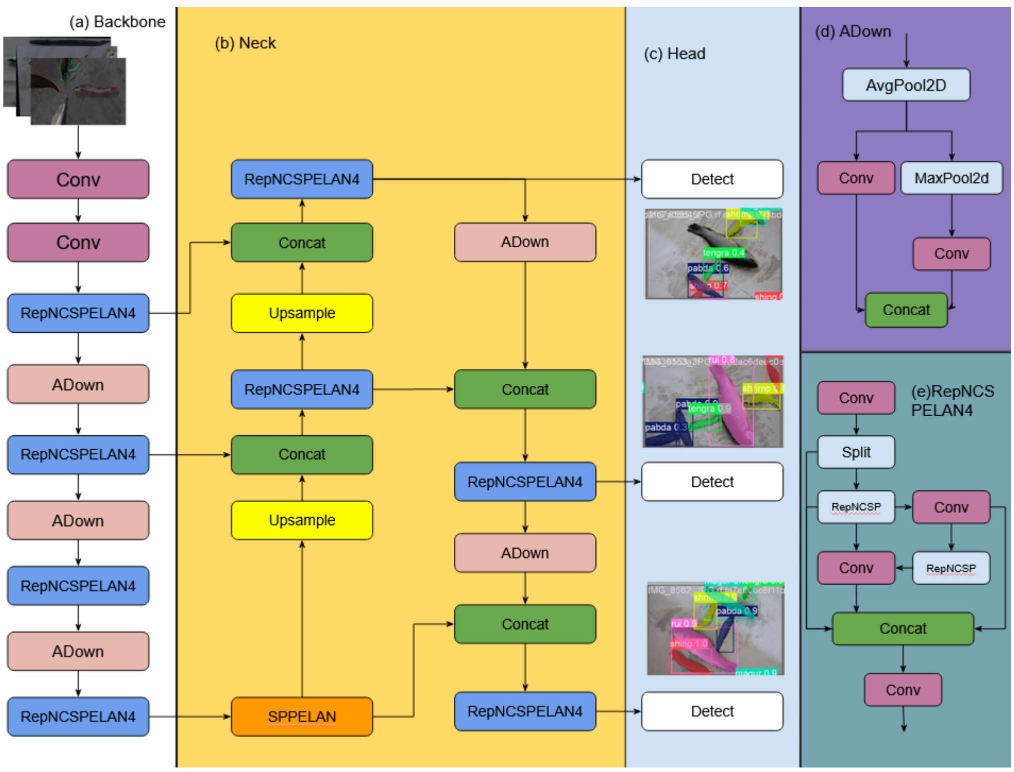


Figure 3.22: Architecture Of YOLOv9

#### 3.3.3.3 YOLOv11

YOLO v11 integrates advanced transformer-based architectures and state-of-the-art feature extraction techniques to address the demands of complex detection tasks. The model is designed to improve spatial awareness and multi-scale feature representation, ensuring high precision even in cluttered or complex environments. By employing transformers, YOLO v11 enhances its ability to capture relationships between distant features in the image, which is especially crucial for detecting overlapping objects. Additionally, the model incorporates advanced modules that optimize computational efficiency while maintaining accuracy, making it suitable for real-time applications. YOLO v11's primary purpose is to provide robust segmentation capabilities in scenarios like fish classification, where distinguishing between overlapping or closely placed fish is essential. Its adaptability to different scales and ability to generalize across datasets with varying complexities make it a powerful tool for ecological studies and biodiversity monitoring. Validated in our experiments, the model excels in fish segmentation tasks, particularly with datasets featuring varying angles and lighting conditions, delivering consistently reliable results.

Backbone: The backbone leverages advanced C3K2 layers, which progressively reduce spatial dimensions while expanding channel depth. These layers enable the model to effectively extract both local and global features crucial for detecting objects in complex environments. C3K2's design focuses on hierarchical feature extraction, which captures high-resolution details in the initial layers and transitions to broader contextual features in deeper layers. This balanced extraction allows the model to generalize well across diverse datasets, including those with varying angles, scales, and lighting conditions. By maintaining high spatial resolution during the early stages, the backbone ensures that finer details, such as the texture or edges of overlapping fish, are preserved. As spatial dimensions are progressively reduced, channel depth increases, allowing the model to focus on more abstract representations, such as the shape or orientation of objects. This intricate balance between feature detail and abstraction makes the backbone a robust foundation for downstream tasks like fish segmentation, where precision and adaptability are paramount.

Neck: The neck incorporates C2PSA modules to refine multi-scale features and improve feature fusion, enhancing detection accuracy for objects of different sizes. C2PSA modules are specifically designed to prioritize channel and spatial attention, allowing the model to focus on the most relevant features within each layer. This refinement ensures that both fine-grained and large-scale details are effectively combined, which is particularly beneficial for detecting small or overlapping fish in cluttered scenes. Additionally, the neck serves as a bridge between the backbone and head, integrating features from different scales through upsampling and concatenation processes. This multi-scale approach enhances the model's ability to capture the diverse characteristics of fish, such as subtle fin patterns or varying body shapes. By improving feature alignment and fusion, the neck plays a critical role in optimizing the model's overall performance for fish segmentation tasks, ensuring accuracy across a range of environmental conditions and dataset complexities.

Head: The detection head generates predictions across multiple scales, optimizing both localization and classification accuracy. Its design ensures reliable detection of small, overlapping, or occluded objects, crucial for fish segmentation tasks. The multi-scale prediction mechanism allows the head to adapt to objects of varying sizes, ensuring precise bounding box generation even in challenging scenarios. For instance, in datasets with overlapping fish, the head employs advanced classification layers to differentiate between objects while maintaining high spatial precision. Furthermore, the head integrates confidence scoring mechanisms to assess the reliability of each prediction, reducing false positives and improving overall segmentation quality. This adaptability makes the head indispensable for ecological applications, where accurate identification and segmentation of fish species are vital for biodiversity monitoring and conservation efforts. By focusing on scalability and precision, the detection head elevates the model's effectiveness in real-world aquatic environments.

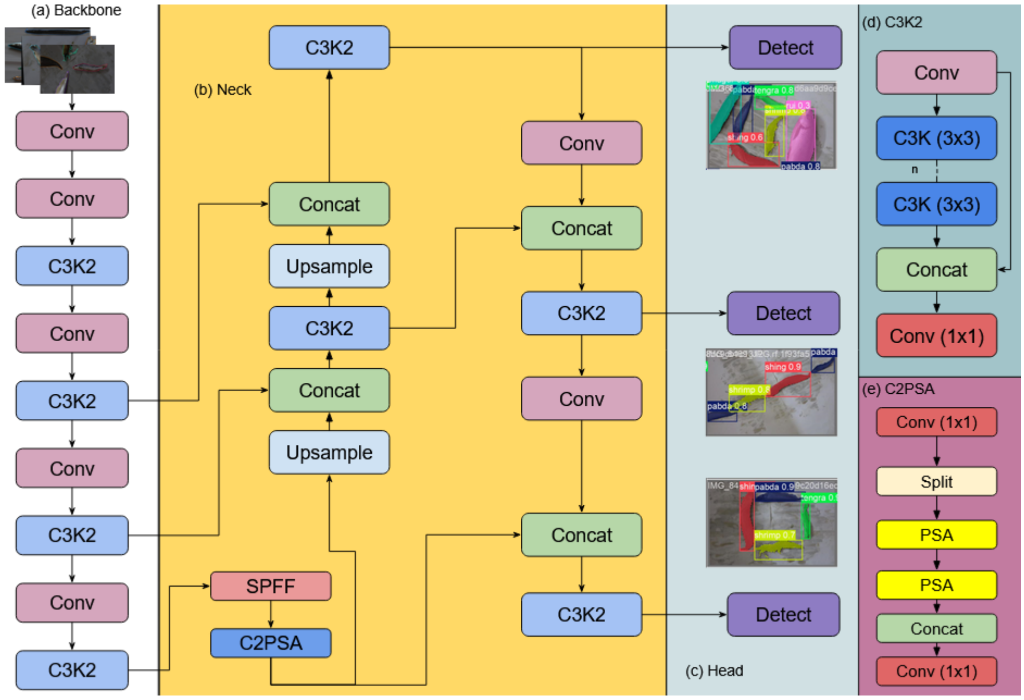


Figure 3.23: Architecture Of YOLOv11

### 3.3.4 Model Architecture Comparison

| **Features** | **YOLO-V11** | **YOLO-V9** | **YOLO-V8** |
|---|---|---|---|
| Release date | September 2024 | February 2024 | January 2023 |
| Architecture | Enhanced CSP-Darknet with ELAN++, deeper PANet, **Transformer** modules for attention | Improved CSP-Darknet with ELAN+, refined PANet | Enhanced CSP-Darknet, PANet head, SPP, ELAN |
| Backbone | Transformer-augmented CSP-Darknet | More optimized CSPDarknet with ELAN+ | Improved CSP-Darknet with ELAN |
| Neck | Advanced PANet with Swin Transformer integration | Further improved PANet | PANet with improvements in PANet |
| SPPF | Yes | Yes | Yes |
| Model Variants | YOLOv11-tiny, YOLOv11, YOLOv11-x | YOLOv9-tiny, YOLOv9, YOLOv9-x | YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x |
| ELAN | Yes | Yes | Yes |
| Parameters | 5.8m (YOLOv11-tiny), 28m (YOLOv11), 55m (YOLOv11-x) | 6.5m (YOLOv9-tiny), 30m (YOLOv9), 60m (YOLOv9-x) | 6.2m (YOLOv8n), 12m (YOLOv8s), 25m (YOLOv8m), 43m (YOLOv8l), 68m (YOLOv8x) |
| Anchor-free | Yes | No | Yes |
| Training time | Shorter due to efficient parallelism | Moderate | Faster training with new features |
| Performance | High precision and | Better optimal **speed** | Diverse detection in |

Table 3.1: Comparison of YOLO Variants

# Chapter 4

# Performance And Analytics

## 4.1 Performance Parameters

### 4.1.1 Classification Parameters

The model's performance can be evaluated by using various parameters, that is listed in the table below. The model was trained on 80% of the dataset, with 20% reserved for testing. Moreover, in managing the computational load, a batch size of 32 was chosen, which allowed the model to process the images in manageable segments. Later, the fish images were resized to a target size of 230x230 pixels with an RGB channel included, which is a crucial step for ensuring that all images have a consistent input size for the model. After that every model was trained for 20 epochs, the execution environment was set to GPU, this greatly accelerated the training process, enabling the model to handle the computations more efficiently. In addition, we used the optimizer Adam which is a well known choice for deep learning models because of its adaptability and efficiency. It alters the learning rate of each parameter, making it particularly effective for training complex models like CNNs. The loss function that is applied was Categorical Cross Entropy, which is an ideal choice for multi-class classification problems. The class mode chosen was categorical, thus ensuring that the model treats the output as distinct categories rather than continuous values. The model's learning rate was set to 0.00001, a relatively low value, which helped to fine-tune the network without causing sudden shifts in weight updates. All these parameters were the same for every model.

| Parameters | Basic CNN, VGG-16, Inception-v3 and ResNet-50 |
|---|---|
| Training Data | 80% |
| Testing Data | 20% |
| Batch Size | 32 |
| Target Size | 230x230 |
| Epochs | 20 |
| Execution Environment | GPU |
| Optimizer | Adam |
| Loss Function | Categorical Cross Entropy |
| Class Mode | Categorical |
| Learning Rate | 0.00001 |

Table 4.1: Proposed Models and Their Parameters

### 4.1.2 Segmentation Parameters

The segmentation model's parameter can be evaluated using various parameters which can be observed in the table below. The models was trained using 80% of the dataset and the rest 20% was reserved for testing. In order to reduce the computational load, a batch size of 16 was chosen in order to process the images properly. The input size of the images were 640x640 pixels for this segmentation task. After that all the models were trained on 20 epochs, the execution environment was set to GPU which greatly accelerated the training process. The optimzer, AdamW was used for all the models. The class mode chosen was multi-label. The learning rate varied due in case of overlapping and non-overlapping dataset. For non-overlapping dataset, the learning rate was set to 0.000769 and for overlapping dataset, the learning rate was increased to 0.001. Other than that all the parameters were the same for every model.

| Parameters | YOLOv8, YOLOv9 and YOLOv11 |
|---|---|
| Training Data | 80% |
| Testing Data | 20% |
| Batch Size | 16 |
| Target Size | 640x640 |
| Epochs | 20 |
| Execution Environment | GPU |
| Optimizer | AdamW |
| Class Mode | Multi-Class |

Table 4.2: Proposed Models and Their Parameters

## 4.2 Model Accuracy

### 4.2.1 Classification Accuracy

#### 4.2.1.1 Basic CNN

The two graphs below represent the testing and training accuracy of the Basic CNN model as well as the testing and training loss of the model. We got a validation accuracy of 97.79% and a training accuracy of 93.17% using this model. The graph indicates that over time the performance of the model on the training data improves steadily. The validation performance also improves rapidly, eventually stabilizing at a high level, but for the first graph, the validation curve is slightly above the training curve reason can be model is predicting classes very quickly and the complexity is less. The same observation goes for the second graph. The validation loss is lower than the training loss because for the model it is easier to predict class due to less noise in validation data.

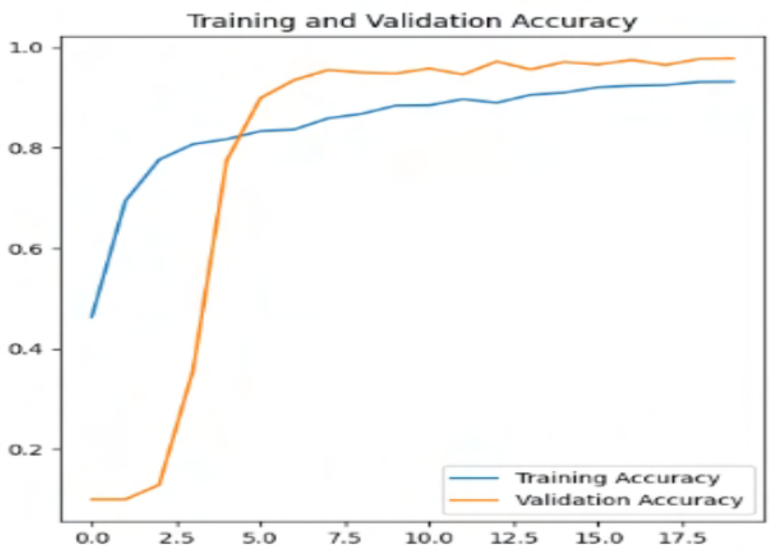


Figure 4.1: Training And Validation Accuracy Curve

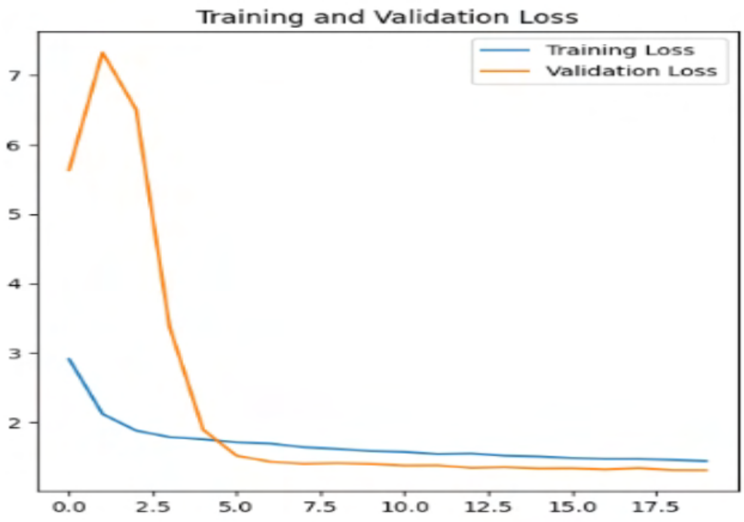


Figure 4.2: Training and Validation Loss Curve

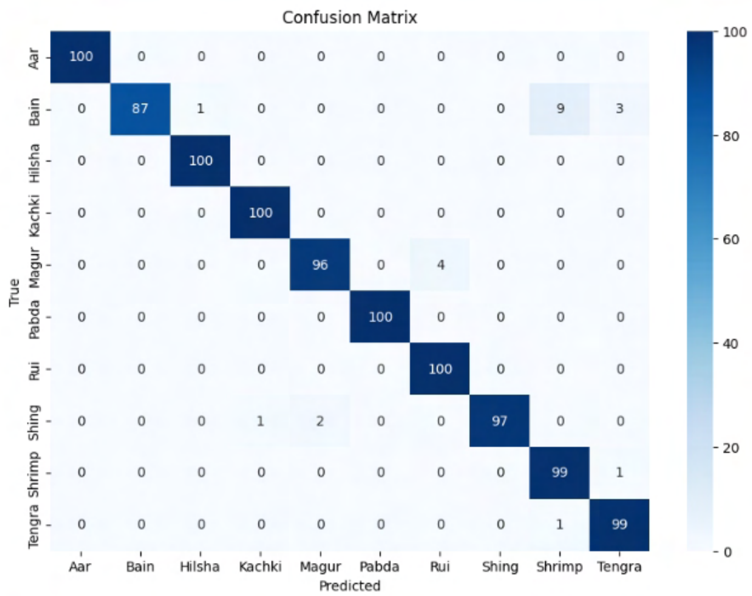


Figure 4.3: Confusion Matrix Of Basic CNN

The confusion matrix illustrated in the above figure classifies the performance of a Basic CNN model across multiple fish categories. The diagonal cells represent the number of correct predictions, with values of 100 fish samples of classification such as Aar, Hilsha, Kachki, Pabda, Shing, and Tengra; these categories were classified with 100% accuracy. However, some classes, particularly Bain, showed misclassification errors, where the model correctly identified 87 samples but misclassified 9 as Shrimp and 3 as Tengra, other classes like Pabda had a few misclassifications, where 4 samples were wrongly predicted as Rui. The color gradient from light to dark blue indicates the percentage of correct or incorrect predictions; darker shades reflect high accuracy, while lighter shades represent errors. Overall, the model performs well, but it struggles with distinguishing between certain classes, particularly those with similarities, like Bain, Shrimp, and Tengra and this suggests the need for improvements in the ability of the model to differentiate between these specific categories, perhaps through better feature extraction or more data for the confused classes. Despite these errors, the CNN achieves high accuracy for most categories, reflecting its general effectiveness.

#### 4.2.1.2 VGG-16

The two graphs below represent the testing and training accuracy of the VGG-16 model as well as the testing and training loss of the model. We got a validation accuracy of 99.40% and a training accuracy of 98.32% using this model. In the accuracy graph, both lines eventually converge near the top, which indicates the model is achieving high accuracy on both training and validation data. The validation accuracy starts higher than the training accuracy and remains slightly above or equal throughout suggesting that the model is slightly overfitting. It might also indicate that the validation data is easier to predict or that the model benefits from regularization during training. For the second graph, Training loss is a little higher than validation which means it is easier to predict class due to less noise in validation data.

