DEBRE MARKOS UNIVERSITY

COLLEGE OF TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

INDIGENOUS FISH SPECIES IDENTIFICATION USING DEEP LEARNING APPROACHES.

MSC.THESIS

BY

ZELALEM ADUGNA TEREFE

ADVISOR: ESHETE DERIB. (PhD)

FEBRUARY 2025
DEBRE MARKOS, ETHIOPIA

INDIGENOUS FISH SPECIES IDENTIFICATION USING DEEP LEARNING APPROACHES

A Thesis Submitted to Debre Markos College of Technology, Post Graduate Studies Program

Debre Markos University

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Technology

By

Zelalem Adugna Terefe

Advisor: Eshete Derib. (Ph.D)

February 2025
Debre Markos, Ethiopia

DECLARATION

This is to certify that the thesis entitled Designing & Develop a Model for Indigenous Fish Species Identification Using Deep Learning Approaches.

For identifying objects and movement using Deep Learning Algorithms. I declare that this thesis is my original work and has not been submitted for any Degree at any other University. I have undertaken the study independently with the guidance and support of the research advisor.

Name: Zelalem Adugna

Signature 

Date of Submission 09/02/2025G.c

This MSc thesis has been submitted for examination with my approval as the thesis.

Advisor.

Name: Eshete Derib (PhD)

Signature: 

Date of Submission 09/02/2025G.c

DEBRE MARKOS UNIVERSITY

COLLEGE OF TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

INDIGENOUS FISH SPECIES IDENTIFICATION USING DEEP LEARNING APPROACHES

Submitted by:

Zelalem Adugna Terefe	____________________	09/02/2025G.c
Name of Student	Signature	Date

Approved by:

1.	Eshetie Derib (PhD)		09/02/2025 G.c
	Name of Major Advisor	Signature	Date
2.	____________________	____________________	____________________
	Name of Internal Examiner	Signature	Date
3.	____________________	____________________	____________________
	Name of External Examiner	Signature	Date
4.	____________________	____________________	____________________
	Name of Chairman	Signature	Date
5.	____________________	____________________	____________________
	Name of Department Head	Signature	Date

Acknowledgment

First and foremost, I give extraordinary thanks to my Almighty God and His Mother, Saint Mary, for giving me the courage and endurance to complete this research work.

I want to express my gratitude and heartfelt thanks to my advisor, Eshete Derib (Ph.D) for his interesting advice, constructive suggestions, comments, and encouragement from scratch. He gave me the direction of the research and his keen insight, continuous guidance and support, and unreserved advice.

In addition, I am also very grateful to the Bahir-Dar Fishery and Other Aquatic Life Research Center staff for their help in collecting the research data phase and evaluating our model. Their help and dedication have been crucial in gathering the necessary data for this research, and I sincerely appreciate their support. I am also very thankful to my instructors and all staff members of the ICT Directorates especially Mr. Getachew Kumie and Mr. Tigabu Tasie for their contribution in one way or another to the success of my study. I would like to extend my appreciation to my Family Mintamir Mekuraiw for her support.

Abstract

In ecological research, sustainable fisheries management, and biodiversity conservation, the identification of indigenous fish species is essential. Traditional expert systems are inconsistent and time-consuming for fish identification. To improve identification accuracy, various iterations of the You Only Look Once (YOLO) algorithm, such as YOLOv5, YOLOv7, YOLOv8, and YOLOv11, were used. The Bahir Dar Fishery and Other Aquatic Life Research Center in Ethiopia provided a collection of 13,000 images of 16 indigenous fish species. Various image enhancement techniques, CSPDarkNet, Histogram of Oriented Gradients (HOG), image segmentation-based feature extraction, and Roboflow annotations were applied to preprocess the dataset. We employed deep learning models with diverse hyperparameters in this study, such as different activation functions (ReLU, Softmax), optimizers (AdamW, SGD), batch sizes (16, 32), learning rates (0.001, 0.01), and epochs (50, 100). Standard criteria for fish identification, including mean Average Precision (mAP), precision, recall, and F1-score were used in the evaluation. The findings showed that the maximum mAP of 94.7% was attained by YOLOv11n with 100 epochs and AdamW optimizer, followed by YOLOv11s (50 epochs, SGD, 0.001 learning rate) on 94.4% and YOLOv11s (50 epochs, AdamW, 0.001 learning rate) on 94.0%. A mAP of 94.0% was also attained by YOLOv8n (100 epochs, AdamW, 0.01 learning rate), demonstrating the efficacy of modern YOLO models in fish species identification. YOLOv7 (50 epochs, Adam, 0.001 learning rate) demonstrated its comparative performance versus more recent models by achieving a mAP of 92.98%. The study concludes that YOLOv11 and other deep learning-based models offer a reliable and effective way to identify indigenous fish species.

Keywords: Indigenous Fish Species, Deep Learning, YOLO, Biodiversity Conservation

Table of Contents

List of Tables

List of Figures

List of Acronym

YOLO**:**	You Only Look Once
CNN:	Conventional Neural Network
ML:	Machine Learning
DL:	Deep Learning
RCNN:	Region-based Convolutional Neural Networks
DNN:	Deep Nural Network
XML:	Extendible Markup Language.
ANN	Artificial Neural Networks
RPN:	Region Proposal Network
SSD:	Single Shot Detector
CSP	Cross Stage Partial
CSV:	Comma-Separated Values
GPU:	Graphics Processing Unit
NMS:	Non-Maximum Suppression
SGD:	Stochastic Gradient Descent
FPN:	Feature Pyramid Network
SEG:	Segmentation
CLS:	Classification
OBB:	Oriented Bounding Box
SSP PANet:	Spatial Semantic Pyramid Network
mAP:	Mean Average Precision
Lr:	Learning Rate
PreLU:	Parametric Rectified Linear Unit
TN:	True Negative
TP:	True Positive
Lrf:	Learning Rate factor
ReLU	Rectified Linear Unit
FC	Fish Classification

CHAPTER ONE

1. INTRODUCTION

1.1 Background of the Study

Ethiopia's indigenous fish species are essential to the nation's aquatic biodiversity and fundamental to many people's means of subsistence. By supporting local economies through aquaculture and fishing, these species help ensure food security. Concerns over the preservation of these crucial ecosystems have been raised, nevertheless, by the growing dangers posed by habitat degradation, climate change, and overfishing. Effective biodiversity monitoring is made difficult by the inconsistency and time commitment of traditional fish species identification techniques, which frequently depend on expert knowledge[1].

As Research [2], [3], [4] Lake Tana, A hotspot for biodiversity, the largest lake in Ethiopia, and the source of the Blue Nile River is especially well-known for its fish species. Twenty of the lake's twenty-eight fish species are indigenous, which means they are found nowhere else in the world. This high degree of endemism highlights the distinctive biological circumstances of Lake Tana, where fish have evolved to particular water chemistry, temperature, and habitat architecture. These Indigenous species are essential to the ecosystem of the lake, supporting local fisheries and food webs that are essential to the livelihoods of the communities around them[5].

The diversity and unparalleled originality of Lake Tana's fishes are astounding. About 68% of the 28 species of big cyprinids that are known to exist in the lake are endemic, making it home to the sole flock of these animals in the world[6].

Conservation activities are crucial because overfishing, habitat loss, pollution, and climate change pose serious risks to this rich biodiversity. Furthermore, because they are a food supply and participate in customs, the indigenous fish species have cultural value. The health of the environment and the welfare of the communities that depend on it depend on the protection of these rare species[2].

In recent years, image classification tasks in a variety of domains, including ecology and biodiversity monitoring, have been transformed by deep learning approaches. Automated species discovery from visual data is now possible thanks to deep learning models' exceptional effectiveness in accurately classifying images, especially Convolutional Neural Networks (CNNs)[7]. This development offers a rare chance to create a reliable method for recognizing Ethiopia's Indigenous fish species, enabling prompt conservation and management actions [8].

Using deep learning techniques, the proposed study seeks to develop a model that can correctly identify indigenous fish species from images taken in their natural habitats. This study aims to offer a scalable system that improves the ability to monitor biodiversity and promotes sustainable fisheries management by fusing technology and ecological research [9].

Innovative monitoring techniques are required to address the swift reduction of indigenous fish species brought on by habitat loss, overfishing, and environmental changes. Conventional identification techniques are frequently time-consuming and necessitate specialized knowledge, which is not always accessible. The YOLO (You Only Look Once) model, a well-liked deep-learning technique for real-time object detection, presents a viable way to automatically identify Indigenous fish species from images taken in their natural environments[10].

By processing images in a single pass, YOLO sets itself apart from conventional object identification frameworks and enables real-time detection. The model efficiently detects many items in a single image by dividing it into a grid and predicting bounding boxes and class probabilities at the same time. Because of this feature, YOLO is especially well-suited for recognizing multiple fish species in a single frame, increasing the effectiveness of biodiversity monitoring initiatives[11].

1.2 Contribution and Novelty of the Study

The creation of a highly accurate model that improves species identification above conventional techniques is one of the study's main contributions to the field of deep learning approaches for Indigenous fish species recognition, which advances ecological monitoring activities and collecting data physically. Future studies in fish species identification and biodiversity evaluation will benefit greatly from the study's creation of an extensive dataset of images from various habitats and situations. Furthermore, the study showed how deep learning may be used to overcome difficulties in biodiversity monitoring and effectively incorporate contemporary technology into conservation initiatives. This study's focus on community involvement is an important component, as it seeks to inform and educate the local populace about the value of indigenous fish species and their ecological roles while encouraging stewardship. The study's focus on indigenous species which is frequently disregarded in previous research and its strategy of making cutting-edge technology accessible to non-experts through an intuitive application are what makes it distinctive. The study is positioned as a trailblazing endeavor that connects technology and ecological conservation because it combines scientific investigation and community engagement, which also provides insightful frameworks for further research in the area.

1.3 Statement of the Problem

There are several important issues with employing expert systems for fish species identification rather than deep learning image recognition techniques. Because expert systems depend on a structured knowledge base and a set of rules, their ability to recognize novel or uncommon species may be limited if these rules are not updated regularly [12]. Scalability problems may arise from the intricacy of managing a growing number of species since it becomes more difficult to create and maintain thorough rule sets [13]. Furthermore, expert systems are rigid in their ability to adjust to changes in environmental conditions or species traits since they are unable to learn from fresh data [14]. Biases may be introduced by the subjective nature of rule construction, leading to inaccurate species discovery[15]. Additionally, the strict frameworks of expert systems might not be able to fully capture tiny morphological changes between species, while deep learning algorithms are quite skilled at doing so because of extensive training on a variety of datasets [16]. It can take a lot of resources to maintain expert systems, which need constant expert input to keep the knowledge base up to date. Furthermore, these systems frequently have trouble with ambiguity and uncertainty, which raises the possibility of misclassification[17]. Lastly, developing a user-friendly interface for non-experts adds another layer of complexity, alongside the challenge of acquiring comprehensive initial data essential for effective rule development. These factors collectively underscore the limitations of expert systems in fish species identification compared to the more adaptive capabilities of deep learning techniques. Those Indigenous fishes are used for multipurpose community health and used for children with matured mindsets, so those Indigenous fishes are not accurately detected and get proper protection to use these fish for other purposes like food and other tasks[18]. From this statement of problem, the following research question was used to satisfy the specific objective of the study.

- ❖ What methods are most effective for collecting high-quality images of indigenous fish species in different aquatic habitats?
- ❖ Which deep learning model was used most effectively to enhance the model's performance in identifying indigenous fish species?
- ❖ How to evaluate this training model for detecting Indigenous species of fish in the study area?

1.4 Objectives

1.4.1 General Objective

The study's general objective was to develop a deep learning model for detecting Indigenous fish species using deep learning approaches.

1.4.2 Specific Objectives

The specific objective of the study was

- ➢ To gather a comprehensive dataset of images representing various indigenous fish species from diverse habitats.
- ➢ To train the deep learning model using different versions of the Latest State of Art Algorithm.
- ➢ To evaluate the effectiveness of the trained model’s performance based on different hyperparameters.

1.5 Scope

The scope of the study on indigenous fish species identification using deep learning approaches encompasses several key elements. It was focused on specific geographical regions known for their biodiversity, targeting freshwater lakes, rivers, and coastal areas that host unique indigenous fish populations from Lake Tana in Bahir Dar, the capital city of Amhara Regional State. The research was concentrated on a defined set of species that are ecologically significant or endangered, ensuring a comprehensive representation of local biodiversity. Data collection involved gathering a diverse dataset of images through field surveys and capturing various environmental conditions and angles to enhance model robustness. Deep learning approaches were the main tool used in the study, especially the CNN pre-trained algorithm, which we have used above YOLOv5 and variants including each. In addition to developing an easy-to-use application for non-experts to aid in species identification, evaluation criteria like accuracy, precision, recall, and F1-score were employed to gauge the model's efficacy. To raise awareness of indigenous fish species and biodiversity conservation, the study also involved local communities and conservation organizations. To give a fair assessment of the research findings, limitations such as possible biases in the dataset and difficulties in taking pictures were also noted.

1.6 Significance of the Study

The potential to improve ecological monitoring and biodiversity protection makes this study on the recognition of indigenous fish species using deep learning techniques significant. The study's precise identification and classification of indigenous fish species will help researchers, conservationists, and policymakers create focused conservation plans and management techniques by offering useful data. Compared to conventional methods, the use of sophisticated deep learning techniques enables more accurate and efficient species identification, enabling prompt responses to problems like habitat loss and climate change. Additionally, by encouraging a sense of responsibility for their natural resources, the study's user-friendly program can enable non-experts and local groups to take part in biodiversity monitoring. This engagement can raise awareness about the importance of indigenous species and promote sustainable practices in fisheries and conservation efforts. Ultimately, the study aims to contribute to the broader field of ecological research, providing insights that could be applied globally to protect and preserve aquatic biodiversity.

In general, this study was used for different stakeholders. The major one is this indigenous fish used for extracting fish oil for children’s minds to be fast and memorable so using this indigenous fish for food purposes is better use for children and fish trappers use this fish for children and sell it better price to the community. So, this model is used for automatically detecting this kind of fish for using such purposes. The other is used to trap these fish species and control them from other injured fish for better purposes rather than food and promote the vulnerability of this species.

1.7 Application Area of the Study

The developed deep learning model for indigenous fish species identification has several practical applications across various fields. From this application area, the following is listed below.

- ❖ **Monitoring Biodiversity:** The model can be used by ecologists and conservationists to keep an eye on fish populations in their indigenous environments. The model's precise species identification enables it to monitor biodiversity shifts, evaluate the effects of environmental changes, and pinpoint endangered species that require preservation.
- ❖ **Fisheries Management:** By accurately identifying species, the model can help fisheries implement sustainable management approaches. This can support sustainable fishing methods, avoid overfishing of species that are already at risk, and guarantee that restrictions are followed.
- ❖ **Ecological Research:** The model can be used to help researchers study fish distribution, behavior, and habitat preferences. The approach can expedite data-gathering procedures by automating species identification, freeing up researchers to concentrate on analysis and interpretation.
- ❖ **Community Involvement and Education:** To increase knowledge of Indigenous fish species and their ecological importance, educational programs can make use of the user-friendly application created in conjunction with the model. This tool can be used by schools and local groups to support sustainable practices and conservation initiatives.
- ❖ **Environmental Impact Assessment:** To examine how development initiatives affect aquatic ecosystems, the model can be used in environmental impact assessments. Precise species identification can guide mitigation plans and aid in evaluating possible threats to biodiversity.
- ❖ **Policy Development:** Using the model's observations, policymakers can create well-informed conservation laws and policies.

1.8 Organization of the Paper

This study was organized into five chapters. The first chapter includes an introduction of the study, the background of the study, research questions, a statement of the problem, general and specific objectives, the scope of the study, the significance of the study, the application area of the study, and Organization of the Study.

The second chapter includes a literature review of indigenous fish species classification and identification model and related works on fish species classification and identification and a comparison of different related works and algorithms used.

The third chapter included methodology. Data source, data preparation, data preprocessing, and feature extraction, the algorithm used to develop Indigenous fish species identification , and finally proposed model Architecture of the new model.

The fourth chapter includes the experiment result and a discussion of the result based on different algorithm hyperparameters and compression with different algorithm.

The last chapter includes the conclusion, recommendation, and future works of the paper and references of the paper.

CHAPTER TWO

2. LITERATURE REVIEW

2.1. Overview of the Study

The increasing concern over biodiversity loss has prompted researchers to explore innovative technologies for monitoring and conserving aquatic ecosystems. Indigenous fish species play a crucial role in maintaining ecological balance, yet their identification and identification remain challenging due to factors such as habitat complexity and morphological similarities among species. Recent advancements in deep learning have shown promise in automating the identification of fish species through image analysis, providing a valuable tool for researchers and conservationists alike.

Lake Tana supports a large fishing industry, mainly based on the *Labeobarbus* barbs, Nile tilapia and sharptooth catfish. According to the Ethiopian Department of Fisheries and Aquaculture, 1,454 tons of fish were landed in 2011 at Bahir Dar, which the department estimated was 15% of its sustainable amount. Nevertheless, in a review that compared catches in 2001 to those ten years earlier, it was found that typical sizes of both the tilapia and the catfish had significantly decreased, and populations of the *Labeobarbus* barbs that breed in the tributaries had significantly declined. Among the indigenous fish, most are considered threatened endangered , or vulnerable [19].

Figure 2. 1: Labeobarbus barbs and African sharptooth catfish caught in the lake

2.2 Indigenous Fish Species

Indigenous fish species are vital components of aquatic ecosystems, contributing to biodiversity, ecological stability, and local economies. They are often well-adapted to their environments, playing critical roles in food webs and ecosystem functions. However, many Indigenous fish species face threats from habitat degradation, climate change, and overfishing, leading to declines in their populations and biodiversity loss [20].