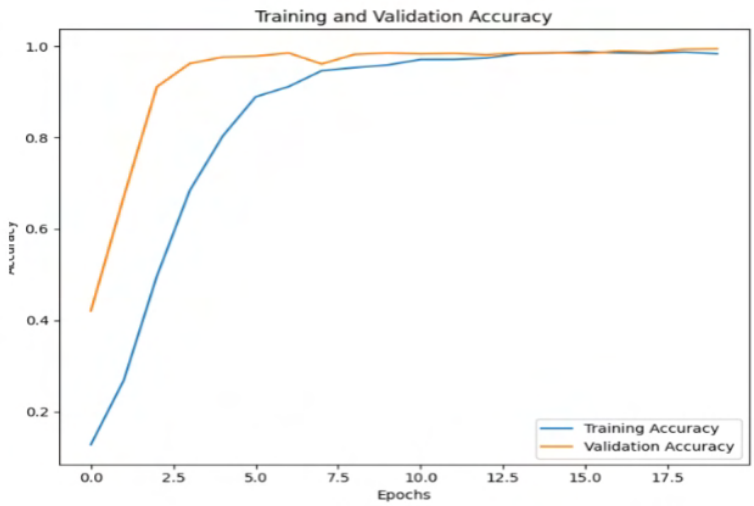


Figure 4.4: Training and Validation Accuracy Curve

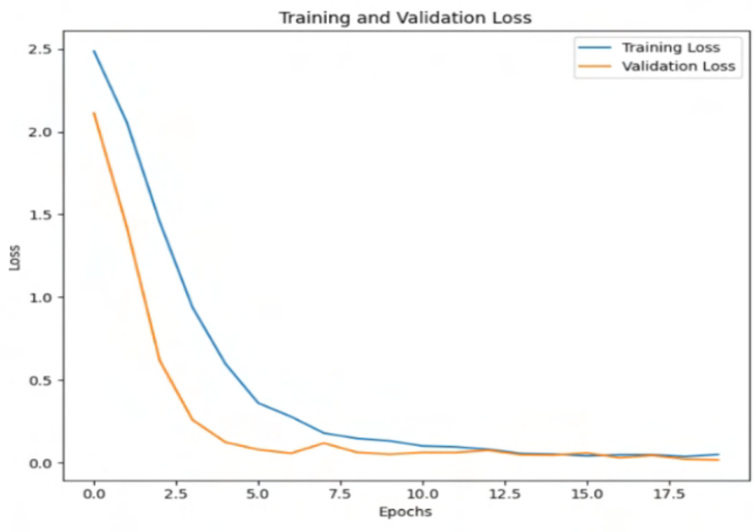


Figure 4.5: Training and Validation Loss Curve

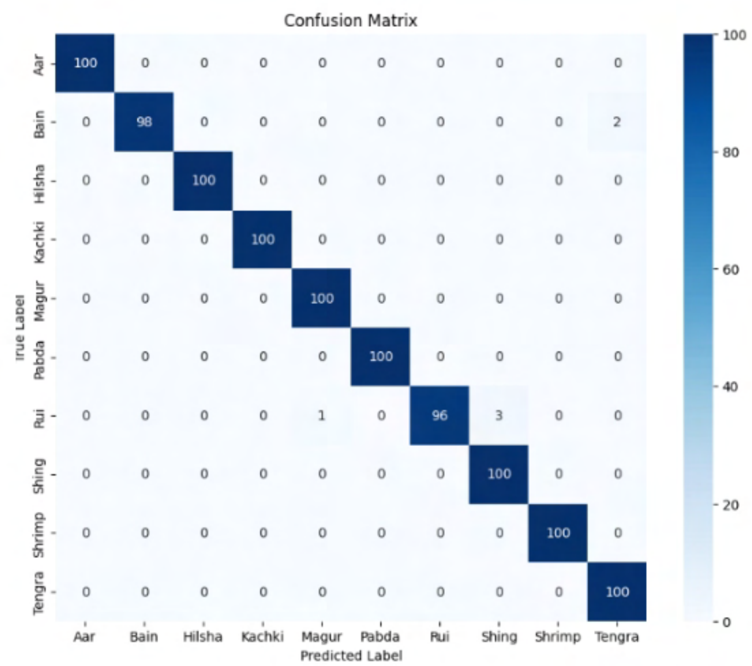


Figure 4.6: Confusion Matrix of VGG-16

The confusion matrix demonstrates the model's classification performance for different fish categories. The diagonal cells represent correct classifications, where several categories, including Aar, Hilsha, Kachki, Pabda, Shing, and Tengra, achieved perfect accuracy, meaning the model classified all samples from these classes correctly. However, the model shows some errors in a few categories. For instance, 98 Bain samples were classified correctly but 2 were misclassified as Tengra. Similarly, in the case of Rui, while 96 samples were correctly predicted, 3 were incorrectly classified as Shing and 1 as Magur. The dark blue cells along the diagonal represent high accuracy in class predictions, while lighter off-diagonal cells highlight where misclassifications occurred. Although the model performs very well overall, with high classification accuracy across most categories, the presence of a few misclassifications suggests that there could be minor challenges in differentiating between some similar fish types, such as Bain, Rui, and Shing. In conclusion, it suggests the need for refinement to improve the performance of the model further, particularly in distinguishing between fish species with more subtle differences. Nonetheless, the overall accuracy is impressive, reflecting a strong classification ability.

#### 4.2.1.3 ResNet-50

We got a validation accuracy of 73% and a training accuracy of 67.82%.c The two graphs represent the model's accuracy as well as loss for both the training and validation datasets over 20 epochs. In the first graph, Model Accuracy, both training and validation accuracy improve steadily, with the validation curve slightly outperforming the training curve at several points. Moreover, the accuracy continues to rise throughout the epochs, reaching approximately 70% by the end and shows the model is learning well. The validation accuracy curve is higher than training because of its simpler complexity and it's predicting fish too quickly. The second graph, Model Loss, shows a decrease steadily in both training and validation loss which is a good sign but the loss is high. The reason can be that the model has not fully converged as the learning rate is high. The validation loss decreases more rapidly at first, showing a better fit in the early epochs, but then converges closely with the training loss, indicating stable performance without significant overfitting. Overall, the graphs show a positive learning trend with the model improving its predictive performance over time.

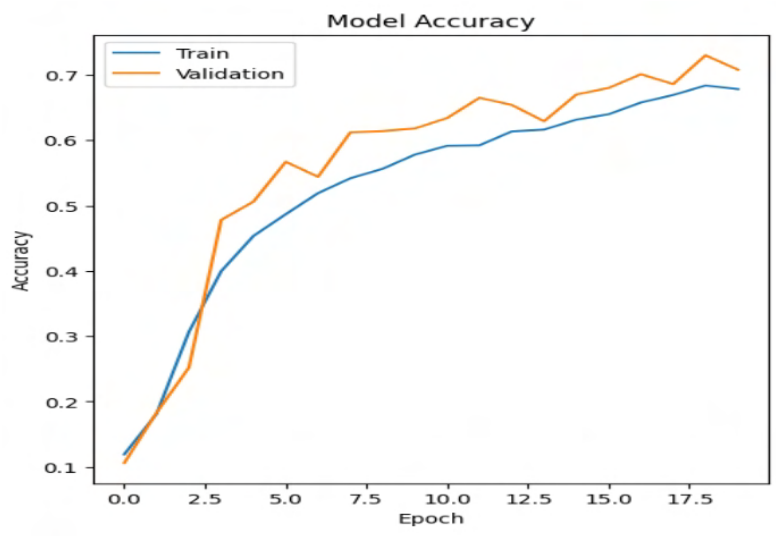


Figure 4.7: Training and Validation Accuracy Curve

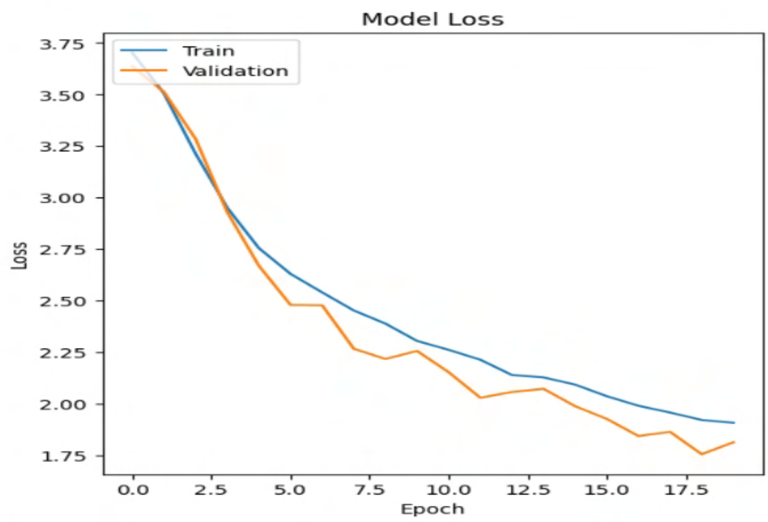


Figure 4.8: Training and Validation Loss Curve

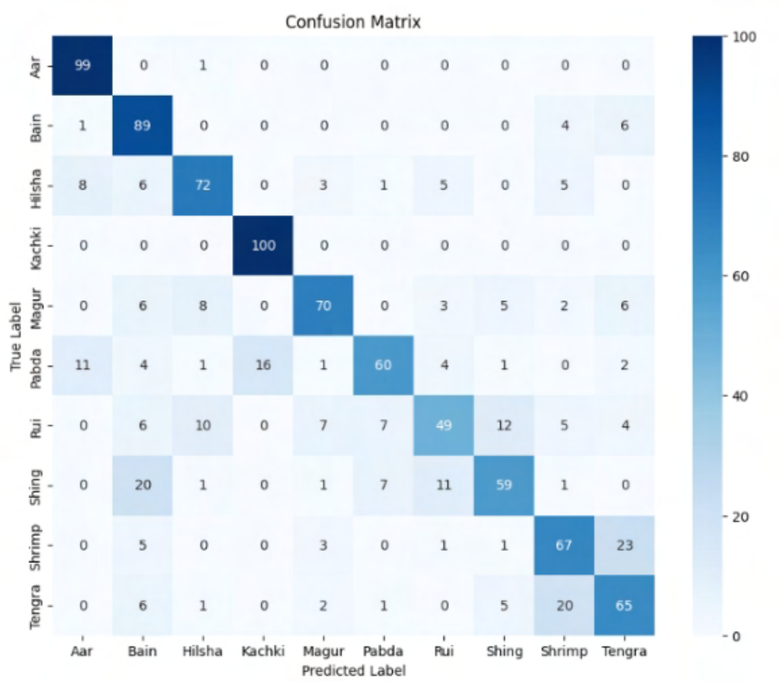


Figure 4.9: Confusion Matrix of ResNet-50

The confusion matrix presents the performance of ResNet-50 across 10 fish classes. Each row cell shows the true label, and each column cell shows the predicted label. The diagonal entries represent correct classifications. Moreover, the model performs very well in some classes, such as Kachki with 100% accuracy, Aar with 99 correct predictions, and Shing with 59 correct predictions, but struggles with others like Pabda and Rui. For instance, Pabda is often confused with Magur and Hilsha, with 16 misclassifications in total. Similarly, Shrimp is confused with Tengra, and Shing has some confusion with Rui. Misclassifications between fish that have similar visual features may explain these confusions. Overall, the model performs best with certain distinct classes but needs improvement in handling those with more overlap or similarity.

#### 4.2.1.4 Inception-v3

We got a validation accuracy of 99.69% and a training accuracy of 98.32% using this Inception-v3 model. The training accuracy starts low but continuously increases, reaching nearly 100% by the 15th epoch. The testing accuracy improves rapidly during the early epochs, stabilizing around the same point as the training accuracy. It suggests the model is learning well, and the validation accuracy closely mirrors the training accuracy, indicating good generalization without significant overfitting. However, validation loss is at the top of the training accuracy curve which is unusual. A possible reason can be less image augmentation. The model is learning too quickly on the validation dataset. The second graph displays the training and validation loss over the same 20 epochs. Initially, both the training loss blue line and validation loss red line are high but decrease rapidly as training progresses. At the end of training, both losses converge at around 1.0. Again, the testing loss curve is lower than the training loss curve indicating it's easy for the model to predict the validation data quickly. Altogether, these graphs demonstrate that the model has been well-tuned, achieving high accuracy.

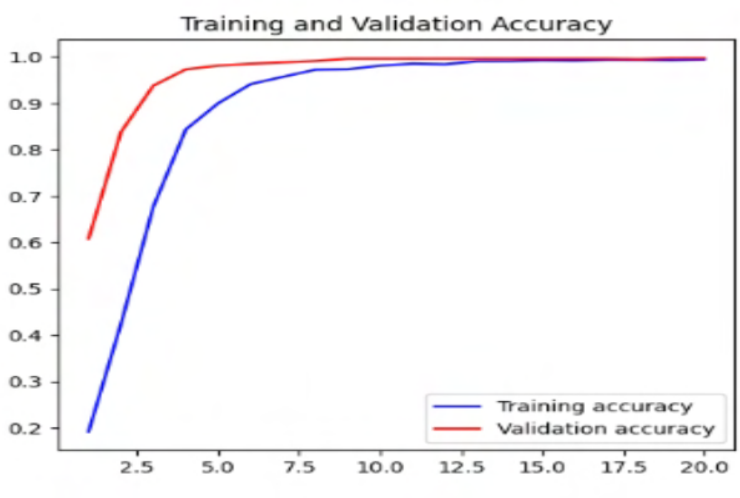


Figure 4.10: Training and Validation Accuracy Curve

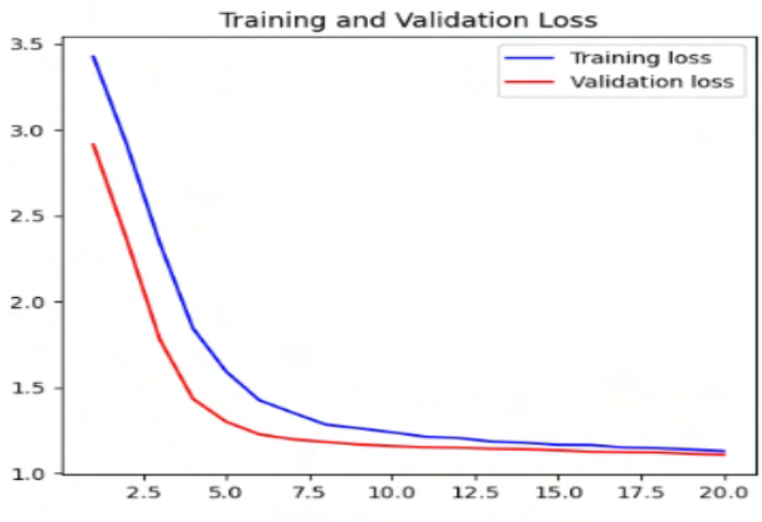


Figure 4.11: Training and Validation Loss Curve

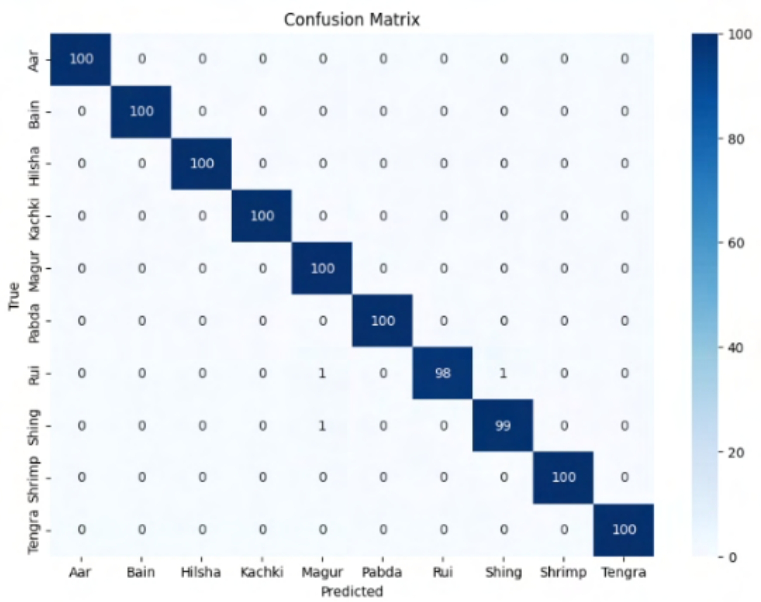


Figure 4.12: Confusion Matrix of Inception-v3

The confusion matrix displayed here presents the performance of Inception-v3 in predicting different types of fish species. Each row cell shows the actual fish species, while each column cell shows us the predicted species by the model. The diagonal cell in the middle indicate correct predictions, while the other diagonal cells represent misclassifications. The matrix shows excellent overall accuracy, with the majority of species such as Aar, Bain, Hilsha, Kachki, etc. being predicted with 100% accuracy. However, there are slight misclassifications for Rui and Shing, where two misclassifications are observed. One Rui fish was misclassified as Shing, and vice versa. Despite these small errors, the model performs well, especially for species like Shrimp and Tengra, where there are no misclassifications. This suggests the model is highly effective for most species but may struggle in distinguishing between visually or structurally similar fish like Rui and Shing. The color intensity in the heatmap from light to dark blue represents the density of correct predictions, with darker shades indicating higher accuracy. Overall, the confusion matrix reflects strong model performance with only minor room for improvement in a few specific categories.

### 4.2.2 Non-Overlapping Segmentation Accuracy

#### 4.2.2.1 YOLOv8

| Class | Box(Precision | Recall | mAP50 | map50-95) |
|---|---|---|---|---|
| All | 0.796 | 0.804 | 0.761 | 0.709 |
| Aar | 0.850 | 0.830 | 0.797 | 0.764 |
| Bain | 0.816 | 0.889 | 0.809 | 0.747 |
| Hilsha | 0.830 | 0.784 | 0.805 | 0.768 |
| Magur | 0.797 | 0.813 | 0.768 | 0.721 |
| Pabda | 0.753 | 0.775 | 0.705 | 0.666 |
| Rui | 0.811 | 0.813 | 0.772 | 0.757 |
| Shing | 0.753 | 0.807 | 0.743 | 0.695 |
| Shrimp | 0.770 | 0.792 | 0.723 | 0.643 |
| Tengra | 0.780 | 0.732 | 0.724 | 0.622 |

Table 4.3: YOLOv8 Training Results After 20 Epochs

The following table shows the results for all the classes as well as each class. The overall result of mAP50 for all classes is 76.1% which is really good. The Bain class showed the highest mAP50 score of 80.9% whereas the Pabda class showed the lowest among the other classes which is 70.5% and Hilsha showed the highest mAP50-95 which is 76.8% and Tengra showed the lowest mAP50-95 which is 62.2%.

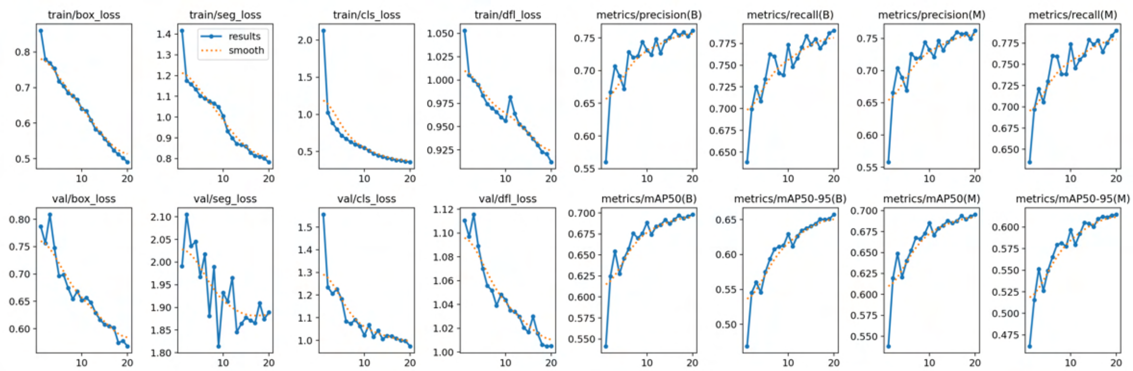


Figure 4.13: Training Graph of YOLOv8

The graphs showcase the training vs loss, validation vs loss and evaluation metrics graph which was run within 20 epochs. In the training vs loss graph, we can see that the loss curve is gradually decreasing close along the smooth line which indicates that the model is learning effectively. The validation vs loss graph also shows that the model is learning well as it is gradually decreasing as well with some fluctuations indicating struggle with unseen data. In the metrics evaluation graphs, we can see that the values are gradually increasing for both precision and recall which indicates that the model is learning effectively and is able to balance correct and complete detections well.