Research has shown that indigenous fish species are often particularly vulnerable to environmental changes due to their specialized habitat requirements and life history traits. For instance, [21] examines the ecological functions of indigenous fish species in freshwater environments, emphasizing their significance as markers of ecosystem health and in the cycling of nutrients. The ecological equilibrium may be upset by the extinction of these species, highlighting the necessity of efficient conservation measures. Technological developments have become effective instruments for tracking and protecting indigenous fish populations, especially in the fields of deep learning and image recognition.[22] created a deep learning model with high accuracy rates that is specifically intended for the identification of different fish species from images. Researchers can now quickly evaluate fish diversity across a range of environments thanks to this technology, which speeds up conservation efforts.

To comprehend the distribution and abundance of indigenous fish species, data collection techniques are also essential. As [23] stated that to enhance species identification models, extensive datasets containing images from various habitats are crucial. They contend that incorporating citizen science studies can improve data gathering and enable more community participation in biodiversity monitoring[24].

Furthermore, cooperation between scientists, decision-makers, and local populations is necessary for the conservation of indigenous fish species. As [25] highlighted the importance of education and awareness in promoting stewardship for aquatic environments while discussing different approaches for involving local stakeholders in conservation efforts. According to their findings, community involvement fosters sustainable practices that benefit local livelihoods and biodiversity in addition to improving conservation outcomes.

Fish species can be identified using families, genera, and species to gain a better understanding of the relationship between the species and the specific habitat. Accurate fish classification allows for the tracking and evaluation of aquatic biodiversity. Fish classification aids in monitoring the health of aquatic ecosystems and evaluating shifts in the distribution and population dynamics of fish species. Fish species identification can be used to develop conservation efforts [26].

Fish Classification (FC) plays a vital role in fields like fishery management and ecological research. Morphological characteristics like body shape and patterns are the main determinants of conventional FC techniques. Despite their effectiveness, these techniques might be interpreted subjectively and frequently need specialized understanding. However, the use of deep learning (DL) approaches for fish species classification has been made possible by technological improvements and the availability of large datasets [27].

Figure 2. 2: Fish classification in Ecological.

The successful conservation of indigenousfish species still faces obstacles despite these develop ments. The research [28] Stated that The statement draws attention to the potential biases in species identification and data collecting, which could result in insufficient conservation efforts. To guarantee the correctness and dependability of ecological evaluations, they advise the creation of regular procedures for data collecting [29].

One important use of deep learning (DL), which helps locate particular underwater species, like fish, is fish identification. There are several applications for identifying marine fish, and many techniques have been created to identify the right objects, helping people overcome this difficulty.

Furthermore, the fishing business and conservation initiatives depend on the tracking and counting of fish populations[29].

The literature concludes by highlighting the necessity of comprehensive conservation measures and the vital role that indigenous fish species play in preserving ecological equilibrium. Technological developments, community involvement, and reliable data collection techniques can greatly improve efforts to save these important species and their habitats[30]

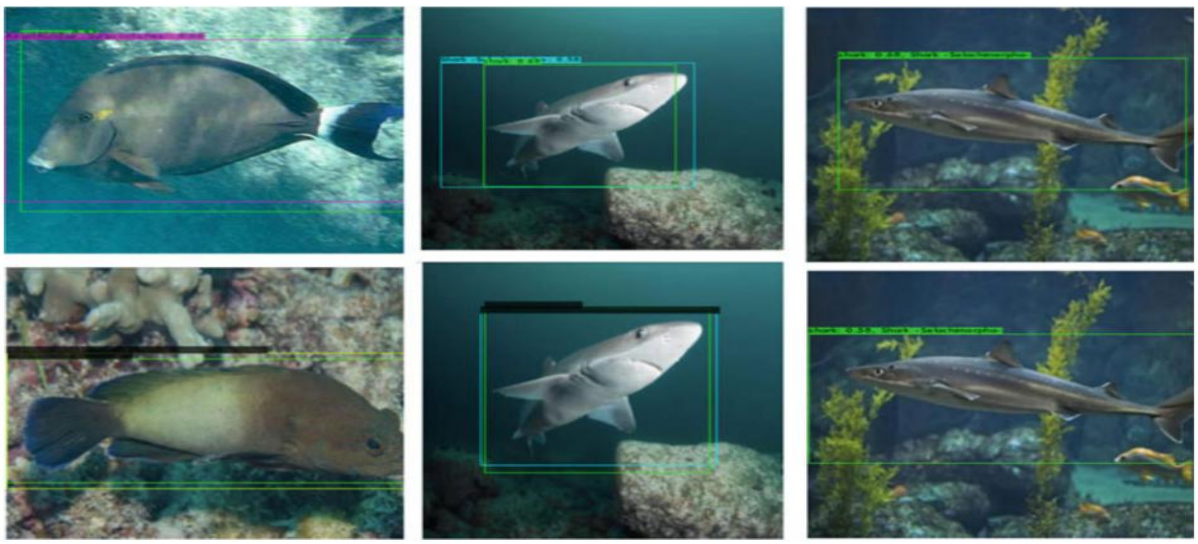


Figure 2. 3: Fish Species [25]

2.3 Deep Learning for Object Detection

Machines can now precisely recognize and locate many things inside images and video streams thanks to deep learning, which has greatly enhanced the field of object detection. Applications such as surveillance systems, medical imaging, and driverless cars all depend on these capabilities. Convolutional Neural Networks (CNNs), which are at the heart of the majority of contemporary object identification systems, are excellent at extracting features from images and enable the learning of intricate patterns [31].

2.3.1 CNN-Based Object Detection

One of the pioneering approaches, Region-Based Convolutional Neural Networks (R-CNN), introduced by [32] utilizes selective search to generate region proposals, which are then classified using CNNs. R-CNN is computationally demanding, though, which has led to advancements like Fast R-CNN, which increases efficiency by processing the full image to produce feature maps [32]. By incorporating a Region Proposal Network (RPN) that shares convolutional features, Faster R-CNN considerably improves performance and enables real-time identification [33].

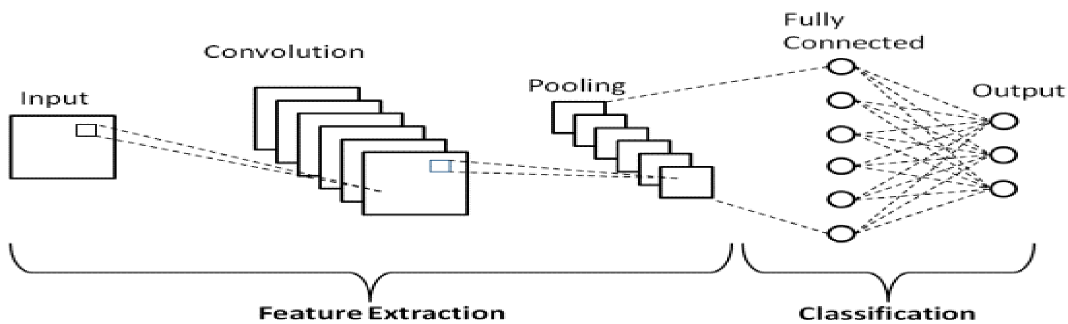


Figure 2. 4: CNN Architecture [29]

2.3.2 Single Shot Object Detection

In computer vision, Single Shot Object Identification (SSD) is a very effective technique that allows several objects in an image to be detected in a single forward run through a neural network. This method's speed and efficiency make it especially beneficial for real-time applications. At each feature map position, SSD discretizes the bounding box output space into a set of default boxes with different aspect ratios and scales. The network creates scores for each object category's presence in these default boxes during the prediction phase and modifies the boxes to better fit the objects that are detected[34].

The two primary parts of the SSD architecture are the SSD head, which has extra convolutional layers in charge of predicting bounding boxes and class scores, and the backbone model, which is usually a pre-trained convolutional neural network (CNN) such as ResNet. Because of its design, SSD can successfully recognize objects at different scales by utilizing information from different layers of the backbone model. In the SSD architecture, each grid cell is in charge of determining whether objects will be present in its area; anchor boxes are used to account for various item sizes and shapes[34].

One of SSD's main benefits is its capacity to strike a fair balance between speed and accuracy, which makes it appropriate for uses where real-time object identification is essential, like robotics, video surveillance, and autonomous driving [3]. All things considered, SSD is a breakthrough in object identification techniques, fusing the advantages of single-shot processing with strong identification powers.

2.3.3 Multiple Object Detection

Finding and identifying many items within a single image or video frame is the main goal of the crucial computer vision job known as multiple object detection. This procedure is crucial for several applications where the capacity to identify and categorize numerous objects in real-time is critical, such as robotics, autonomous driving, and video surveillance. Deep learning techniques are the mainstay of modern methodologies, with strategies like You Only Look Once (YOLO), Single Shot Detectors (SSD), and Region-Based Convolutional Neural Networks (R-CNN) becoming more and more popular. To improve identification accuracy for objects of diverse forms, these models make use of architectures that can process images effectively. They frequently use anchor boxes with varying sizes and aspect ratios. Challenges including managing occlusions, scale differences, and the requirement for real-time processing still exist despite the progress[34].