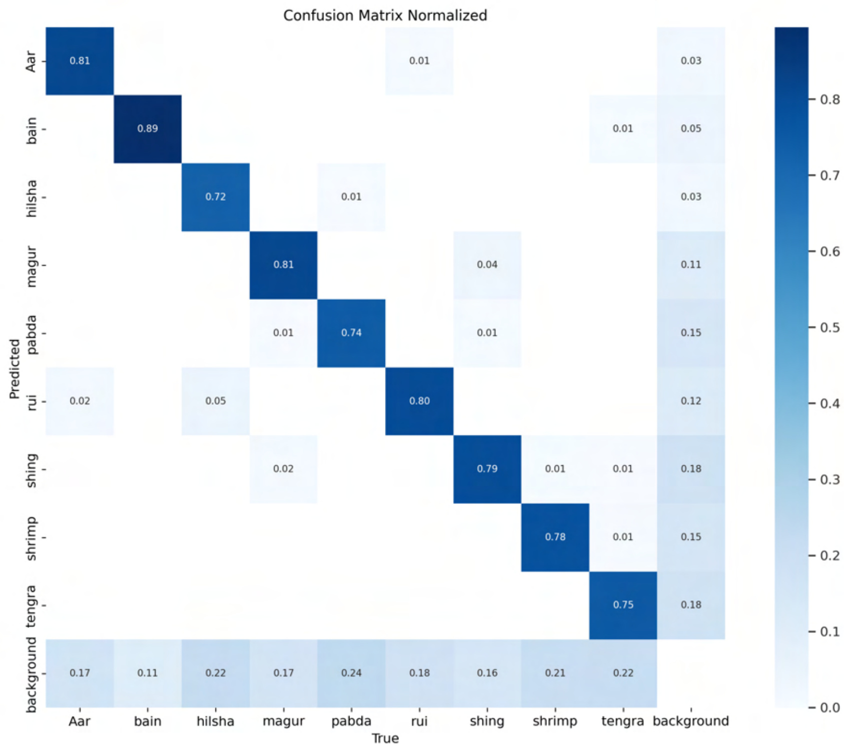


Figure 4.14: Normalized Confusion Matrix of YOLOv8

The figure visualizes the normalized confusion matrix of the YOLOv8 model. The occurrences where the predicted labels match the true labels are represented by the diagonal cells, which go from top left to bottom right. The cell with the higher values indicates better performance and correct prediction. The bain class has the highest value which is 0.89 and the Hilsha class has the lowest value which is 0.72.

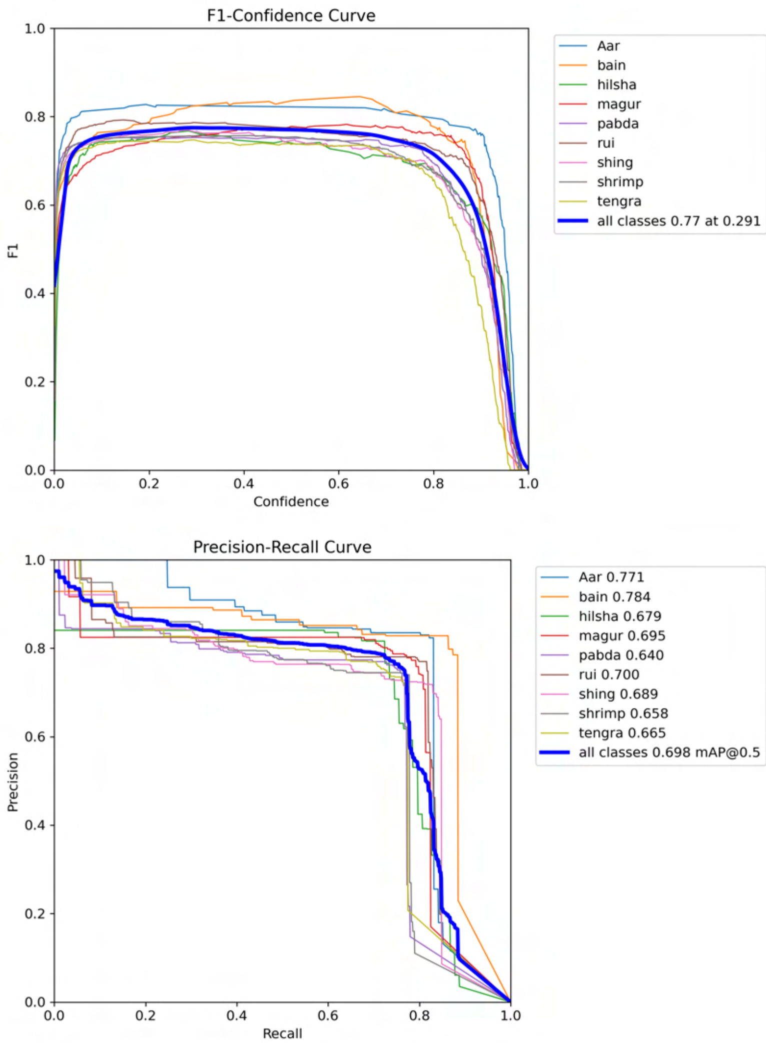


Figure 4.15: F1 Confidence and Precision-Recall Curve of Box

The following graphs contain the F1 confidence and precision-recall for the box. The F1 confidence graph shows a higher curve for Bain class and lower curve for Tengra. The PR curves show how successfully the model separates true and false positives at various recall levels for each class. In this case, the curve shows highest precision for the Bain class and lowest for Pabda and shrimp. The overall class F1 score for all the classes indicated by the blue line.

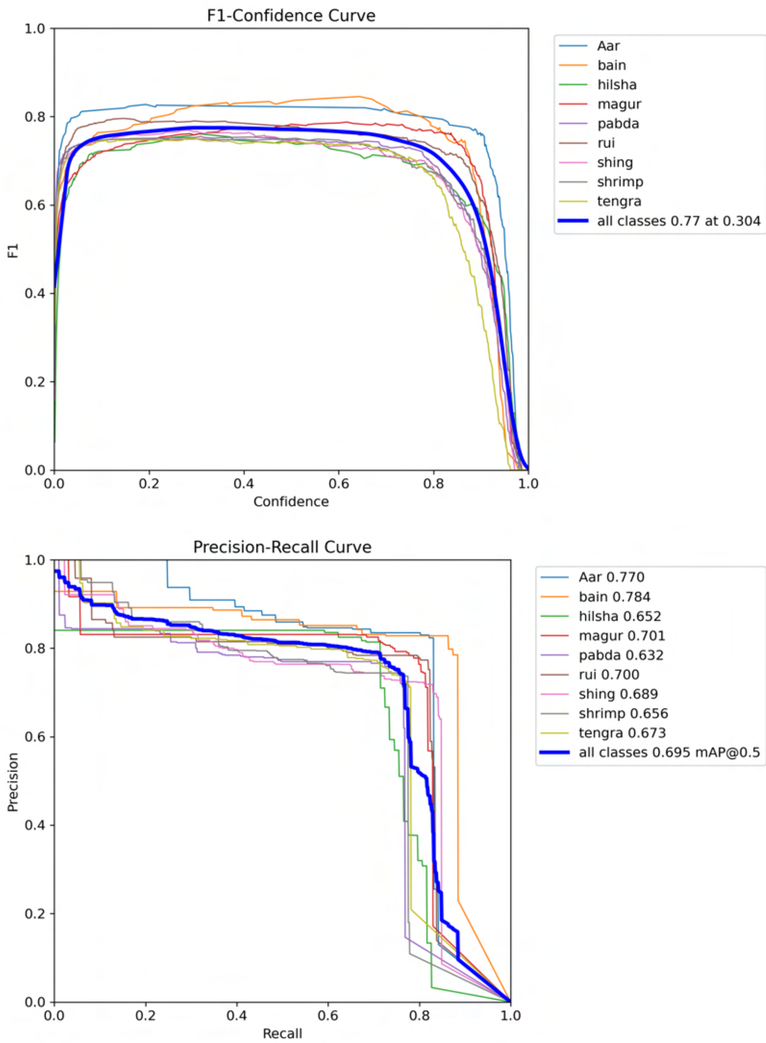


Figure 4.16: F1 Confidence and Precision-Recall Curve of Mask

The Mask F1 Confidence Curve illustrates the F1 score as a function of confidence threshold, with each colored line representing a different fish class and the blue line showing overall model performance which is 0.77 for all classes at a confidence threshold of 0.304. The Mask Precision Recall Curve, plots precision against recall to evaluate how well the model balances false positives and false negatives, with each class represented separately. The peak of each curve indicates the optimal confidence threshold, with the highest AP value for all classes reaching 0.784 which is Bain and lowest Pabda at 0.632. The overall performance is shown in black, and the mean Average Precision value summarizes precision across different recall levels. The higher curves indicate better precision which is Bain and lowest is Tengra recall trade-offs, but some classes perform better than others.

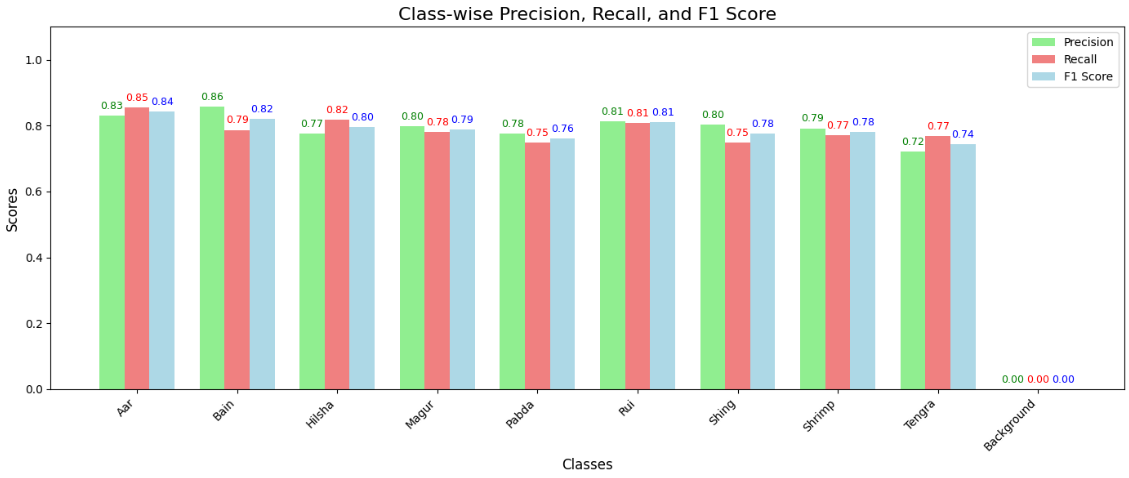


Figure 4.17: YOLOv8 Class-Wise Precision, Recall and F1 Score

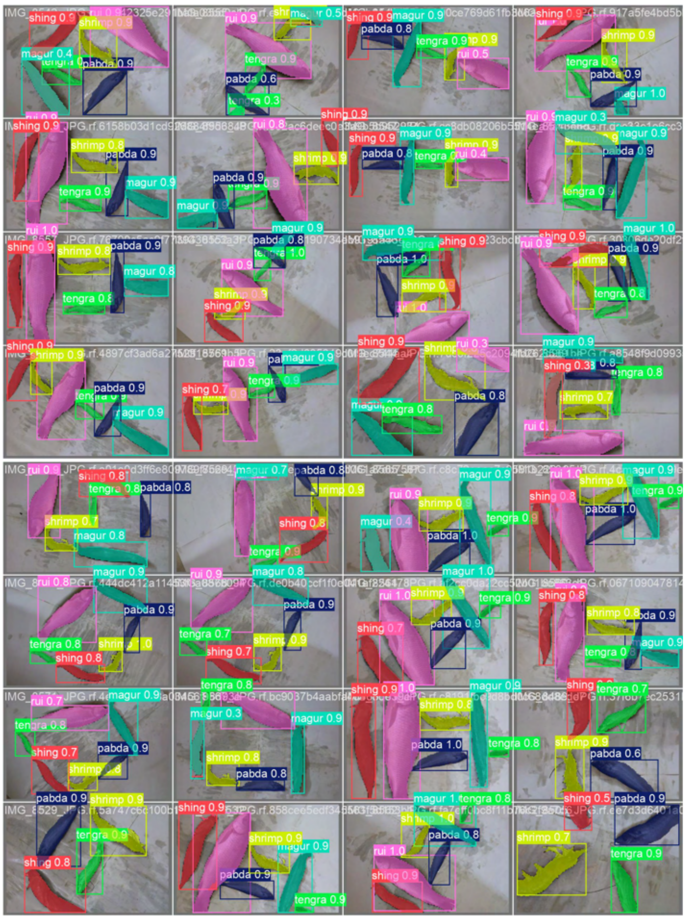

Figure 4.18: YOLOv8 Predicted Non-Overlapping Segmentation

#### 4.2.2.2 YOLOv9

| Class | Box(Precision | Recall | mAP50 | map50-95) |
|---|---|---|---|---|
| All | 0.769 | 0.791 | 0.700 | 0.671 |
| Aar | 0.832 | 0.831 | 0.717 | 0.703 |
| Bain | 0.796 | 0.884 | 0.796 | 0.784 |
| Hilsha | 0.771 | 0.724 | 0.721 | 0.685 |
| Magur | 0.774 | 0.817 | 0.674 | 0.648 |
| Pabda | 0.762 | 0.762 | 0.634 | 0.605 |
| Rui | 0.781 | 0.756 | 0.704 | 0.687 |
| Shing | 0.731 | 0.822 | 0.718 | 0.693 |
| Shrimp | 0.732 | 0.768 | 0.675 | 0.615 |
| Tengra | 0.745 | 0.750 | 0.662 | 0.615 |

Table 4.4: YOLOv9 Training Results After 20 Epochs

The table lists precision, recall, mAP50 and mAP50-95 for all classes individually and also the overall model performance.The overall mAP50 score is 70.0%, indicating strong detection capabilities across all classes.The Bain class scored the highest mAP50 at 79.6%, reflecting the model's strong ability to identify Bain whereas the Pabda class scored the lowest mAP50 at 63.4%, suggesting challenges in distinguishing this class from others.On the other hand, Bain has mAP50-95 of 78.4% and Pabda has the lowest with 60.5%.

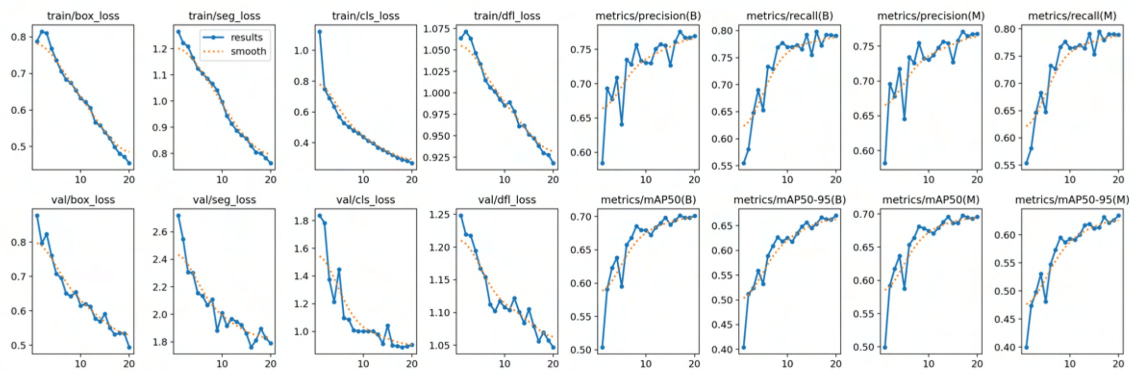


Figure 4.19: Training Graph of YOLOv9

The curve steadily decreases, indicating that the model efficiently minimizes errors on the training data. Moreover, in the final epoch, the loss curve stabilizes at the end and reflects a well converged training process. Similarly, the validation loss decreases, though with some fluctuations in earlier epochs. Moreover,this indicates that the model initially struggles with unseen data but adapts effectively as training progresses thus stabilizing by epoch 20. However, the precision and recall values both exhibit an upward trend. These improvements highlight the model's ability to make accurate predictions while maintaining a high detection rate. The mAP50 metric converges at 0.70 whereas the mAP50-95 stabilizes in between 0.67- 0.68.

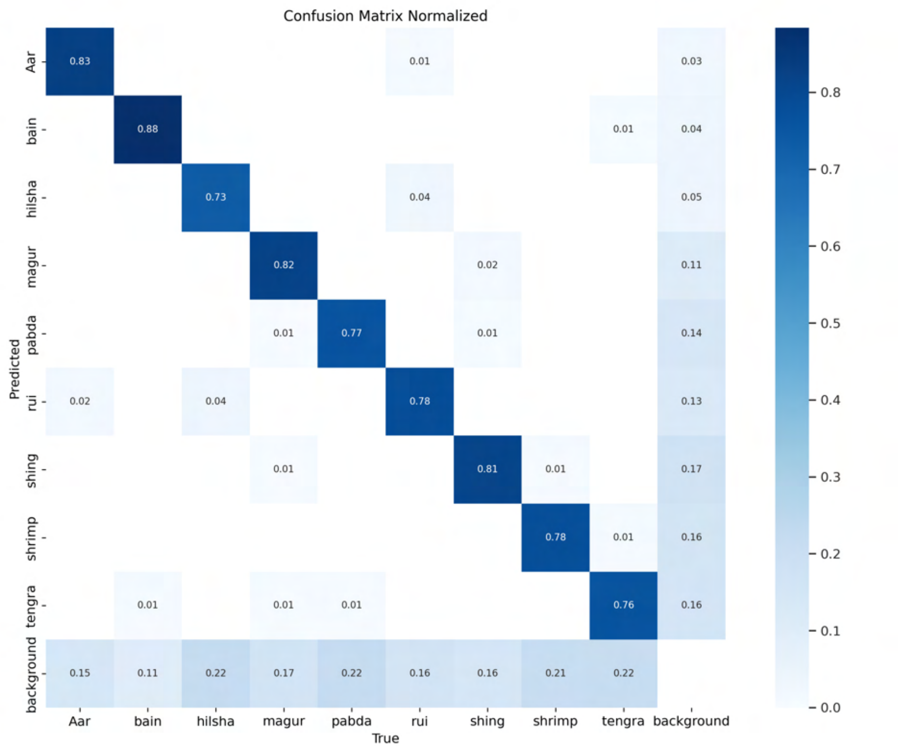


Figure 4.20: Normalized Confusion Matrix of YOLOv9

The confusion matrix illustrates YOLOv9's class predictions for non-overlapping datasets, with diagonal cells showing correct classifications and off-diagonal values indicating misclassifications. Best-performing class is Bain with an accuracy of 0.88. Lowest-performing class is Hilsha with 0.73 accuracy, reflecting some confusion with other visually similar classes. YOLOv9 improves in certain areas compared to YOLOv8 but still faces challenges in distinguishing similar features between certain fish classes. Bain's strong performance demonstrates effective feature recognition.

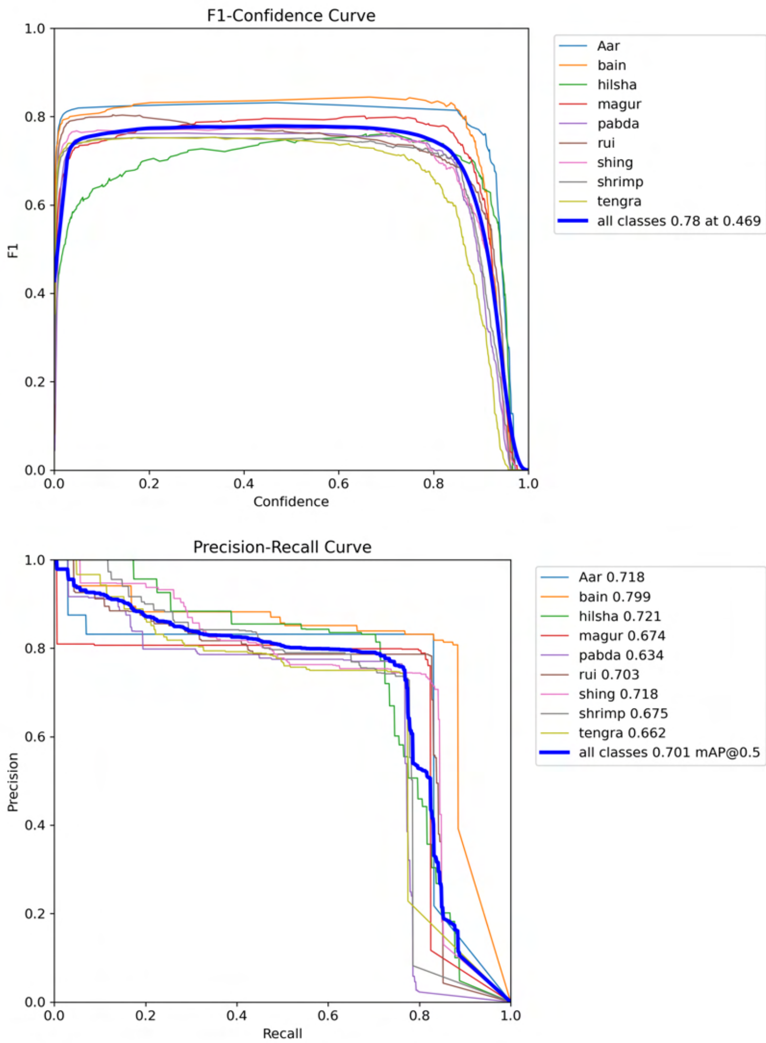


Figure 4.21: F1 Confidence and Precision-Recall Curve of Box

The F1 Curve shows the balance between precision and recall for all classes at different thresholds and the PR Curve examines the tradeoff between precision and recall for true and false positives. Here, all classes shows a F1 score of 0.78 at 0.469 confidence threshold. Morever, the best PR value is Bain class, reflecting its distinct feature representation and the worst PR value is Pabda class, which display a decline in precision as recall increases. While YOLOv9 achieves better performance in handling certain classes like Bain, its weaker PR curve for Tengra suggests difficulties in distinguishing features under varying lighting or angles.

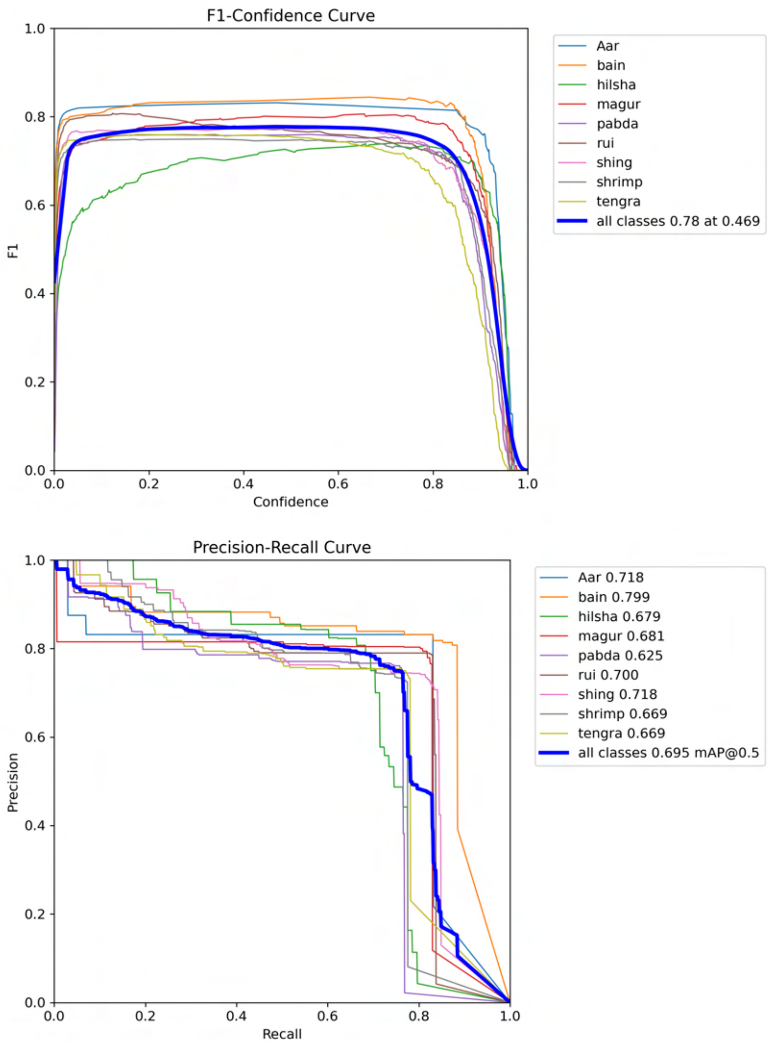


Figure 4.22: F1 Confidence and Precision-Recall Curve of Mask

The curve of F1 Confidence evaluates the balance between precision and recall for segmentation masks across all classes at varying confidence thresholds. Observed for fish classes which indicates strong segmentation performance for these distinct fish types, Bain and Aar, reflecting challenges in accurately segmenting these classes. The precision-recall curve measures how well the model separates true positives from false positives for segmentation masks under varying recall levels achieved by Bain which is 0.799, reflecting its distinct features and effective segmentation across recall levels and observed for Pabda which is 0.625 where precision drops significantly as recall increases, highlighting issues with false positives.

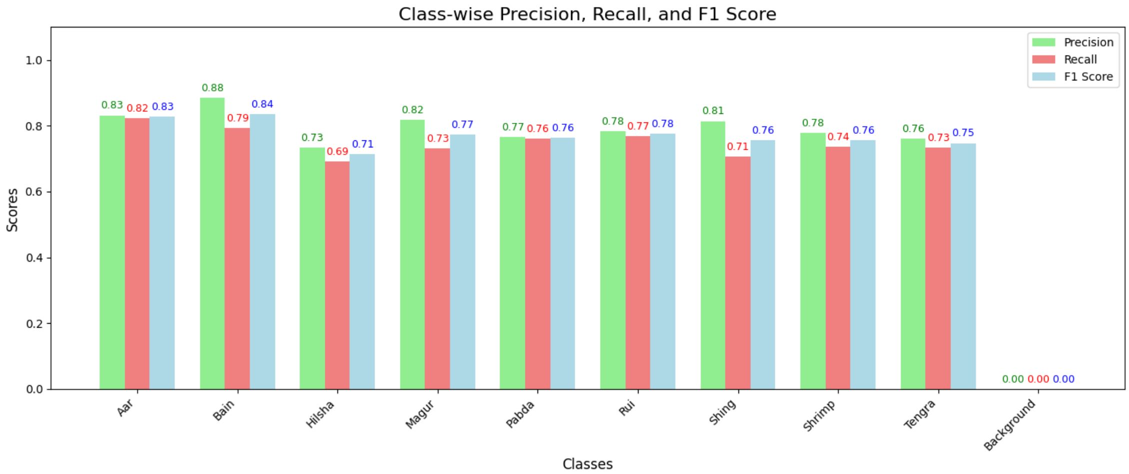


Figure 4.23: YOLOv9 Class-Wise Precision, Recall and F1 Score

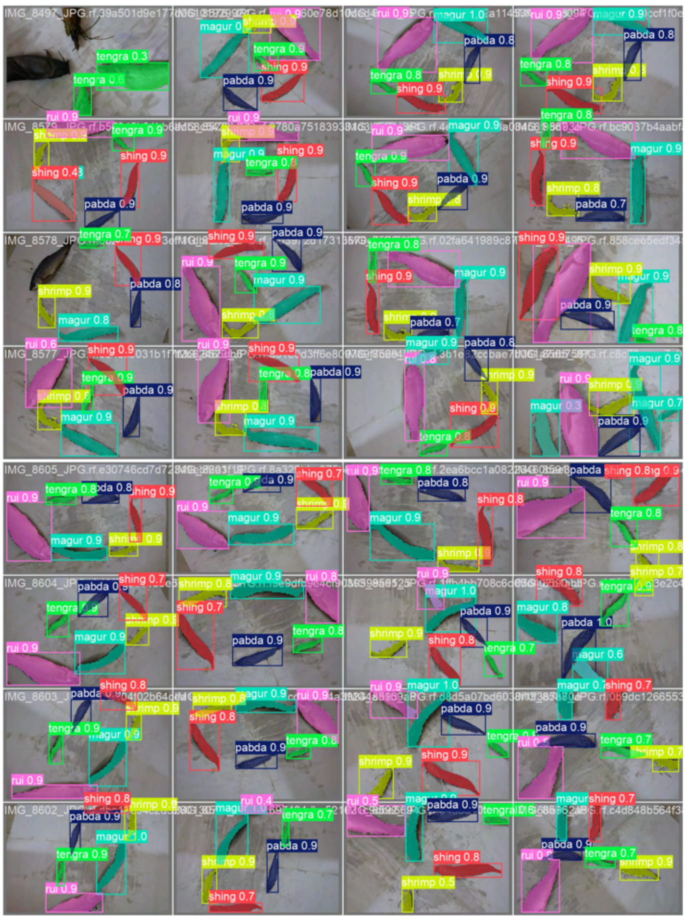

Figure 4.24: YOLOv9 Predicted Non-Overlapping Segmentation

#### 4.2.2.3 YOLOv11

| Class | Box(Precision | Recall | mAP50 | map50-95) |
|---|---|---|---|---|
| All | 0.816 | 0.801 | 0.756 | 0.711 |
| Aar | 0.883 | 0.868 | 0.809 | 0.774 |
| Bain | 0.849 | 0.879 | 0.801 | 0.754 |
| Hilsha | 0.857 | 0.775 | 0.756 | 0.733 |
| Magur | 0.795 | 0.830 | 0.782 | 0.743 |
| Pabda | 0.783 | 0.747 | 0.706 | 0.662 |
| Rui | 0.834 | 0.799 | 0.786 | 0.771 |
| Shing | 0.775 | 0.784 | 0.735 | 0.697 |
| Shrimp | 0.774 | 0.778 | 0.725 | 0.651 |
| Tengra | 0.795 | 0.746 | 0.701 | 0.616 |

Table 4.5: YOLOv11 Training Results After 20 Epochs

The table lists YOLOv11's performance metrics for all fish classes after 20 epochs, highlighting precision, recall, mAP50, and mAP50-95 values. Overall mAP50 score is 75.6%, reflecting robust detection across all classes. Best-performing class is Aar, achieving 80.9% mAP50. Weakest performing class is Tengra, with an mAP50 of 70.1%, indicating challenges in differentiating this class. YOLOv11 demonstrates strong segmentation capabilities across most classes, particularly Aar and Bain, with noticeable improvements compared to YOLOv8 and YOLOv9. However, lower performance for Tengra suggests room for optimization.

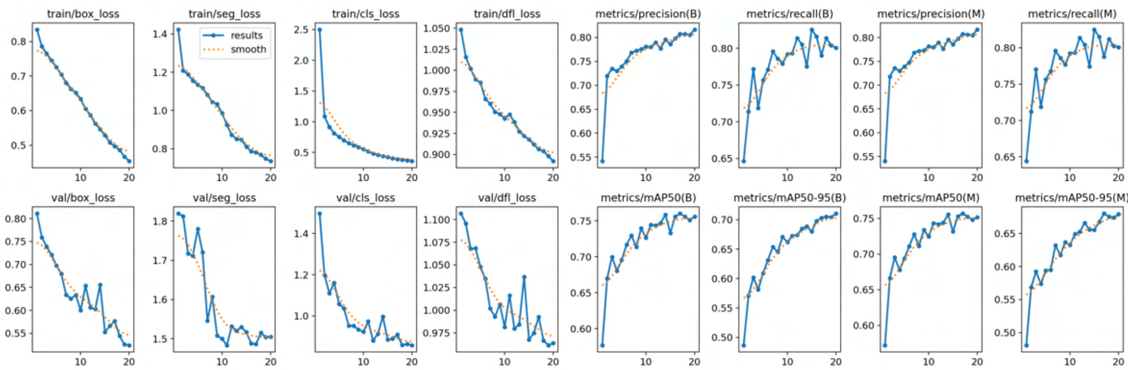


Figure 4.25: Training Graph of YOLOv11

YOLOv11's training and validation trends across 20 epochs, highlighting training loss, validation loss, and evaluation metrics. The graph gradually decreases over the epochs, stabilizing at the end. This steady decline indicates efficient learning and the reduction of prediction errors on the training dataset. Initially higher than training loss, fluctuating slightly in the early epochs. The close alignment of validation loss with training loss reflects that the model generalizes well without overfitting. Moreover, both metrics show consistent improvement throughout the epochs and the final precision is around 0.81 which demonstrates the model's accuracy in detecting fish and final recall is around 0.79 thus highlighting the model's ability to correctly identify positive instances. The mAP50 metric improves steadily and converging to 0.756 which showcases a strong detection performance across all classes. Moreover, mAP50-95 stabilizes approximately in between 0.69-0.71 and indicating strong performance under varying IoU thresholds.

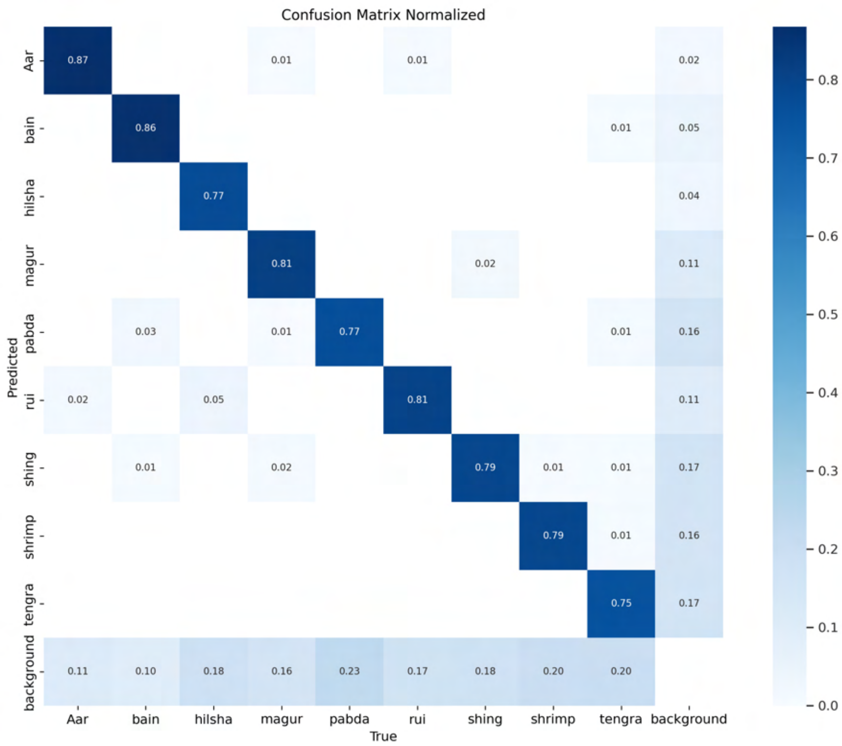


Figure 4.26: Normalized Confusion Matrix of YOLOv11

YOLOv11's segmentation accuracy for each fish class, highlighting both correct and incorrect predictions. The Aar class achieves the highest accuracy at 0.87, indicating robust detection with minimal misclassifications. In contrast, the Tengra class exhibits the lowest accuracy at 0.75. Other classes, such as Bain and Aar maintain strong performance with accuracies above 0.80, showcasing effective feature extraction and differentiation. Misclassifications are more pronounced in visually similar classes, suggesting the need for improved handling of inter class similarities and overlapping segmentation. Overall, the matrix highlights YOLOv11's consistent performance while pointing out specific challenges with certain fish classes.

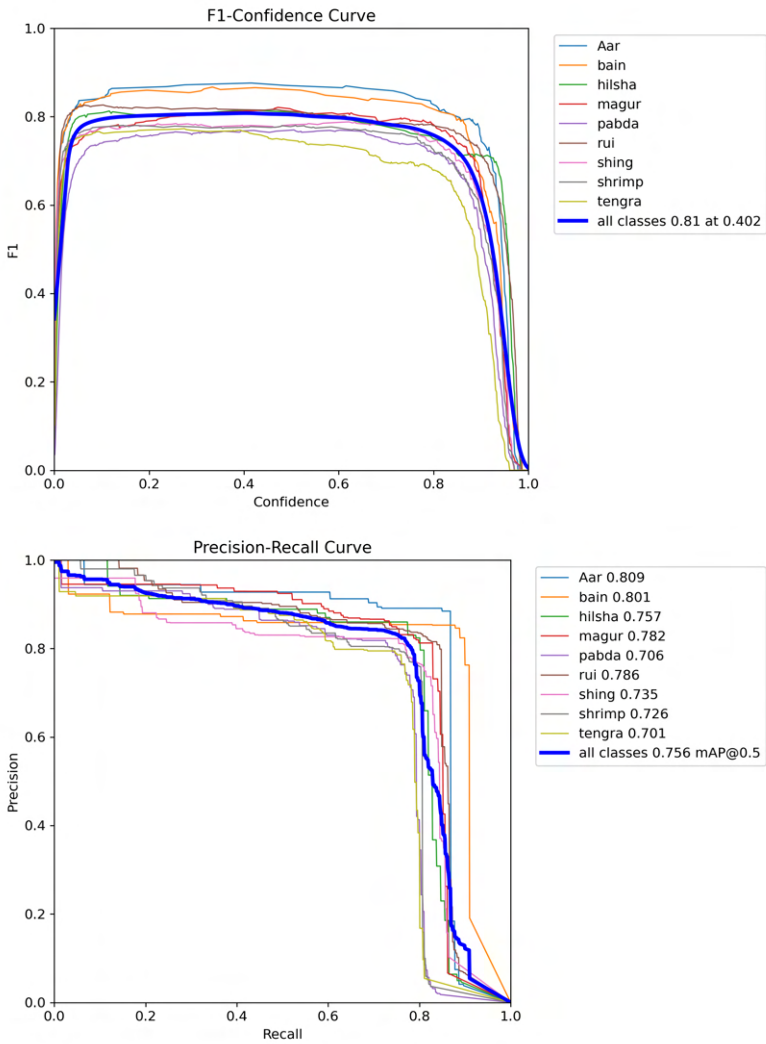


Figure 4.27: F1 Confidence and Precision-Recall Curve of Box

The curve evaluates the precision recall and F1 confidence for bounding boxes. Tracks the tradeoff between precision and recall for true and false positives. Best F1 scores are Aar and Bain, indicating strong detection and lowest are Tengra and Pabda. Lowest PR value is Tengra with 0.701 and highest is Aar with .809. YOLOv11 demonstrates strong performance for most classes, with clear room for improvement in differentiating visually similar classes like Pabda .