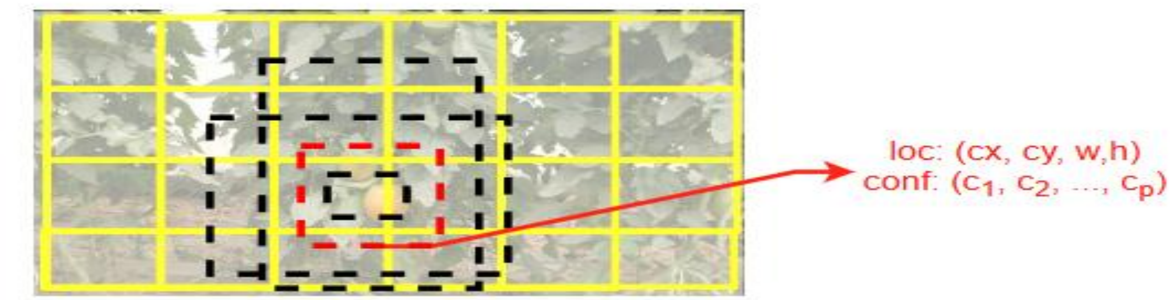


Figure 2. 5: Object Identification MOTA [30].

2.3.4 You Only Look Once (YOLO) Object Detection

You Only Look Once (YOLO), which frames object identification as a regression problem, enabling real-time processing capabilities [35]. RetinaNet, introduced by [36] Addresses class imbalance in object identification through a focal loss function, focusing on hard-to-detect objects. These technologies have several uses; for instance, object identification is essential to autonomous cars' ability to recognize barriers and pedestrians[10], while in medical imaging, it aids in detecting tumors and anomalies [11]. Despite the advancements, challenges remain, such as variability in object appearance due to changes in lighting and occlusion [37] and achieving real-time performance while maintaining high accuracy. Nevertheless, ongoing research continues to enhance the capabilities of object identification systems, promising smarter and more responsive applications in the future.

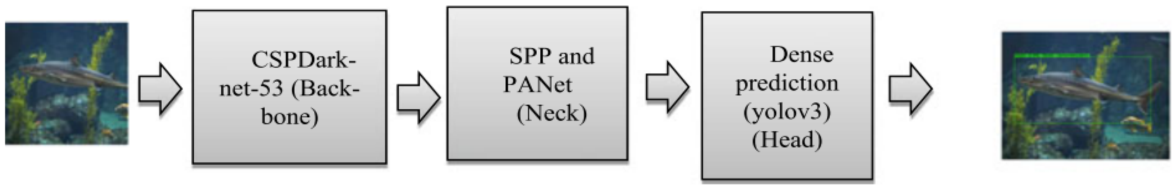


Figure 2. 6: Architecture of YOLO [34].

Numerous fish species are depicted in the picture, each with unique traits and ecological importance. One of these is the Black Sea Sprat, a slim, tiny fish distinguished by its silvery body and tendency to school. A highly sought-after catch in the culinary world, the Gilt Head Bream is distinguished by its muscular physique and golden stripe between the eyes. Known for its black stripes and sleek form, the Horse Mackerel is a swift swimmer that is typically found in warmer areas. The Red Mullet is prized in Mediterranean cooking and has whisker-like sensory organs. It is a brightly colored bird. Another popular species is the Red Sea Bream, which is prized for its firm texture and pinkish color. Known for its smooth body and mild flavor, sea bass can be used in a variety of recipes. Shrimp, which are praised for their sweet flavor and frequent use in seafood recipes, are also included even though they are not fish. While the Trout, a freshwater fish that can live in brackish water, is known for its delicate flavor and speckled appearance, the Striped Red Mullet, which is similar to the red mullet but has distinguishing stripes, is prized for its culinary attributes. Collectively, these species demonstrate the abundance of marine life and its significance for ecosystems and cuisine[1].

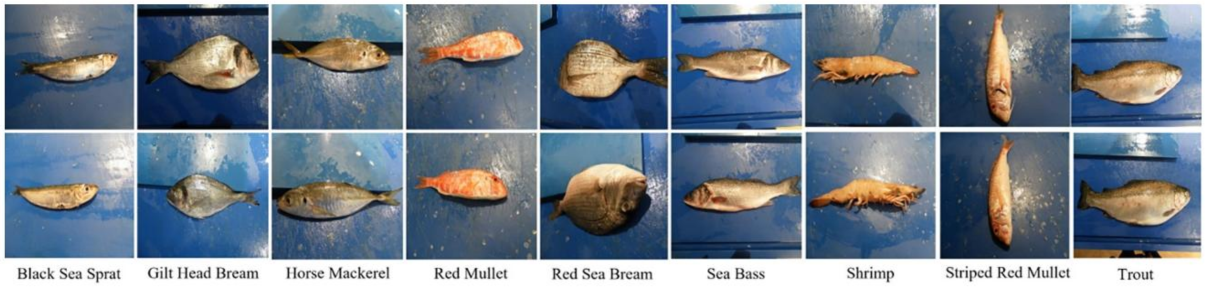


Figure 2. 7: Fish Species [1].

2.3.4.1 YOLOv5

According to [38], Ultralytics created the real-time object identification model YOLOv5 (You Only Look Once version 5), which is renowned for its effectiveness and speed. By analyzing images in a single pass and immediately predicting bounding boxes and class probabilities, it expands upon the YOLO architecture. The model is scalable with several sizes (small, medium, big, and extra-large) to accommodate various hardware constraints and performance requirements. It also has a redesigned neck and backbone structure for better feature extraction and aggregation. YOLOv5 successfully strikes a balance between precision and recall, achieving competitive results on benchmark datasets such as COCO and PASCAL VOC. With pre-trained weights and thorough documentation, it is simple to use and is implemented in PyTorch. It also includes capabilities like data augmentation, automatic mixed precision training, and compatibility with several deployment platforms, all of which are backed by a strong community that actively participates in its continuous improvement. Because of its speed, accuracy, and ease of use, YOLOv5 is a well-liked option for a variety of object identification applications [38].

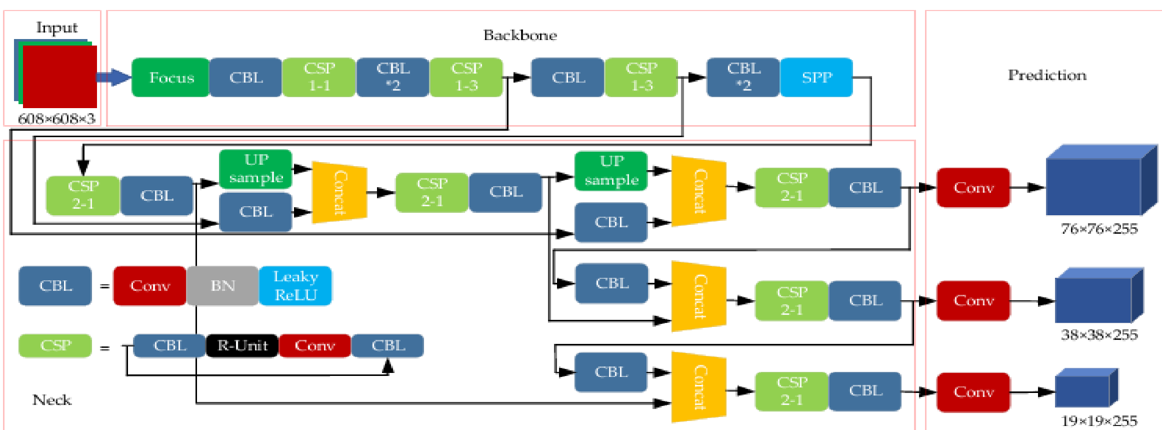


Figure 2. 8: Network Structure of YOLOv5

2.3.4.2 YOLOv8

The most recent development in the YOLO series of object identification
models, YOLOv8 (You Only Look Once version 8) is intended to provide better performance in real-time applications.

This version offers notable architectural enhancements that improve speed and accuracy, enablin g efficient object identification across a range of scales and environments [39]. Because of its improved feature aggregation techniques and more effective backbone network, YOLOv8 can process images rapidly without sacrificing accuracy. It is adaptable for deployment across many hardware configurations, from edge devices to potent GPUs, thanks to its support for varied input resolutions and a variety of model sizes. Additionally, YOLOv8 optimizes computing resources without compromising identification accuracy by integrating sophisticated data augmentation algorithms and supporting automatic mixed precision training [40].

2.3.4.3 YOLOv7

The object identification module (ODM) uses the improved YOLOv7 algorithm, which improves identification performance. The Study [1] Outlines the recognition module's use of the FD_Net technique. This method computes the loss and extracts features using a network. The enhanced DenseNet-169 network is used for feature extraction, and the Arcface loss (AFL) function is used as the loss function.