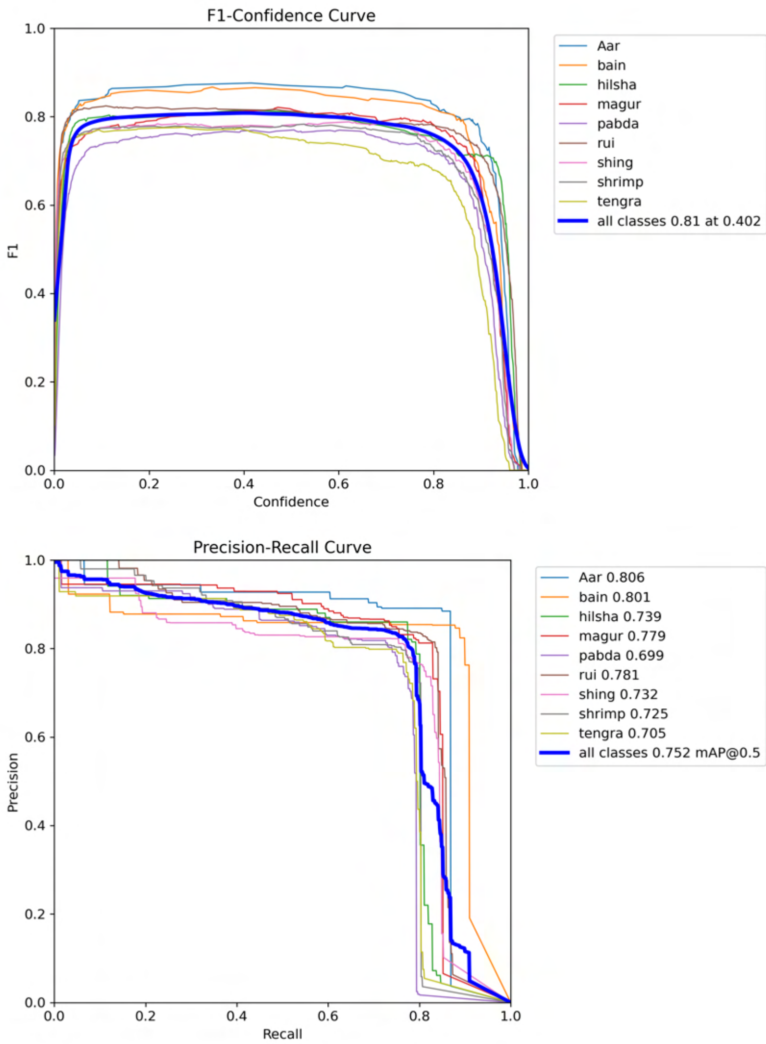


Figure 4.28: F1 Confidence and Precision-Recall Curve of Box

The curve highlights segmentation performance across confidence thresholds and tracks precision versus recall for segmentation masks. Best mask performance is Aar, achieving consistent F1 and PR scores. Weakest performance is Tengra, reflecting difficulties in precise mask segmentation. YOLOv11 achieves competitive segmentation results, with strong mask performance for distinct classes like Shrimp. However, challenges in segmenting Tengra suggest a need for better feature representation.

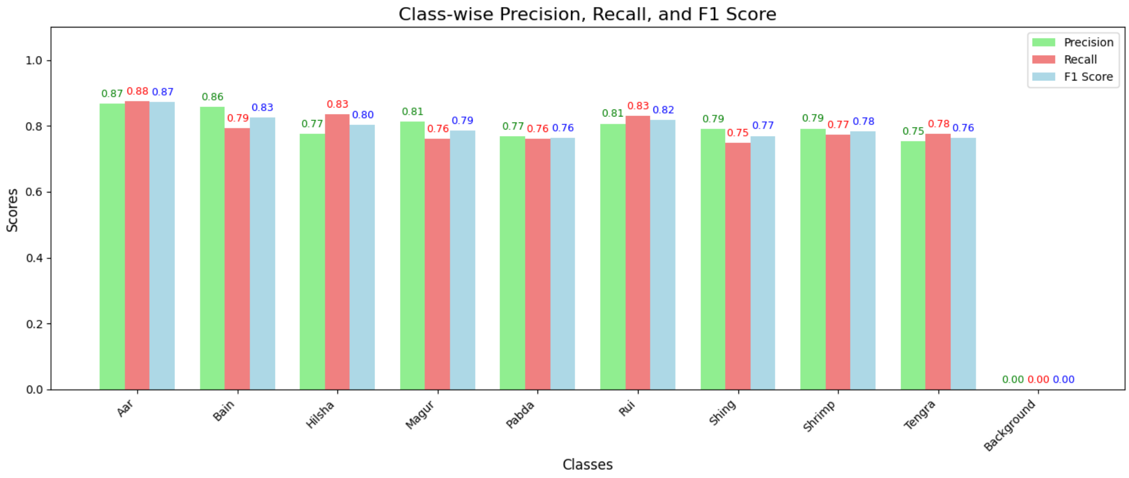


Figure 4.29: YOLOv11 Class-Wise Precision, Recall and F1 Score

Figure 4.30: YOLOv11 Predicted Non-Overlapping Segmentation

### 4.2.3 Overlapping Segmentation Accuracy

#### 4.2.3.1 YOLOv8

| Class | Box(Precision | Recall | mAP50 | map50-95) |
|---|---|---|---|---|
| All | 0.674 | 0.651 | 0.614 | 0.516 |
| Aar | 0.729 | 0.69 | 0.683 | 0.578 |
| Bain | 0.666 | 0.555 | 0.573 | 0.490 |
| Pabda | 0.593 | 0.667 | 0.600 | 0.534 |
| Rui | 0.704 | 0.578 | 0.583 | 0.484 |
| Shing | 0.683 | 0.833 | 0.720 | 0.595 |
| Shrimp | 0.671 | 0.583 | 0.526 | 0.412 |

Table 4.6: YOLOv8 Overlapping Training Results After 20 Epochs

This table shows precision, recall, mAP50 and mAP50-95 for overlapping datasets. Overall mAP50 is 61.4%, reflecting a moderate detection performance. Best class is Shing, achieving an mAP50 of 72.0%, indicating effective identification. However, weakest-performing class is Shrimp with an mAP50 of 52.6%, reflecting challenges in handling this class under overlapping conditions. YOLOv8 shows average performance on overlapping datasets. Shing benefits from distinct features, while Shrimp struggles due to feature overlap.

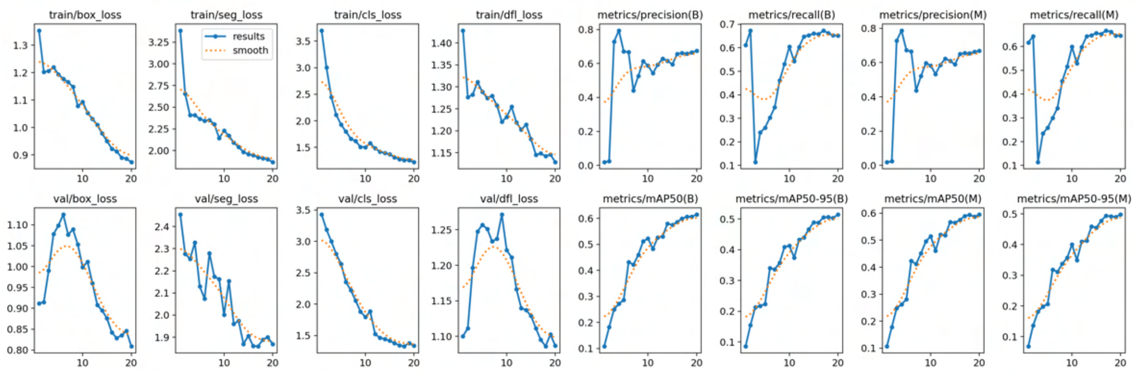


Figure 4.31: Training Graph of YOLOv8 (Overlapping)

The graph illustrates the loss trends for training and validation, along with evaluation metrics like precision and recall over 20 epochs. Training loss decreases consistently, stabilizing at the end thus indicating an effective learning. Validation loss follows a similar trend with some initial fluctuations due to unseen data. The final precision and recall metrics are showing a reasonable detection ability under overlapping conditions. The stable loss curves and gradual improvement in precision and recall highlight that YOLOv8 learns effectively but faces limitations in achieving high accuracy for overlapping instances.

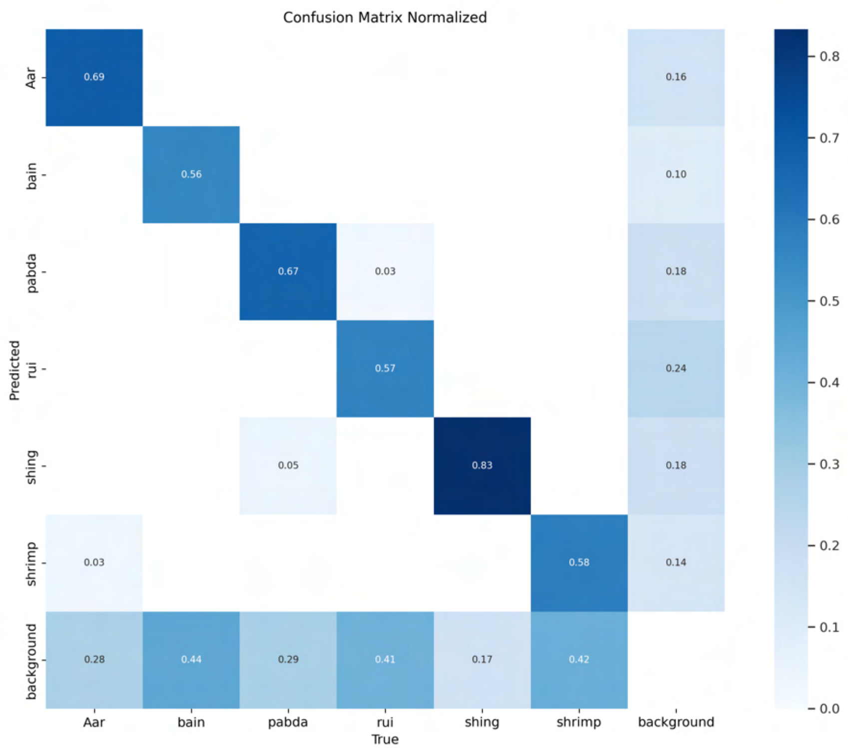


Figure 4.32: Normalized Confusion Matrix of YOLOv8 (Overlapping)

The confusion matrix represents normalized predictions, with diagonal values showing correct classifications and off-diagonal values indicating misclassifications. Shing, with an accuracy of 0.83 showing minimal confusion. Overall accuracy trends are moderate, with most misclassifications occurring between similar classes. YOLOv8 shows strong performance for certain distinct classes like Shing but struggles with overlapping features in classes like Bain thus indicating the need for enhanced feature extraction techniques.

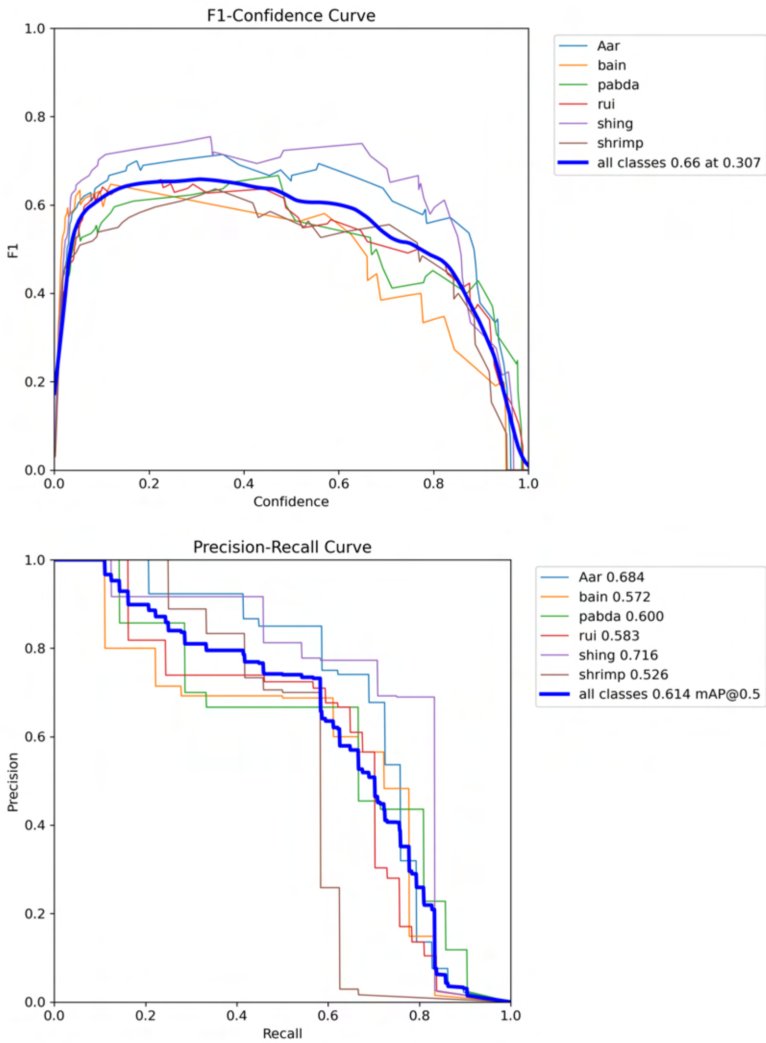


Figure 4.33: F1 Confidence and Precision-Recall Curve of Box

Measures the balance of precision and recall for bounding boxes and tracks the trade-off between precision and recall under varying conditions. Shing is demonstrating strong box segmentation performance. Shrimp reflecting difficulties in maintaining high precision under higher recall values. YOLOv8's box segmentation excels for certain classes but struggles with overlapping scenarios, as evident from lower PR scores for classes like Shrimp but Shing on the other hand shows a strong detection.

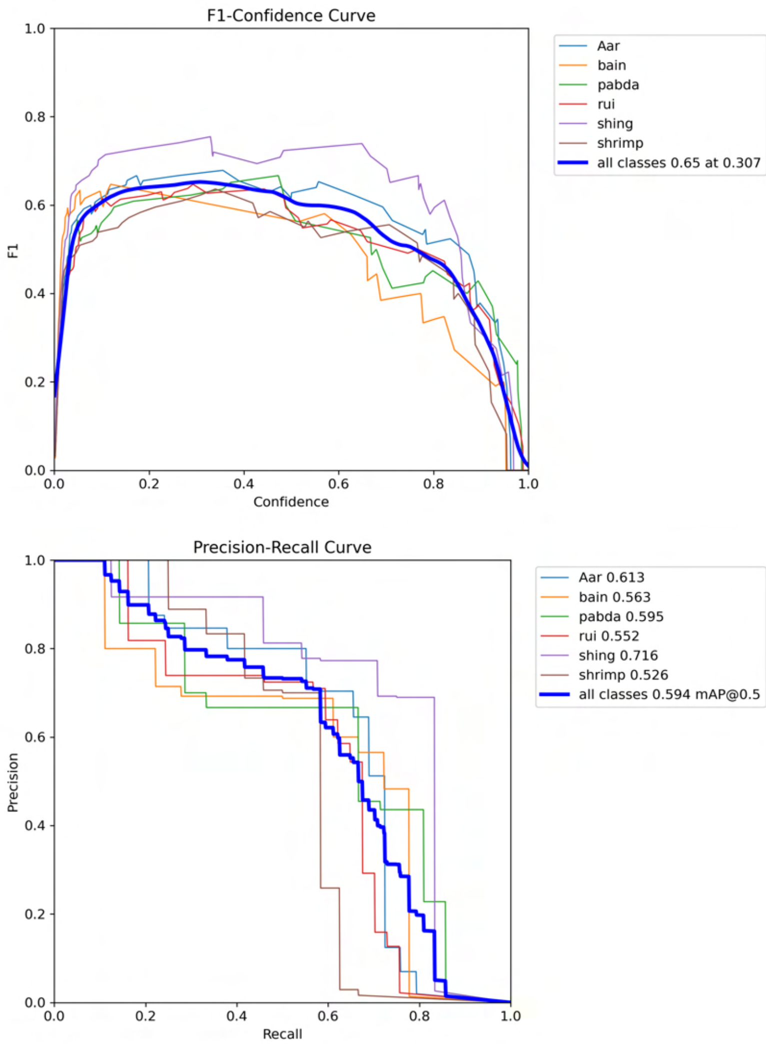


Figure 4.34: F1 Confidence and Precision-Recall Curve of Mask

Highlighting segmentation mask performance for different confidence thresholds and evaluating the tradeoff between precision and recall for segmentation masks. Shing, achieving consistent high F1 and PR values and Shrimp with a declining precision at 0.526. YOLOv8's mask segmentation performs well for distinct classes like Shing but needs improvements for overlapping and challenging classes like Shrimp.

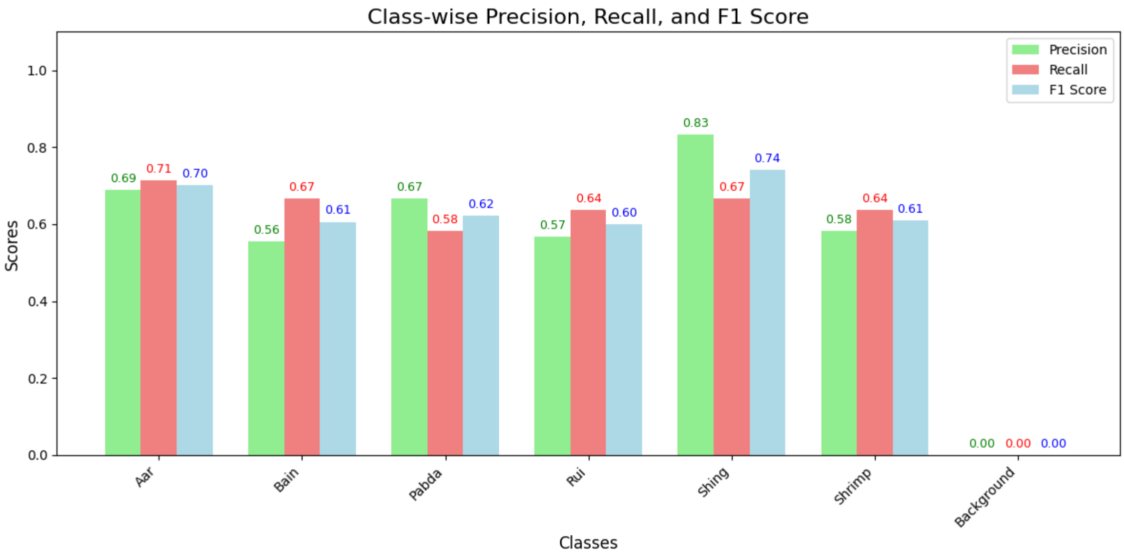


Figure 4.35: YOLOv8 Class-Wise Precision, Recall and F1 Score (Overlapping)

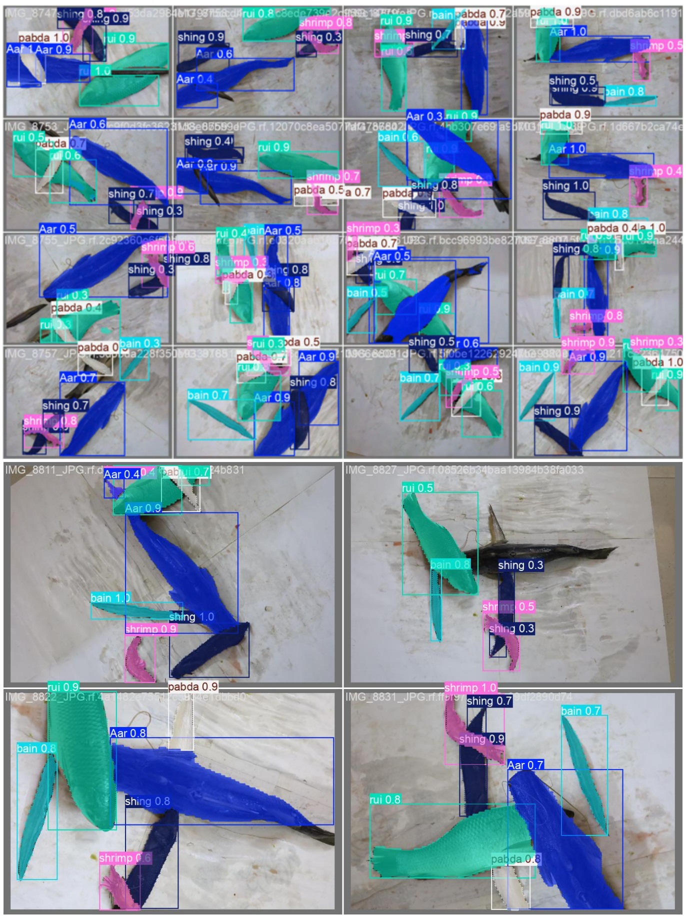

Figure 4.36: YOLOv8 Predicted Overlapping Segmentation

#### 4.2.3.2 YOLOv9

| Class | Box(Precision | Recall | mAP50 | map50-95) |
|---|---|---|---|---|
| All | 0.713 | 0.722 | 0.647 | 0.543 |
| Aar | 0.741 | 0.689 | 0.605 | 0.488 |
| Bain | 0.737 | 0.777 | 0.693 | 0.601 |
| Pabda | 0.722 | 0.810 | 0.730 | 0.627 |
| Rui | 0.697 | 0.683 | 0.549 | 0.443 |
| Shing | 0.705 | 0.792 | 0.683 | 0.607 |
| Shrimp | 0.675 | 0.583 | 0.621 | 0.493 |

Table 4.7: YOLOv9 Overlapping Training Results After 20 Epochs

The table provides precision, recall, mAP50, and mAP50-95 values for overlapping dataset. Overall mAP50: 64.7%, demonstrating moderate detection accuracy for overlapping data. On the other hand, the best-performing class is Pabda, with an mAP50 of 73.0%, highlighting the model's improved segmentation for this class and the weakest-performing class is Rui with an mAP50 of 54.9%, indicating challenges in identifying this class under overlapping conditions. YOLOv9 outperforms YOLOv8 in handling overlapping segmentation, with notable improvements in classes like Pabda. However, certain classes like Rui still exhibit segmentation difficulties.