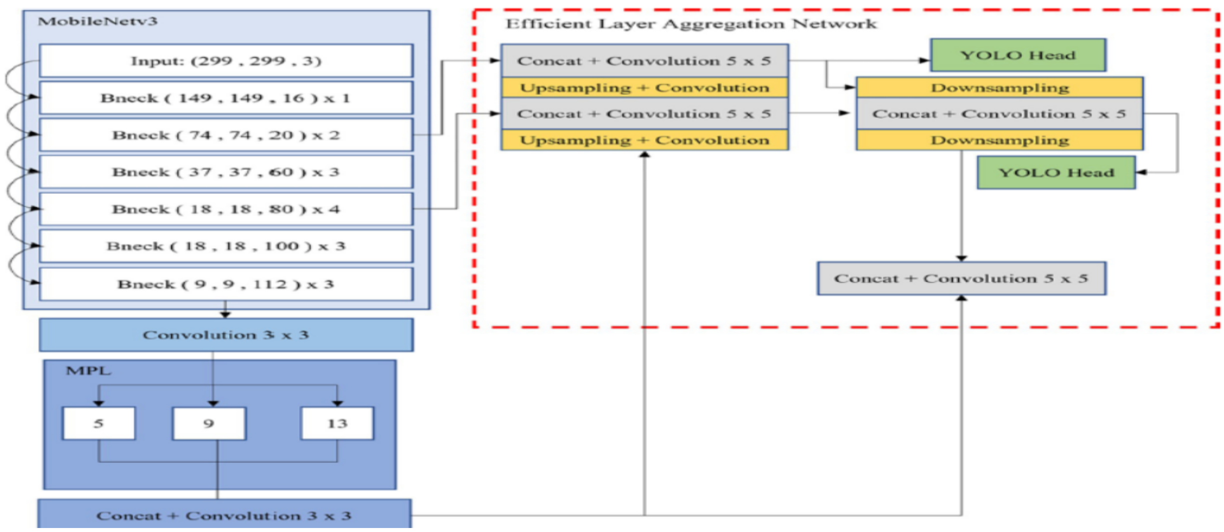


Figure 2. 9:Enhanced YOLOv7 architecture.

2.3.4.4 YOLOv11

Real-time object identification and image segmentation have advanced significantly with Ultralytics' YOLOv11. Building on the advantages of its predecessors, it improves object identification and handles complicated tasks more effectively by introducing increased feature extraction capabilities with an upgraded backbone and neck design. The model offers quicker processing speeds that are perfect for large-scale and real-time applications by striking a balance between precision and efficiency. The five model sizes of YOLOv11 nano (n), small (s), medium (m), large (l), and extra-large (x) allow customers to select the one that best suits their needs. With mean Average Precision (mAP) scores that range from 39.5% for the nano model to 54.7% for the extra-large model, as well as variable inference timeframes, to accommodate a variety of use scenarios, it exhibits remarkable performance on the COCO dataset. Additionally, YOLOv11 provides specialized variations such as YOLOv11-position for posture estimation, YOLOv11-obb for orientated object recognition, YOLOv11-cls for image classification, and YOLOv11-seg for instance segmentation. The adaptability of YOLOv11 is increased by these improvements and specific models, which make it a strong option for a variety of computer vision applications. [29].

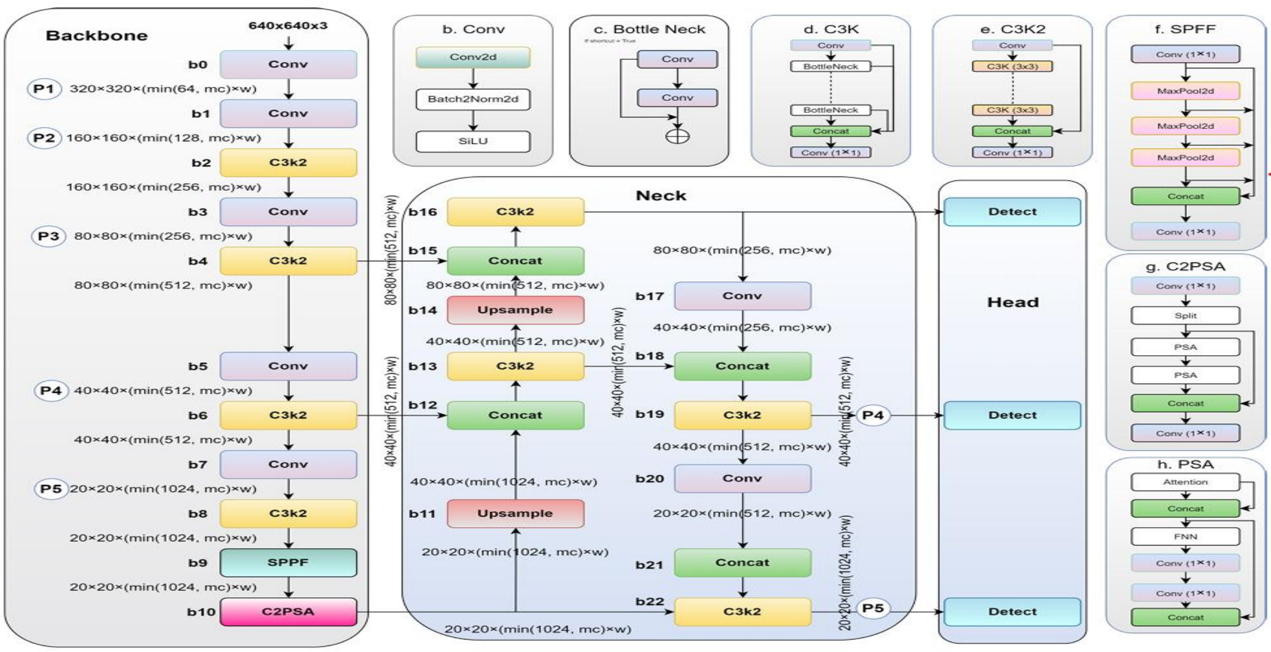


Figure 2. 10: YOLOV11 Architecture.

Figure 2.5 above is an improved version of the YOLO (You Only Look Once) family of object identification frameworks, the YOLOv11 model is intended to improve performance in real-time applications. With several configurations for efficient feature capture at varied resolutions, its backbone extracts key visual features over several stages (P1 to P5). By adding layers like C3K to improve features, the model's neck accumulates these feature maps and gets them ready for the identification head. The identification head integrates layers that make last-minute modifications depending on processed features to provide the final output, which includes bounding boxes and class probabilities for recognized objects. YOLOv11 is ideally suited for a range of real-time object identification jobs since it strives to achieve a balance between speed and accuracy.

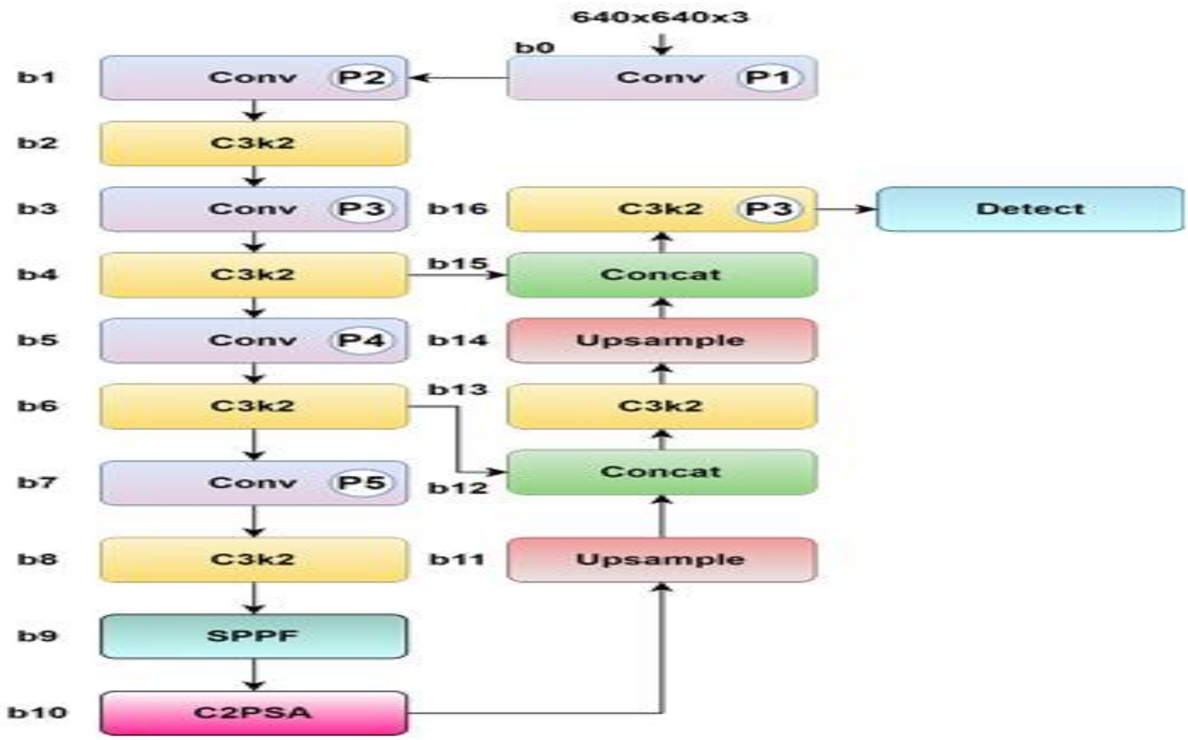


Figure 2. 11: YOLOv11-Small: Architecture Optimized

2.4 Related Work

Fish identification and classification have been the subject of numerous articles in recent years, most of which concentrate on sophisticated deep-learning algorithms and conventional image-processing methods. For example, studies that use simple image processing techniques like edge identification and segmentation have shown a moderate level of accuracy in controlled settings, but their practical application is limited because they frequently struggle in complex backdrops. This structured table template can be used to compare various studies on the identification and classification of fish species:

Table 2. 1: Summery of Related Work

Title	**Author(s)**	**Year**	**Algorithm Used**	**Result in %**	**Limitation or Gap**
Mobile Application to Identify Fish Species Using YOLO [41]	K. Priyankan, T.G.I.	2024	YOLO (You Only Look Once) v3	Accuracy of 77%	Potential challenges in real-time processing and accuracy under varying underwater conditions.
Under Water Fish Species Classification Using CNN [25]	Dhruv Rathi, Sushant Jain, Dr. S. Indu	2021	CNN	Accuracy of 96.29%	Limited by noise and occlusion in underwater images; may not generalize well to all fish species.
Fish Species Identification Using Deep Learning for Industrial Applications [23]	Srinivasa H, Gowrishankar S Nath	2021	Faster R-CNN, CNN	Accuracy of Faster R-CNN 98%, and CNN 95%	High dependency on the quality of input images; performance may degrade in complex industrial environments.
Multi-classification deep neural networks for the identification of fish species using camera-captured images [1]	H. Malik, A. Ahmad	2023	Deep Neural Networks (DNN)	Achieved high accuracy for fish species classification.	Limited dataset diversity; performance may degrade in highly variable environments.

Fish Species Classification from Underwater Images using Large-Scale Dataset via Deep Learning Fish Species Classification from Underwater Images using Large-Scale Dataset via Deep Learning [42]	A. Salman, A. Jalal, F. Shafait, A. Mian, M. Shortis, J. Seager, E. Harvey	2016	Convolutional Neural Networks (CNN)	Correct classification rate > 90% using LifeCLEF datasets.	Challenges with environmental variability; robustness not thoroughly tested across extreme conditions.
Fish Identification and Classification for Automatic Sorting System with an Optimized YOLO Algorithm [11]	A. Kuswantori, T. Suesut, W. Tangsrirat, G. Schleining, N. Nunak	2023	YOLOv4	98.15% accuracy in detecting fish species	No existing dataset for cultured fish on conveyors

CHAPTER THREE

3 METHODOLOGIES

3.1 Description of the Dataset

The Amhara Regional Bahir-Dar Fishery and Other Aquatic Life Research Center, a significant organization devoted to the study and preservation of aquatic ecosystems in the Amhara region of Ethiopia, provided the carefully curated dataset for the deep learning method of fish species identification. The study of indigenous species, which are essential to preserving the biodiversity and ecological balance of nearby water bodies, is greatly aided by this center. 16 of the 20 known indigenous fish species were recorded in this study, demonstrating the center's dedication to gathering a representative sample of the area's distinctive ichthyofauna.

The dataset's exceptional collection of 13,000 images was created by strategically positioning advanced camera systems to capture the fish from various angles after being taken out of the water, ensuring high-quality imagery in a range of lighting conditions without the need for underwater shots. These images offer a wide range of data points that improve the resilience of the deep learning models being created, not only in terms of species but also in terms of characteristics like age, size, and behavior.

This dataset's diverse character, which includes habitats in shallow waters as well as deeper areas, reflects the region's ecological variety. Every picture is useful documentation of the morphological traits and behavioral patterns of the species, which helps create algorithms that can accurately classify species. Additionally, the dataset is crucial for continuing conservation initiatives since it allows scientists to track population trends, evaluate the condition of habitats, and create plans for the survival of these indigenous species. Overall, this comprehensive dataset not only contributes to the advancement of deep learning techniques in species identification but also supports broader ecological and conservation goals within the region.

Table 3. 1: Species and number of Datasets

No	Species	Total Dataset
1.	ClariasGariepinus	810
2.	LabeobarbuGorguari	800
3.	LabeobarbusAcutirostis	810
4.	LabeobarbusBrevicephalus	810
5.	LabeobarbusCrassibarbis	810
6.	LabeobarbusDainellii	812
7.	LabeobarbusDegenii	810
8.	LabeobarbusIntermedius	810
9.	LabeobarbusMacrophtalmus	810
10.	LabeobarbusMegastoma	814
11.	LabeobarbusNedgia	818
12.	LabeobarbusPlatydorsus	812
13.	LabeobarbusSurkis	816
14.	LabeobarbusTruttiformis	818
15.	NemachilusAbyssinicus	820
16.	OreochromisNiloticus	820
Total		13000

As shown in above Table 3.1 the data preparation and collection, we have concentrated on the consistency and ratio of the dataset to train and test the model. So, each 16 species of fish have almost a similar number of images to train, and to test the model we have to take an equal number of images from the total dataset based on the ratio of the dataset.

3.2 Data Preparation

After collecting the relevant dataset from this research center from all species of fish in condition with the treatment and workshop of the research center we have collected all of the dataset for developing this deep learning model. So based on this we have prepared and adjusted the data to make the model better and acutely identify the exact indigenous species of fish.

Data preparation techniques for fish species data sourced from the Bahir-Dar Fishery and Other Aquatic Life Research Center are essential for generating high-quality datasets that enhance research and modeling efforts. The process begins with data collection, gathering a diverse array of images and videos of various fish species captured during research activities and fieldwork at the center. Preprocessing methods are used after data collection to guarantee its variability and resilience. The model can now learn from a wider variety of visual representations and situations that fish may encounter in their actual habitats thanks to the implementation of data augmentation techniques including rotation, flipping, and color modifications.

Pixel values are then normalized to provide consistent input for the YOLOv11 model, which is essential for precise identification and classification. After that, images are scaled to a common size, usually 640x640 pixels, to provide consistency throughout the dataset and make it easier for the model to comprehend them. Another crucial step is cleaning the dataset, which entails deleting unnecessary or subpar images and carefully examining the annotations to ensure that only high-quality training data is kept.

The next step is to prepare the model's data using various training and validation ratios. The dataset is divided into different training and validation sets utilizing ratios like 70% training and 30% validation or 80% training and 20% validation to get it ready for efficient model training. This section is essential for assessing the model's performance during training and making sure it performs effectively when applied to new data. Roboflow techniques are used to accurately label the images, allowing researchers to identify and mark the fish species found in each picture. To ensure the data can be used by the model efficiently, the annotation data is then transformed into formats that are compatible with the YOLO algorithm, such as XML, CSV, and YOLO format. All of the categorized images in our example are converted to Text and images for each Text by Roboflow. According to [43], the necessity to successfully balance model training and evaluation is the basis for the choice of 80/20, 70/30, and 90/10 ratios for dividing datasets into training and testing sets. The 80/20 split is frequently used because it leaves a sizeable amount of data for testing the model's performance on unknown data, guaranteeing accurate assessment metrics, while still providing a huge amount of data for training, enabling the model to learn from a large number of examples. Because the 70/30 split a larger fraction of data for testing, which can assist reduce the danger of overfitting, it is frequently used when the dataset is smaller or when a more thorough evaluation of the model's generalization skills is required. While there is a chance that there will not be enough testing data to sufficiently validate performance, the 90/10 ratio is occasionally selected for larger datasets where a stronger focus on training data can improve the model's learning. In the end, these ratios are chosen to assist the main objective of creating strong and trustworthy predictive models by striking the ideal balance between successfully training the model and making sure it can generalize well to new data.

Table 3. 2: Dataset Splitting with ratio

Ratio	**Training Data**	**Testing Set**	**Total Samples**
80/20	10,400	2,600	13,000
70/30	9,100	3,900	13,000
90/10	11,700	1,300	13,000

Model training uses a Windows 11 operating system with the Anaconda Python library installed, which makes it possible to install the required machine-learning modules, such as PyTorch and TensorFlow. Additionally, Google Colab is an excellent training platform since it provides free RAM and GPU resources to users, which significantly enhances testing and training. To ensure that the model can correctly detect and classify different fish species, its accuracy is rigorously evaluated using the validation dataset after training. By using these meticulous data preparation methods, researchers can significantly improve the quality of fish species data from the Bahir-Dar Fishery and Other Aquatic Life Research Center. As a result, ecological research and the identification of aquatic species will produce more reliable and accurate results.