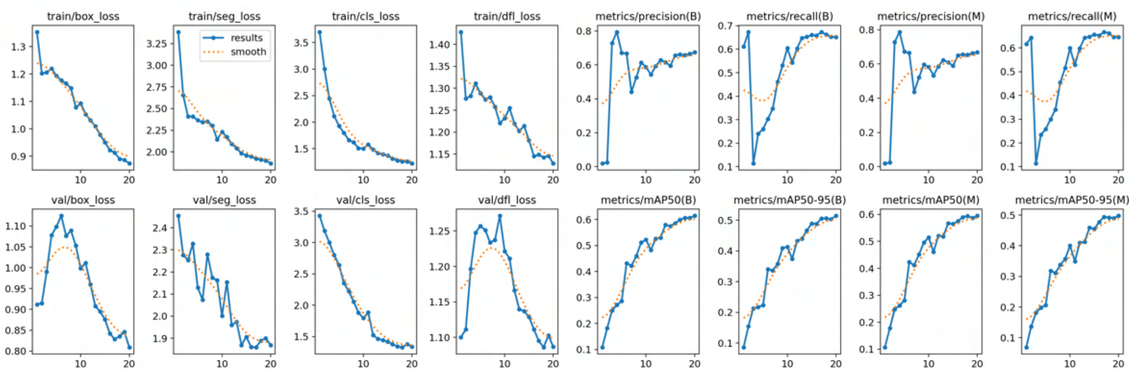


Figure 4.37: Training Graph of YOLOv8 (Overlapping)

The above graph illustrates the loss curves and evaluation metrics across 20 epochs for YOLOv9 on overlapping dataset. However, the training loss decreases steadily, stabilizing at the end by epoch 20 which reflected an efficient learning and the validation loss also fluctuates initially but converges and therefore showing a generalization to unseen data and finally the Precision and Recall shows the Final precision reaches around 0.71, while recall stabilizes approximately at 0.72 which is demonstrating balanced predictions. YOLOv9 shows consistent training convergence and balanced evaluation metrics thus indicating its effectiveness in overlapping segmentation tasks.

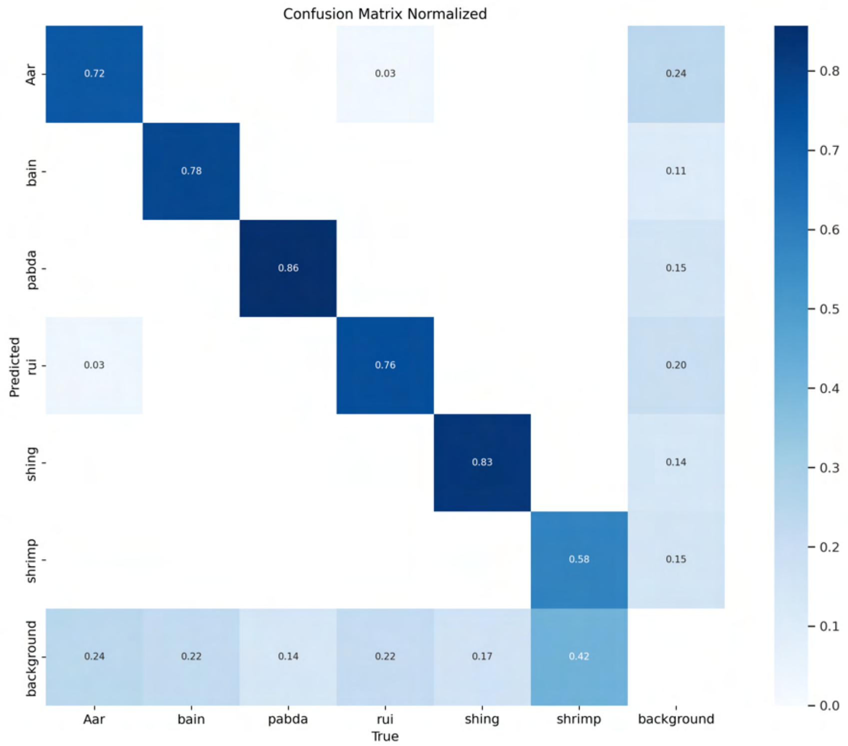


Figure 4.38: Normalized Confusion Matrix of YOLOv9 (Overlapping)

The confusion matrix visualizes normalized predictions for overlapping datasets, showing diagonal values for correct classifications. Moreover, the best-performing class is Pabda, achieving an accuracy of 0.86 showing strong identification and the weakest-performing class is Shrimp with frequent misclassifications. YOLOv9 demonstrates improved segmentation for Pabda, but misclassifications in overlapping scenarios highlight challenges with smaller and visually similar classes like Shrimp

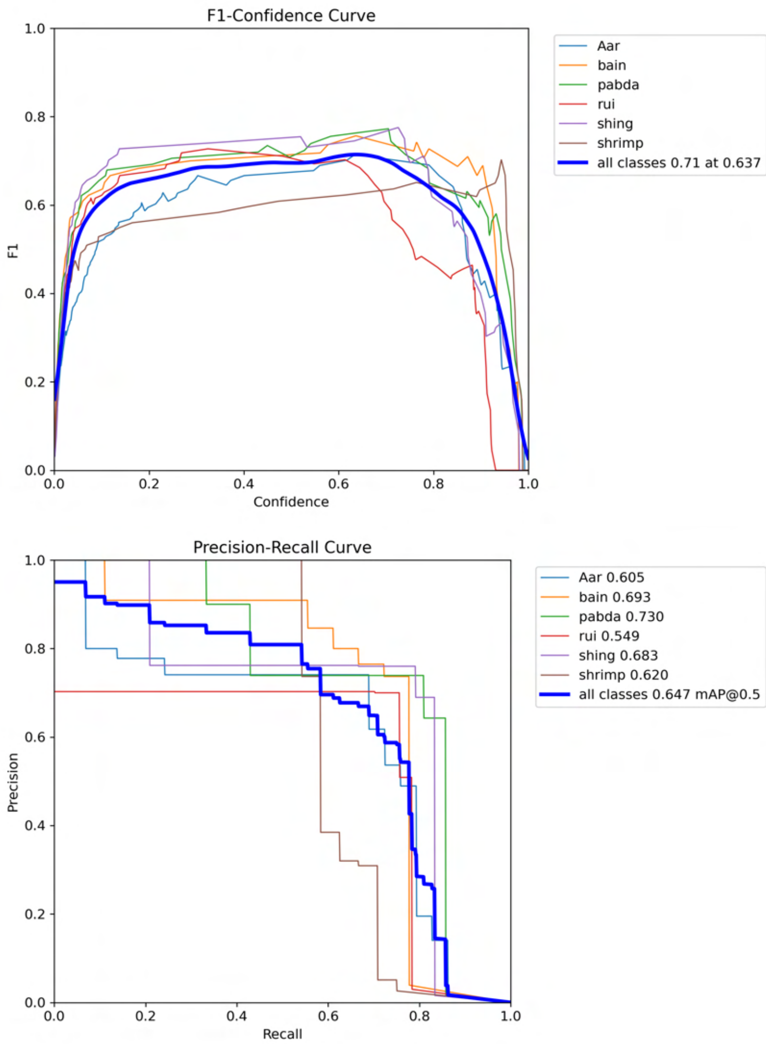


Figure 4.39: F1 Confidence and Precision-Recall Curve of Box

The above graphs shows the balance between precision and recall across confidence thresholds. Evaluates true versus false positives under varying recall. Moreover, the highest F1 and PR values of Pabda is reflecting a strong detection whereas the weakest PR scores of Rui is declining precision at higher recall levels shows that the strong box segmentation performance for Pabda suggests refined feature extraction, while lower PR scores for Rui indicate overlapping segmentation challenges.

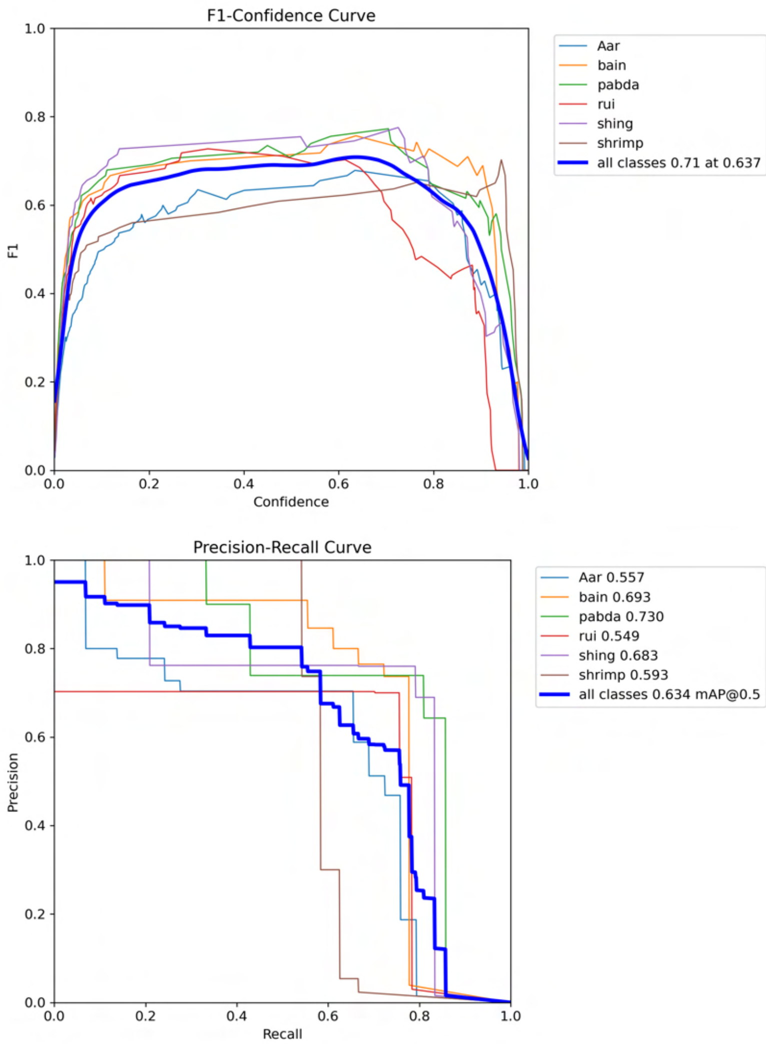


Figure 4.40: F1 Confidence and Precision-Recall Curve of Mask

Highlights segmentation performance under varying confidence thresholds. Analyzes precision recall tradeoff for segmentation masks. Moreover, highest F1 and PR values of Pabda, with high F1 and PR values whereas the weakest F1 and PR scores of Rui is reflecting difficulties in segmentation accuracy. YOLOv9 shows consistent improvements in mask segmentation for Pabda but overlapping instances for Rui require further enhancements.

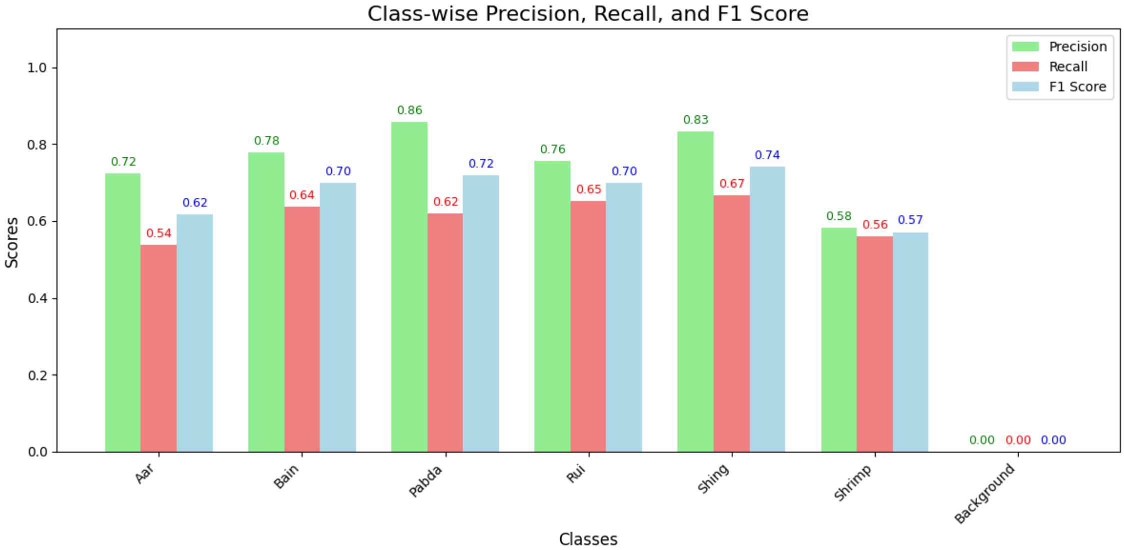


Figure 4.41: YOLOv9 Class-Wise Precision, Recall and F1 Score (Overlapping)

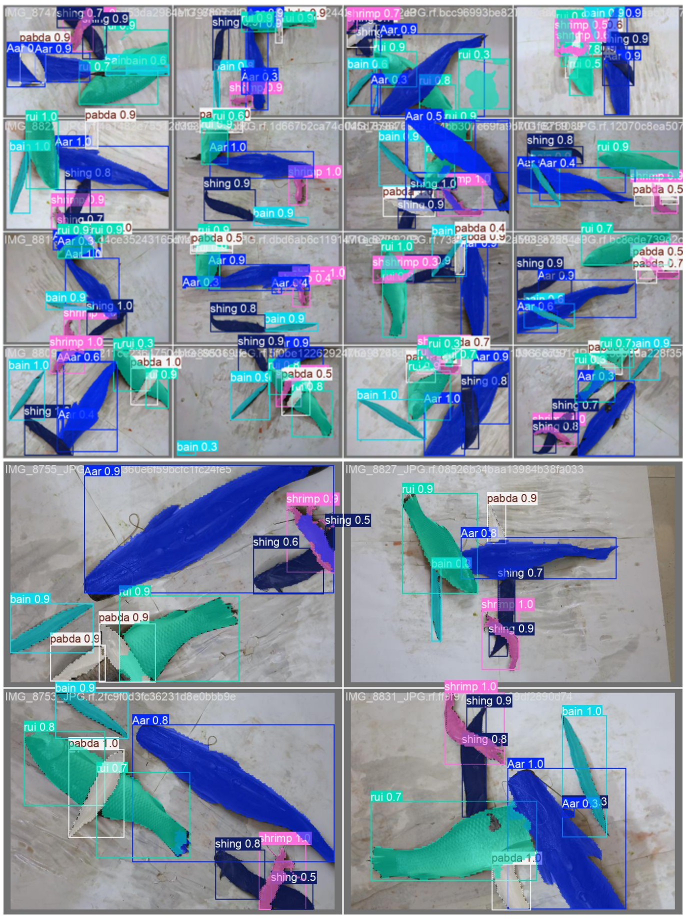

Figure 4.42: YOLOv9 Predicted Overlapping Segmentation

#### 4.2.3.3 YOLOv11

| Class | Box(Precision | Recall | mAP50 | map50-95) |
|---|---|---|---|---|
| All | 0.702 | 0.694 | 0.629 | 0.548 |
| Aar | 0.733 | 0.69 | 0.661 | 0.598 |
| Bain | 0.710 | 0.667 | 0.645 | 0.560 |
| Pabda | 0.706 | 0.799 | 0.674 | 0.572 |
| Rui | 0.666 | 0.593 | 0.566 | 0.481 |
| Shing | 0.732 | 0.833 | 0.714 | 0.673 |
| Shrimp | 0.664 | 0.583 | 0.512 | 0.405 |

Table 4.8: YOLOv11 Overlapping Training Results After 20 Epochs

The table outlines the precision, recall, mAP50 and mAP50-95 for all classes under overlapping conditions. In addition, the overall mAP50: 62.9%, indicating moderate detection capabilities for overlapping dataset whereas the best-performing class is Shing, with an mAP50 of 71.4%, showcasing effective segmentation and the weakest performing class is Shrimp, with an mAP50 of 51.2%, highlighting segmentation challenges. YOLOv11 demonstrates consistent segmentation for distinct classes such as Shing but faces difficulties with classes like Shrimp, which might require better feature separation techniques.

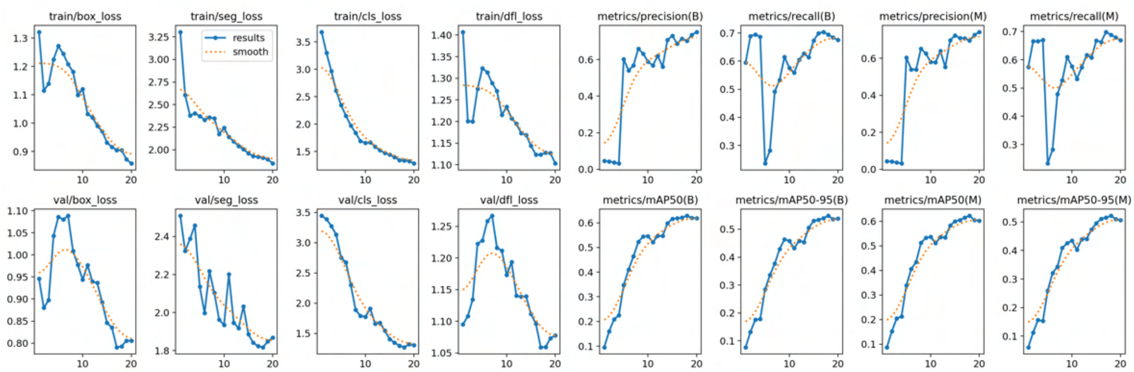


Figure 4.43: Training Graph of YOLOv11 (Overlapping)

The training loss graph steadily decreases, stabilizing at the end as well by epoch 20, reflecting effective optimization, and the validation loss graph converges and indicating a good generalization to unseen data. The graph also reflects the model's ability to maintain consistent training performance with balanced detection metrics, making it suitable for moderately complex overlapping datasets.