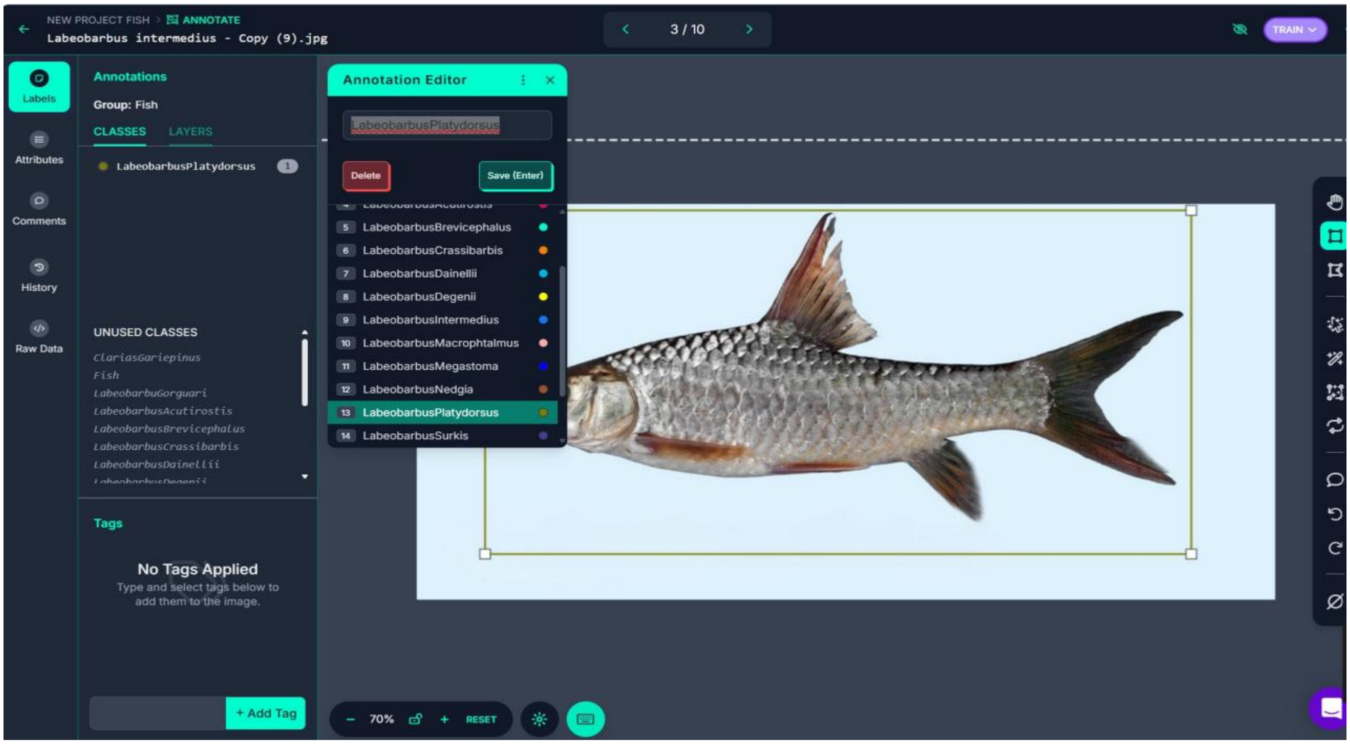


Figure 3. 1: Data preparation Using Roboflow

3.3 Data Preprocessing and Feature Extraction

Fish species identification is accomplished by feature extraction and data preparation.

This incorporates YOLOv11 CSPDarkNet, Histogram of Oriented Gradients (HOG), image seg mentation, and picture enhancement. The captured video has been transformed into a series of image frames, which have then been annotated (labeled) into the above 16 classes.

The Python source code for the Image-labeling tool is accessible on GitHub. It may be integrated with the Anaconda Python IDE installation and installed on Windows.

To extract more features, we used YOLOv11. So, the Feature extraction is primarily achieved through the YOLOv11 architecture, which automatically learns to identify patterns and characteristics specific to the target fish species during the training process. Leveraging a pre-trained YOLOv11 model can enhance feature extraction and reduce training time through transfer learning. Post-processing techniques, such as non-maximum suppression (NMS), are applied to filter out overlapping bounding boxes, ensuring that only the most confident predictions are retained. This comprehensive approach not only aims to improve identification accuracy but also contributes to the conservation efforts and biodiversity management of Lake Tana's indigenous fish populations.

We have also used histogram-oriented gradient (HOG) features to enhance the training feature using a version of YOLO.

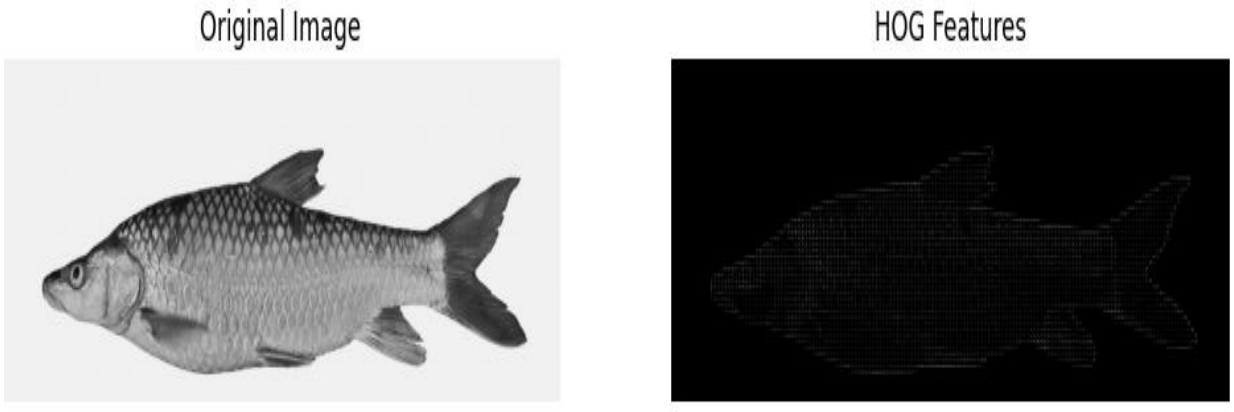


Figure 3. 2: HOG feature extraction method

3.4 Noise Removal and Image Enhancement

The process of improving an image to a higher, more comprehensible quality is called image enhancement. Improving the quality of the image is the goal of image enhancement, with the treated image being superior to the original for object Detection. Noise reduction, edge enhancement, and contrast improvement are all included in image enhancement. Noises in images might include periodic noise, Gaussian noise, speckle noise, salt and pepper noise, and other noises. These kinds of sounds can be eliminated from an image using a variety of methods. For example, median, geometric, and arithmetic mean filters.

3.5 Algorithm Used

To develop a model for detecting the species of fish based on the given dataset we have used the basic three Version of YOLO. From this, we have used YOLOv11, YOLOv8 YOLOv7, and YOLOv5 pre-trained algorithms. To train and test the Model we used the YOLOv11 Latest State of Art deep learning algorithm. To guarantee successful learning, the YOLOv11 training process includes many essential elements. Initially, a mix of localization loss and classification loss is used, where the former is concerned with correctly recognizing the fish species and the latter with precisely predicting the bounding boxes surrounding detected objects. The backpropagation algorithm is used to optimize the model's performance, enabling weight adjustments based on the estimated loss. Furthermore, optimizers like AdamW or Stochastic Gradient Descent (SGD) are used to reduce the loss function during training, which promotes effective convergence and raises the model's overall accuracy. This methodical approach guarantees that YOLOv11 will acquire the ability to recognize and detect indigenous fish species.

3.6 Evaluation Metrics

The performance and accuracy of the deep learning algorithms in YOLOv11 and Deep-SORT are evaluated using some evaluation metrics. When creating a deep learning model, the main considerations are the model's mean aggregate performance and overall performance[44]. So, in this research we tested the performance of the model by the accuracy of the model, Recall and Mean Average precision, and F1 score of the Evaluation Metrics.

In this study, different quantitative parameters are used for evaluating the performance of deep learning-based algorithms YOLO and Deep-SORT person detecting, measuring the social distance between people, and tracking people. The evaluation metrics used in this study are listed below:

Intersection over Union (IOU): is a term used to describe the extent of overlap of two boxes. The greater the region of overlap, the greater the IOU [45]. IOU is simply an evaluation metric for object identification and tracking. IOU is mainly used in applications related to object detection, where we train a model to output a box that fits perfectly around an object [46]. Figure 3.10, describes the intersection over union of the object that is represented by BOX1 and BOX2. The coordinates for BOX1 1 are represented by [x1, y1, x2, y2], and BOX2 is represented by [x3, y3, x4, y4]. The IOU between the two boxes is calculated as shown in equation (3.6).

$$IOU = \frac{Area\ of\ intersection\ of\ two\ boxes}{Area\ of\ union\ of\ two\ boxes} \qquad \text{Equation. 3. 1}$$

Where the intersection of two boxes indicates the common box that is found in the ground truth and unknown image.

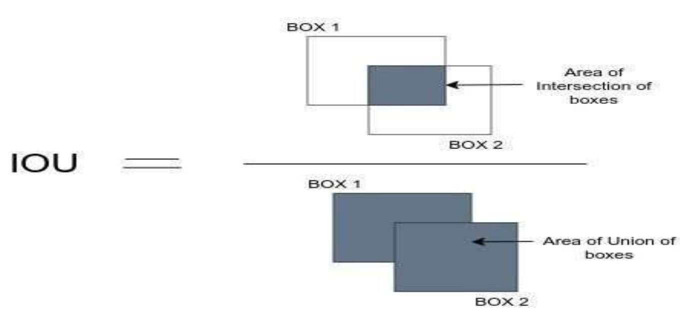


Figure 3. 3: IOU in Object Detection

The IOU of two boxes can have any value between 0 and 1. IOU is also used in non-max suppression, which is used to eliminate multiple boxes that surround the same object, based on which box has a higher confidence. Before starting on evaluation metrics, it is important to compute the true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

TP: indicates the model or the system predicts the value as Positive and the actual value as also Positive.