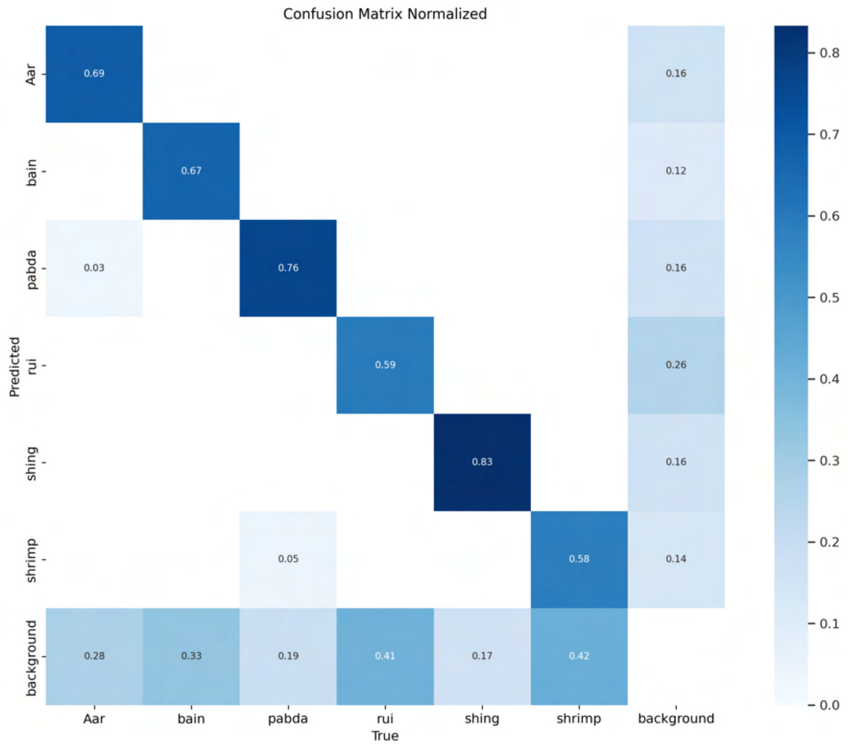


Figure 4.44: Normalized Confusion Matrix of YOLOv11 (Overlapping)

The confusion matrix visualizes normalized predictions, highlighting correct classifications along the diagonal. Best-performing class is Shing, with an accuracy of 0.83, showcasing strong segmentation. Weakest performing class is Shrimp, with frequent misclassifications as Aar. YOLOv11 achieves accurate segmentation for well-defined classes like Shing but struggles with overlapping or visually similar classes like Shrimp, indicating areas for further improvement.

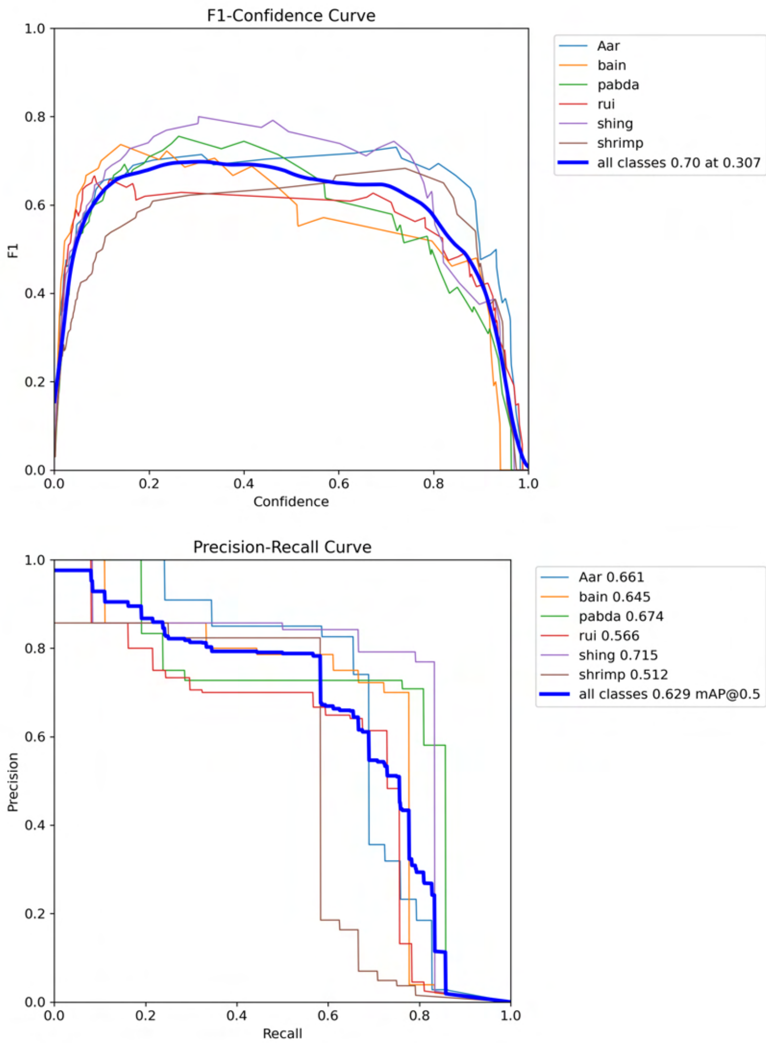


Figure 4.45: F1 Confidence and Precision-Recall Curve of Box

The box F1 graph evaluates the balance between precision and recall for bounding boxes under varying confidence thresholds. The box PR graph assesses precision versus recall trade offs. Highest F1 and PR values is Shing which is indicating effective box segmentation. Lowest PR values is Shrimp as due to significant precision drops at higher recall levels. YOLOv11 excels in box segmentation for distinct classes but requires improved feature extraction for overlapping scenarios like Shrimp.

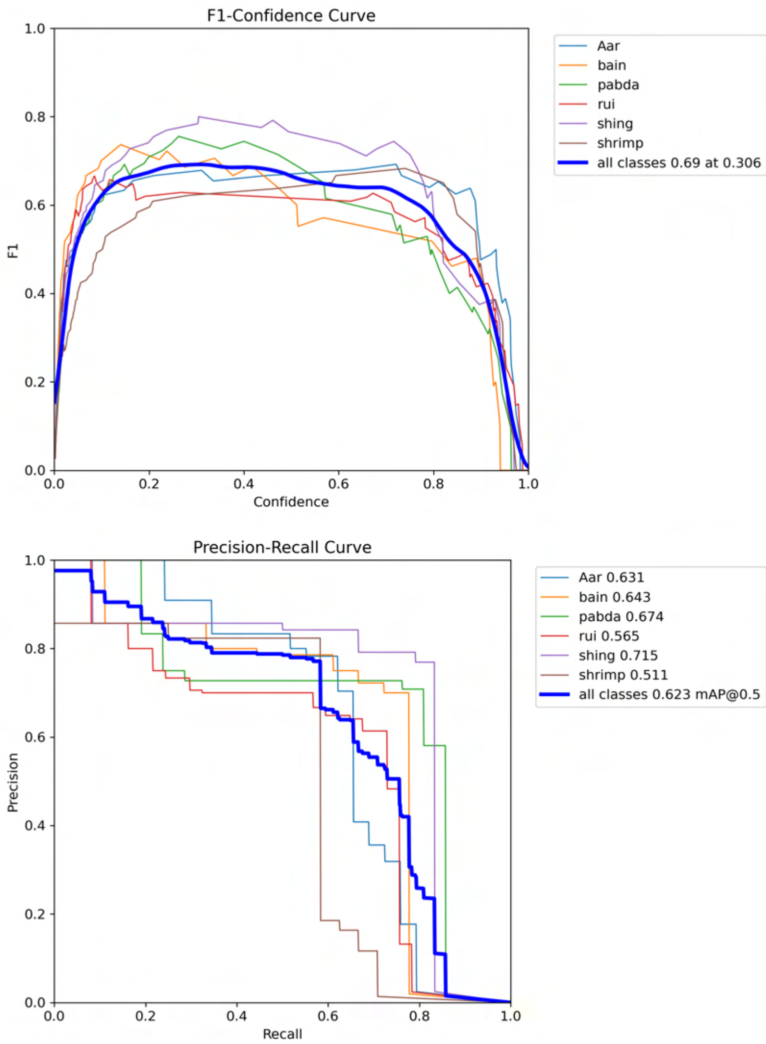


Figure 4.46: F1 Confidence and Precision-Recall Curve of Mask

The Mask F1 Curve reflects segmentation mask performance across confidence thresholds and the Mask PR Curve highlights the model's precision-recall tradeoff for segmentation masks. Best mask performance is Shing, with consistent high F1 and PR values. Weakest mask performance is Shrimp where precision significantly drops at higher recall. YOLOv11 demonstrates strong segmentation for specific classes like Shing, but overlapping instances for challenging classes like Shrimp require enhanced mask segmentation techniques.

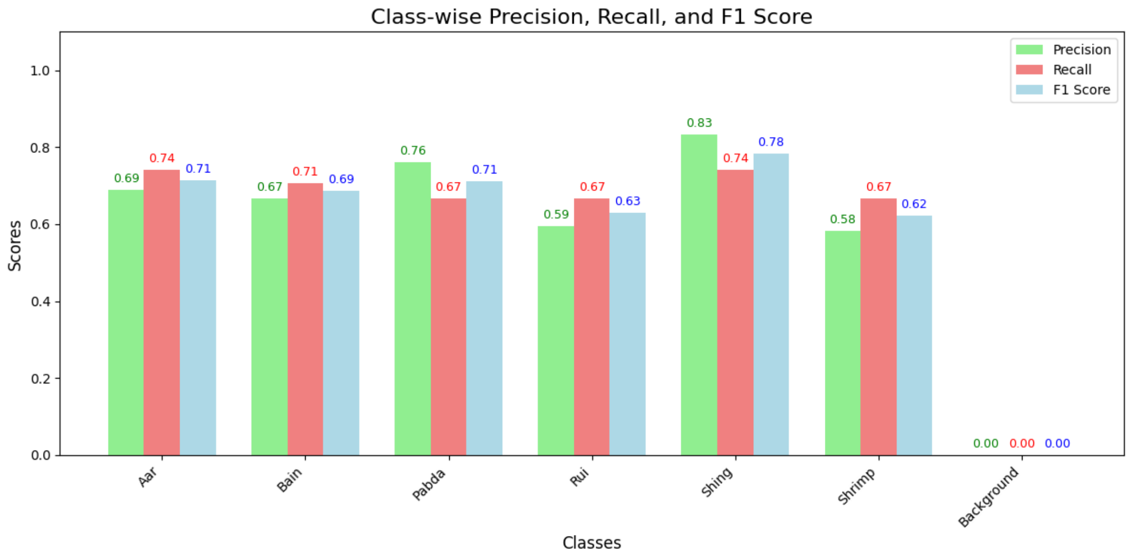


Figure 4.47: YOLOv11 Class-Wise Precision, Recall and F1 Score (Overlapping)

Figure 4.48: YOLOv11 Predicted Overlapping Segmentation

## 4.3 Accuracy Comparison

### 4.3.1 Classification

The two bar charts display the train and test accuracy comparisons across four models: VGG-16, ResNet-50, Basic CNN, and Inception-v3. In addition, in the train accuracy bar chart, Inception-v3 shows the highest performance with 99.44%, followed by the second highest VGG16 which is 98.32%, then Basic CNN 93.17%, and lastly ResNet-50 67.82%. Moreover, the indication shows that Inception-v3 and VGG-16 models excel in learning from the training data, with Basic CNN closely behind while ResNet-50 shows relatively lower performance in training accuracy. In contrast, the test accuracy chart shows that Inception-v3 again leads with 99.69%, followed closely by VGG-16 at 99.40% and Basic CNN at 97.79%, while ResNet-50 still lags at 73%. Our test output demonstrate that Inception-v3 and VGG-16 execute well on unseen data, maintaining high generalization capabilities. Basic CNN also performs comparably, while ResNet-50 continues to show lower test accuracy. The close alignment of training and test accuracy for Inception-v3, VGG-16, and Basic CNN suggests that these models generalize well, while ResNet-50 may be underperforming due to potential issues such as overfitting or suboptimal tuning on the dataset.

| Model Name | Testing Accuracy | Training Accuracy |
|---|---|---|
| Inception-v3 | 99.69% | 99.44% |
| VGG-16 | 99.40% | 98.32% |
| Basic CNN | 97.79% | 93.17% |
| ResNet-50 | 73% | 67.82% |

Table 4.9: Model Performance Comparison

Based on the experimental results displayed in the bar chart, we may come to a conclusion that the Inception-v3 model performs better than other previously trained CNN models. The figure shows a sample bar graph of CNN model accuracy. The accuracy of the models is displayed in the bar chart below.

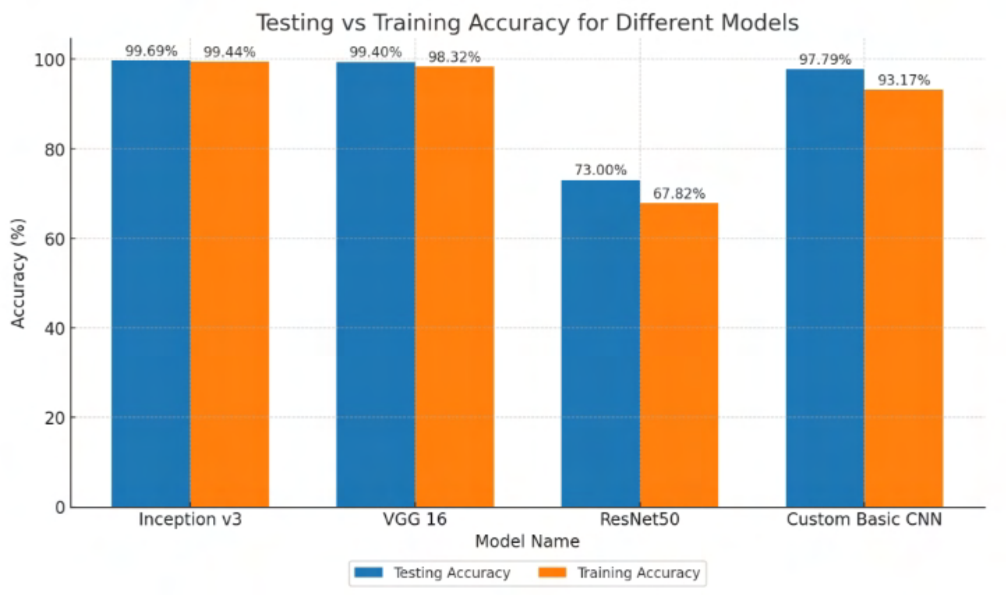


Figure 4.49: Testing Vs Training Accuracy Graph For Different Models

### 4.3.2 Non-Overlapping Segmentation

YOLOv8 achieved an overall mAP50 76.1% accuracy, which is the highest accuracy here. YOLOv11 is slightly lower than YOLOv8 which is 75.6%. In mAP50 we know the IOU threshold is set in 50 which is less trustworthy in terms of accuracy. So, mAP50-95 shows the most reliable accuracy in segmentation where IOU threshold is set between 50-95. Here we see, YOLOv11 shows the highest accuracy which is 71.1%. The reason behind the best accuracy of YOLOv11 is due to it's architectural pattern, augmentation and feature extraction technique. But the accuracy of YOLOv9 is less than both YOLOv8 and YOLOv11. The reasons could be due to not being anchor free, backbone is optimised CSPDarknet with ELAN+, higher training time but less accuracy which means optimization process is less efficient.

| Model Name | mAP50 | mAP50-95 |
|---|---|---|
| YOLOv8 | 76.10% | 70.90% |
| YOLOv9 | 70.00% | 67.10% |
| YOLOv11 | 75.60% | 71.10% |

Table 4.10: Non-Overlapping Segmentation Model Performance Comparison

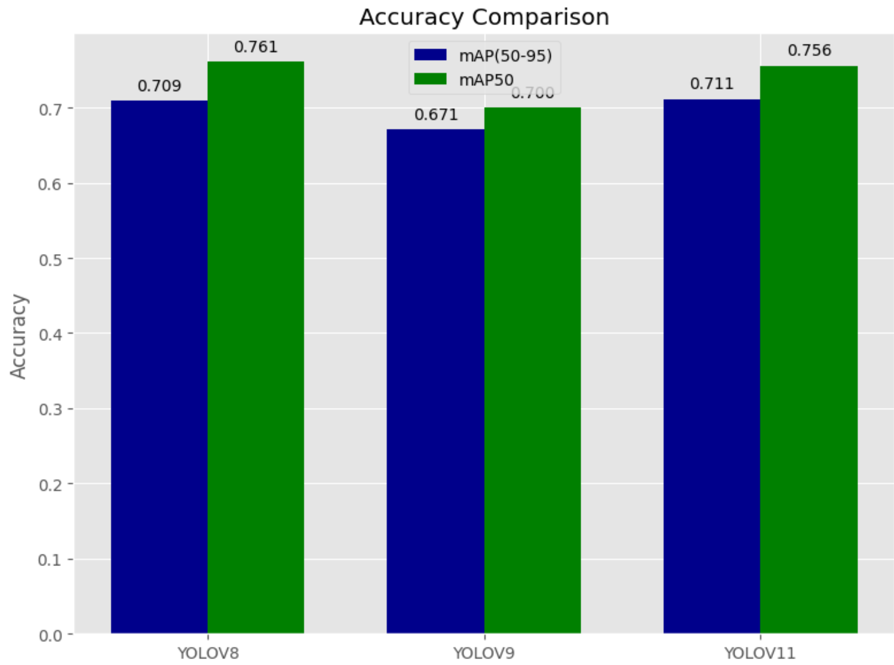


Figure 4.50: Model Accuracy Comparison (Non-Overlapping)

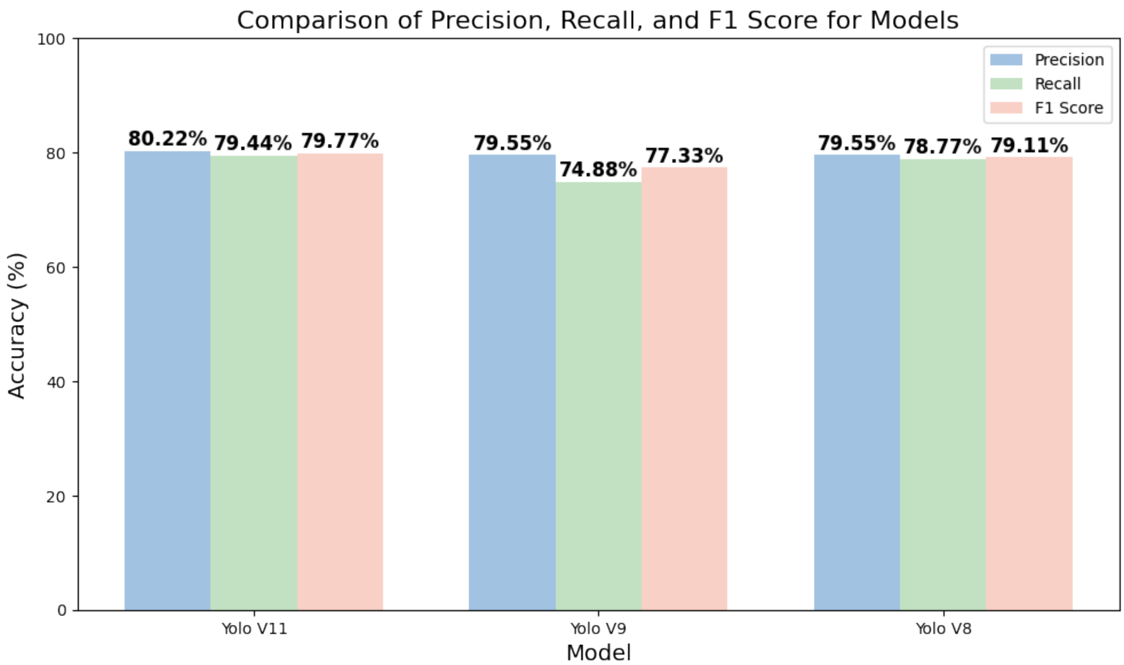


Figure 4.51: Precision-Recall and F1 Score Model Comparison (Non-Overlapping)

As per mAP score of all three models, the precision, recall & F1 score is also showing similar data. Accuracy of YOLOv11 and YOLOv8 are very close where the accuracy of YOLOv11 is the best whereas the accuracy of YOLOv9 isn't better compared to other two models.

### 4.3.3 Overlapping Segmentation

YOLOv9 achieved an overall mAP50 64.7% accuracy, which is the best accuracy here. Here we see, YOLOv11 shows the most accuracy which is 54.8%. For showing the best accuracy of YOLOv11, there are many reasons like it's architectural pattern, augmentation and feature extraction technique which is better than all other models. But the accuracy of YOLOv8 is less than YOLOv11 and YOLOv9. The reasons could be due to YOLOv8 being designed for balanced speed and accuracy but it is struggling in highly dense overlapping scenarios. So, YOLOv8 might lose accuracy for efficiency in complex scenarios.

| Model Name | mAP50 | mAP50-95 |
|---|---|---|
| YOLOv8 | 61.4% | 51.6% |
| YOLOv9 | 64.7% | 54.3% |
| YOLOv11 | 62.9% | 54.8% |

Table 4.11: Overlapping Segmentation Model Performance Comparison

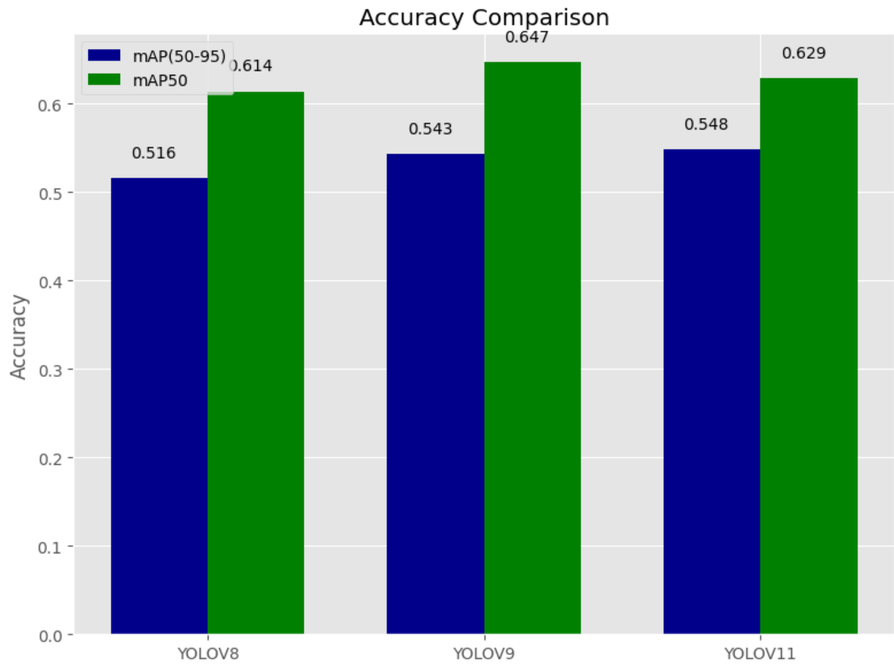


Figure 4.52: Model Accuracy Comparison (Overlapping)

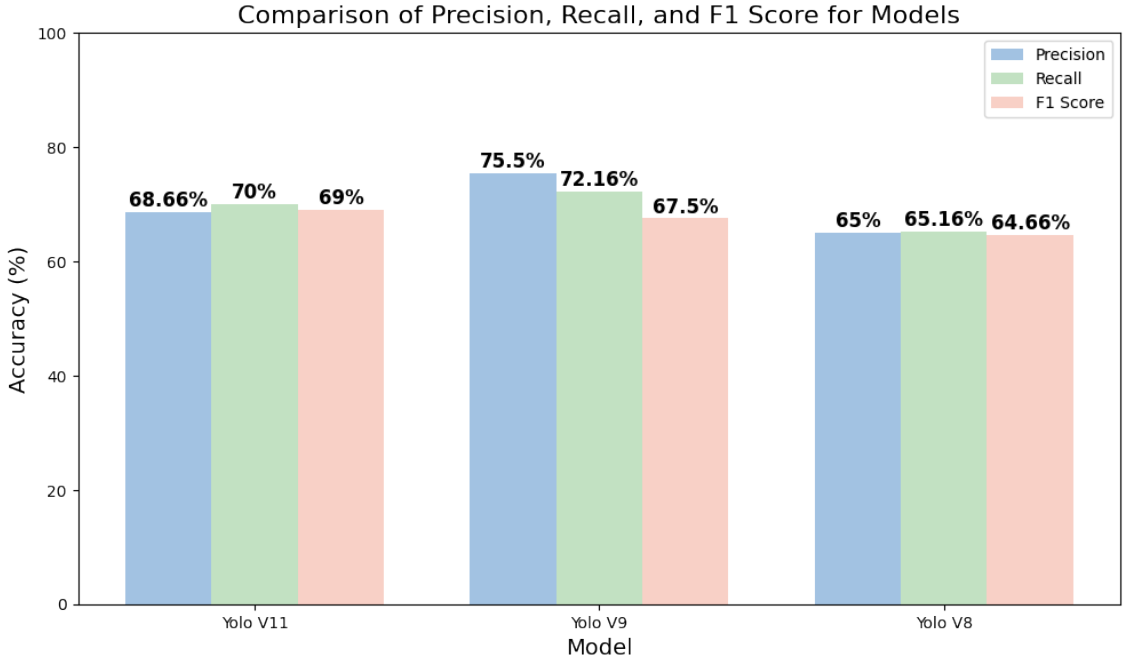


Figure 4.53: Precision-Recall and F1 Score Model Comparison (Overlapping)

In this graph accuracy of YOLOv9 is higher than the other two models which means YOLOv9 is making fewer false detections, strongly detecting without to many false positives. But as the main goal is segmentation so YOLOv11 is performing better in terms of mAP50-95.

### 4.3.4 Class-Wise Accuracy After Segmentation

#### 4.3.4.1 Non-Overlapping

YOLOv11 showed a strong performance for class wise accuracy comparison of 79.0%, and YOLOv8 showed the second best graph result for class wise accuracy of 78.0%. However, YOLOv9 struggled with a lower class-wise accuracy of 74.0%.

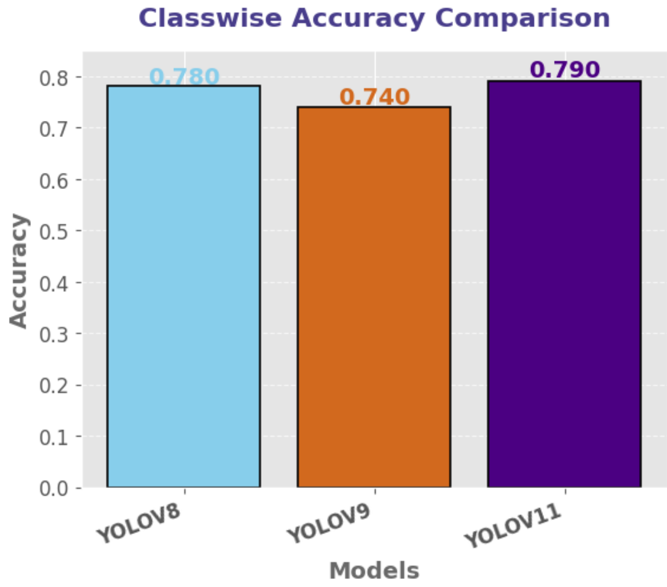


Figure 4.54: Class-Wise Model Accuracy Comparison (Non-Overlapping)

#### 4.3.4.2 Overlapping

YOLOv11 showed an impressive performance for class wise accuracy comparison of 70.0%, and YOLOv8 showed the second best graph result for class wise accuracy comparison of 65.0%. However, YOLOv9 struggled with a lower class-wise accuracy of 61.0%.

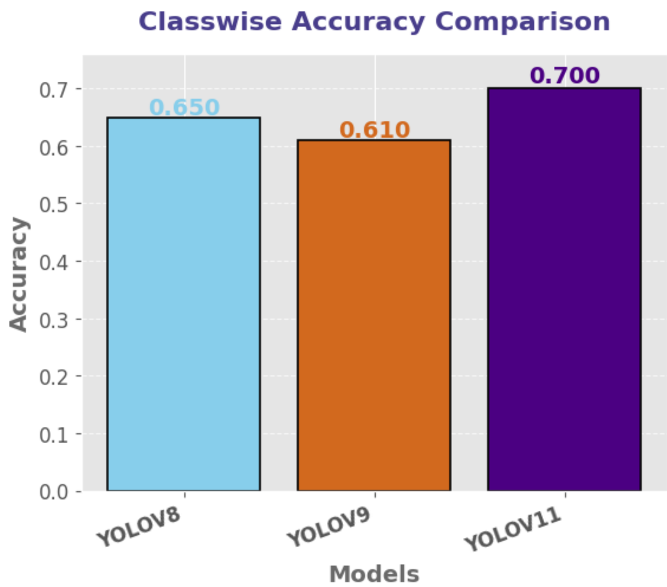


Figure 4.55: Class-Wise Model Accuracy Comparison Overlapping)

# Chapter 5

# Conclusion

In our research papers the significant progress we achieved in fish species classification using deep learning by implementing and analyzing four key models are Inception v3, ResNet50, VGG16 and a Basic CNN.Moreover,the study addressed challenges such as variations in illumination, background noise and fish orientation using advanced preprocessing techniques and segmentation. Inception v3 proved to be the most effective, excelling in accuracy and generalization due to its inception modules and multi scale feature extraction in making it ideal for real-world applications like fisheries management, industrial use and conservation of aquatic ecosystem.In addition, VGG16 emerged as the second best model which is also offering reliable accuracy with a deep but efficient architecture.Thirdly,the Basic CNN is less complex, performed well in resource limited scenarios, demonstrating the potential of models for specialized tasks. On the other hand, the classification accuracy according to our research integrated YOLO-based segmentation models like YOLOv11, YOLOv9 and YOLOv8 and due to their efficiency in real time object detection and the ability to process multiple objects simultaneously. Moreover, YOLOv11 outperformed the others, achieving the highest segmentation in measuring accuracy, the advanced architecture of YOLOv11 is improved and also anchor-free in detecting and has refined feature extraction. YOLOv9 balanced accuracy and speed efficiently, while YOLOv8 provided strong results with lower computational costs, making it suitable for resource constrained environments. The integration of YOLO models significantly refined the segmentation process thus leading to improved classification performance and demonstrating their effectiveness in detecting fish in overlapping and non overlapping scenarios. The inclusion of segmentation techniques and detailed accuracy comparisons further strengthened the study. This research holds significant ecological and societal relevance in offering solutions for biodiversity conservation ,sustainable fisheries management and aquatic ecosystem monitoring. Therefore, leveraging YOLO-based segmentation, this study highlights the importance of real time fish classification techniques thus paving the way for future advancements in marine research and conservation efforts.

## 5.1 Future Work

This research represents our ongoing efforts to classify and segment Bangladeshi fish classes by creating a custom dataset of 7100 images and applying Convolutional Neural Network models and segmentation models. Furthermore, the results of our experiments can be used to create a complete plan for future research and development.

- **Introducing Pipeline:** Our specialized classification model's accuracy is around 95-99 percent whereas the accuracy of segmentation is around 70-75 percent. Our target is to automatically extract every single fish from the segmentation dataset by cropping images and give those cropped images as an input to the classification models. We can expect that specialized classification models will give higher accuracy than segmentation. This hybrid pipeline can be effective in increasing the accuracy of segmented datasets with the help of classification models.

- **Dataset Improvement:** Our custom dataset contains 7,100 images in total. Our target is to add more real-world scenarios based on diverse images into this dataset. We may consider noises in the images. More overlapping images need to be added as well.

- **Model Refinement:** To improve the detection and classification, the research can be introduced to more popular object recognition and segmentation models, such as Yolact, Mask R-CNN, etc.

# Bibliography


[1] C. Spampinato, D. Giordano, R. Di Salvo, Y.-H. J. Chen-Burger, R. B. Fisher, and G. Nadarajan, "Automatic fish classification for underwater species behavior understanding," in *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, 2010, pp. 45–50.

[2] M.-C. Chuang, J.-N. Hwang, and K. Williams, "A feature learning and object recognition framework for underwater fish images," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1862–1872, 2016.

[3] D. Rathi, S. Jain, and S. Indu, "Underwater fish species classification using convolutional neural network and deep learning," in *2017 Ninth international conference on advances in pattern recognition (ICAPR)*, Ieee, 2017, pp. 1–6.

[4] P. Zhuang, L. Xing, Y. Liu, S. Guo, and Y. Qiao, "Marine animal detection and recognition with advanced deep learning models.," in *CLEF (Working Notes)*, 2017, pp. 166–177.

[5] N. E. M. Khalifa, M. H. N. Taha, and A. E. Hassanien, "Aquarium family fish species identification system using deep neural networks," in *International Conference on Advanced Intelligent Systems and Informatics*, Springer, 2018, pp. 347–356.

[6] F. Kratzert and H. Mader, "Fish species classification in underwater video monitoring using convolutional neural networks," 2018.

[7] Y. Ma, P. Zhang, and Y. Tang, "Research on fish image classification based on transfer learning and convolutional neural network model," in *2018 14th international conference on natural computation, fuzzy systems and knowledge discovery (ICNC-FSKD)*, IEEE, 2018, pp. 850–855.

[8] S. A. Siddiqui, A. Salman, M. I. Malik, *et al.*, "Automatic fish species classification in underwater videos: Exploiting pre-trained deep neural network models to compensate for limited labelled data," *ICES Journal of Marine Science*, vol. 75, no. 1, pp. 374–389, 2018.

[9] V. Allken, N. O. Handegard, S. Rosen, T. Schreyeck, T. Mahiout, and K. Malde, "Fish species identification using a convolutional neural network trained on synthetic data," *ICES Journal of Marine Science*, vol. 76, no. 1, pp. 342–349, 2019.

[10] P. Hridayami, I. K. G. D. Putra, and K. S. Wibawa, "Fish species recognition using vgg16 deep convolutional neural network," *Journal of Computing Science and Engineering*, vol. 13, no. 3, pp. 124–130, 2019.

[11] E. Olsvik, C. M. Trinh, K. M. Knausgård, *et al.*, "Biometric fish classification of temperate species using convolutional neural network with squeeze-and-excitation," in *Advances and Trends in Artificial Intelligence. From Theory to Practice: 32nd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2019, Graz, Austria, July 9–11, 2019, Proceedings 32*, Springer, 2019, pp. 89–101.

[12] M. Mathur, D. Vasudev, S. Sahoo, D. Jain, and N. Goel, "Crosspooled fishnet: Transfer learning based fish species classification model," *Multimedia Tools and Applications*, vol. 79, no. 41, pp. 31 625–31 643, 2020.

[13] B. Rekha, G. Srinivasan, S. K. Reddy, D. Kakwani, and N. Bhattad, "Fish detection and classification using convolutional neural networks," in *Computational Vision and Bio-Inspired Computing: ICCVBIC 2019*, Springer, 2020, pp. 1221–1231.

[14] D. Štifanić, J. Musulin, Z. Car, and R. Čep, "Use of convolutional neural network for fish species classification," *Pomorski zbornik*, vol. 59, no. 1, pp. 131–142, 2020.

[15] K. Dey, M. M. Hassan, M. M. Rana, and M. H. Hena, "Bangladeshi indigenous fish classification using convolutional neural networks," in *2021 International Conference on Information Technology (ICIT)*, IEEE, 2021, pp. 899–904.

[16] A. Majumder, A. Rajbongshi, M. M. Rahman, and A. Biswas, "Local freshwater fish recognition using different cnn architectures with transfer learning," *Int. J. Adv. Sci. Eng. Inf. Technol*, vol. 11, no. 3, pp. 1078–1083, 2021.

[17] M. Muslim and M. F. R. ZULKIFLI, "Convolutional neural network architectures performance evaluation for fish species classification," *Journal of Sustainability Science and Management*, vol. 16, no. 5, pp. 124–139, 2021.

[18] S. A. Shammi, S. Das, M. Hasan, and S. R. H. Noori, "Fishnet: Fish classification using convolutional neural network," in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, 2021, pp. 1–5.

[19] G. Wang, A. Muhammad, C. Liu, L. Du, and D. Li, "Automatic recognition of fish behavior with a fusion of rgb and optical flow data based on deep learning," *Animals*, vol. 11, no. 10, p. 2774, 2021.

[20] X. Yang, S. Zhang, J. Liu, Q. Gao, S. Dong, and C. Zhou, "Deep learning for smart fish farming: Applications, opportunities and challenges," *Reviews in Aquaculture*, vol. 13, no. 1, pp. 66–90, 2021.

[21] S. Y. Alaba, M. Nabi, C. Shah, *et al.*, "Class-aware fish species recognition using deep learning for an imbalanced dataset," *Sensors*, vol. 22, no. 21, p. 8268, 2022.

[22] A. Ben Tamou, A. Benzinou, and K. Nasreddine, "Live fish species classification in underwater images by using convolutional neural networks based on incremental learning with knowledge distillation loss," *Machine Learning and Knowledge Extraction*, vol. 4, no. 3, pp. 753–767, 2022.

[23] J. Deka, S. Laskar, and B. Baklial, "Freshwater fish species classification using deep cnn features.," *ICTACT Journal on Image & Video Processing*, vol. 12, no. 4, 2022.

[24] N. P. Desai, "Classifying different sea species using deep learning techniques," Ph.D. dissertation, Dublin, National College of Ireland, 2022.

[25] N. E. Garcia-D'Urso, A. Galan-Cuenca, P. Climent-Pérez, M. Saval-Calvo, J. Azorin-Lopez, and A. Fuster-Guillo, "Efficient instance segmentation using deep learning for species identification in fish markets," in *2022 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2022, pp. 1–8.

[26] K. M. Knausgård, A. Wiklund, T. K. Sørdalen, *et al.*, "Temperate fish detection and classification: A deep learning based approach," *Applied Intelligence*, pp. 1–14, 2022.

[27] S. Mittal, S. Srivastava, and J. P. Jayanth, "A survey of deep learning techniques for underwater image classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 6968–6982, 2022.

[28] A. Mohanty, G. Goldsztein, and R. Pellegrin, "Fish species image classification using convolutional neural networks," *Journal of Student Research*, vol. 11, no. 3, 2022.

[29] K. Auliasari, M. Wasef, and M. Kertaningtyas, "Leveraging vgg16 for fish classification in a large-scale dataset," *Brilliance: Research of Artificial Intelligence*, vol. 3, no. 2, pp. 316–328, 2023.

[30] R. M. Aziz, N. P. Desai, and M. F. Baluch, "Computer vision model with novel cuckoo search based deep learning approach for classification of fish image," *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 3677–3696, 2023.

[31] J. Deka, S. Laskar, and B. Baklial, "Automated freshwater fish species classification using deep cnn," *Journal of The Institution of Engineers (India): Series B*, vol. 104, no. 3, pp. 603–621, 2023.

[32] K. Volkan, İ. Akgül, and Ö. Z. TANIR, "Isvonet8: A proposed deep learning model for classification of some fish species," *Journal of Agricultural Sciences*, vol. 29, no. 1, pp. 298–307, 2023.

[33] A. Constantine-Macías, A. Toala-Paz, M. Realpe, J. Suárez-Moncada, D. Páez-Rosas, and E. P. Jarrín, "Leveraging deep learning techniques for marine and coastal wildlife using instance segmentation: A study on galápagos sea lions," in *2024 IEEE Eighth Ecuador Technical Chapters Meeting (ETCM)*, IEEE, 2024, pp. 1–6.

[34] Z. He, X. Xu, J. Luo, *et al.*, "A measurement model for aquatic animals based on instance segmentation and 3d point cloud," *IEEE Access*, 2024.

[35] S. Cheng, Y. Han, Z. Wang, S. Liu, B. Yang, and J. Li, "An underwater object recognition system based on improved yolov11," *Electronics*, vol. 14, no. 1, p. 201, 2025.

[36] F. J. P. Montalbo and A. A. Hernandez, "Classification of fish species with augmented data using deep convolutional neural network,"