TN: indicates the model or the system predicts the value as Negative and the actual value as also Negative.

FP: indicates the model or the system predicts the value as Positive but the actual value is Negative. FN: indicates the model or the system predicts the value as Negative but the actual value is also Positive.

Accuracy: The most intuitive performance measure provides information regarding how well the algorithm can correctly classify inputs [47]. It is the most popular performance measure used and for good reason. It is extremely helpful, simple to compute, and easy to understand. This is a measure of the actual performance of the system in both correctly detecting and correctly rejecting targets.

$$Accuracy = \frac{TN+TP}{TP+FP+FN+FP} \qquad \text{Equation. 3. 2}$$

Precision: The number of true positives relative to the sum of the true positives and the false positives. That is, precision is the fraction of detected items that are correct. It estimates the predictive value of a class label, either positive or negative, depending on the label for which it is calculated in another expression it assesses the predictive power of the algorithm [47].

$$Precision = \frac{TP}{TP+FP} \qquad \text{Equation. 3. 3}$$

Recall: approximates the probability of the positive class label being true; in another expression, it assesses the effectiveness of the algorithm on a single class [47]. The number of true positives relative to the sum of the true positives and the false negatives. It is computed as shown in equation 3.4.

$$Recall = \frac{TP}{TP+FN} \qquad \text{Equation. 3. 4}$$

F1-Score [48]: -an overall measure of a model's accuracy that combines precision and recall, in that weird way that addition and multiplication just mix two ingredients to make a separate dish altogether. It gives a more balanced approach towards recall and precision. F1-score is a weighted average of recall and precision. It is the most important parameter than accuracy when having an uneven class distribution in the dataset. It is calculated as shown in equation 3.5:

$$F1 - score = 2 * \frac{precision * recall}{precision + rec} \qquad \text{Equation. 3. 5}$$

Multiple Object Tracking Accuracy(MOTA) [49] : it combines false negatives, false positives, and identity switches into a single score to express overall performance with a single value. To compute MOTA as shown in 3.6.

$$MOTA = 1 - \frac{\sum_t mt + fpt + mmet}{\sum_t gt} \qquad \text{Equation. 3. 6}$$

Where mt, and are the number of misses of false positives, of mismatches, and the number of GT objects respectively for time t.

Mean Average Precision: is the most commonly used evaluation metric that is calculated by taking the mean AP over all classes depending on different identification challenges. It is a measure that combines recall and precision for detecting the accuracy of the object [10], where it is calculated with the average precision value for recall value over 0 to 1 with IOU.

$$AP = \frac{\sum_T^R = 1 Pr}{R}$$

$$mAP = \frac{1}{N} \sum_{n \epsilon N} AP(n) \qquad \text{Equation. 3. 7}$$

Where AP is the average precision, PR precision of object detection, and N is the number of datasets used.

3.7 Proposed Model Architectures

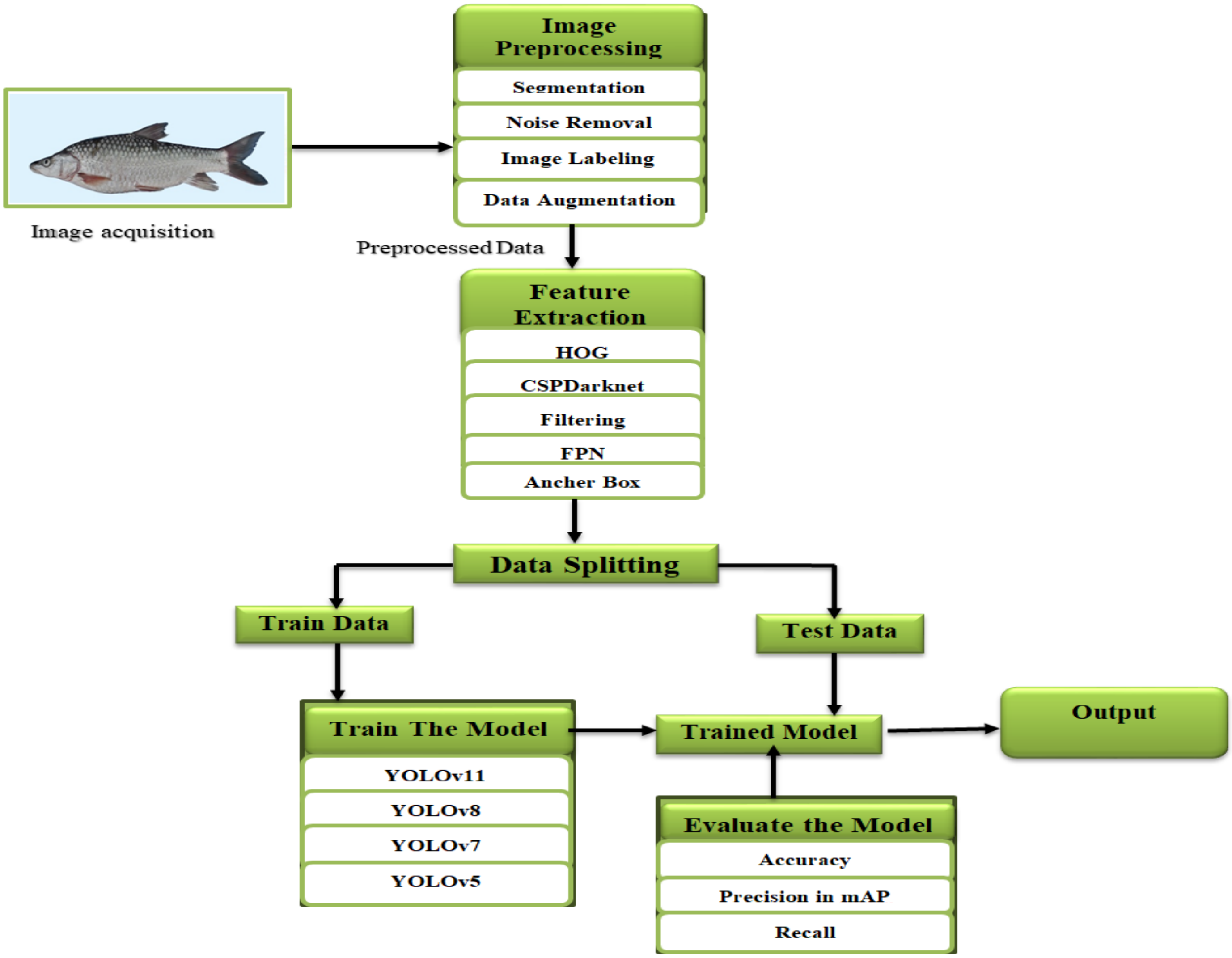


Figure 3. 4: Architecture of the proposed Model.

3.8 Development Tools and Techniques

The entire system has been tested on Windows 11 and developed in Python using Google Colab, which is a popular programming language for Data Science, Machine Learning, and Deep Learning tasks. TensorFlow and PyTorch serve as the frameworks for building and running deep learning models on GPUs. For image processing, OpenCV has been extensively utilized. Ground-truth annotations for validating the object detectors were created using OpenLabeling, specifically with the LabelImg tool.

Python is an interpreted high-level language chosen for implementing the proposed system due to its simplicity and powerful capabilities for developing deep learning applications.

TensorFlow is a widely used platform for deep learning object identification algorithms. It offers an end-to-end framework that simplifies the process of building and deploying deep learning models and serves as the backend for the Keras API. Developed by Google, TensorFlow is a cutting-edge Python library designed for efficient numerical computation.

PyTorch is an optimized tensor library primarily used for deep learning applications, particularly in state-of-the-art object identification like YOLOv5, YOLOv7, YOLOv8, and YOLOv11, utilizing both GPUs and CPUs. It is an open-source machine learning library for Python.

Google Colab is a free Jupyter Notebook environment hosted by Google that operates in the cloud. It provides free access to GPUs (Graphical Processing Units) and TPUs (Tensor Processing Units). In this study, Google Colab's GPU and Jupyter Notebook were employed for training and testing to detect individuals within image frames.

Roboflow is a free, open-source tool designed for graphically Roboflow images, enabling the identification and marking of specific details within images. We utilized Roboflow to annotate image frames extracted from video data.

Various Libraries:

- **NumPy** is a library that provides support for matrix and multi-dimensional array data formats. It allows for performing mathematical operations on arrays, including statistical, algebraic, and trigonometric routines. We used NumPy for pre-processing stages to enhance images for better pixel quality[50].

- **OpenCV**, or the Open-Source Computer Vision Library, is employed in computer vision and deep learning algorithms. It offers various libraries for object identification using deep learning techniques. In this study, OpenCV was used for identification processes within the system, specifically with YOLOv8, YOLOv5, YOLOv7, YOLOv11, and Deep-SORT, and for estimating social distances[51].
- **Matplotlib** is a plotting library that provides a wide range of built-in plots for the graphical representation of data. In this study, it was utilized for histogram equalization during feature extraction using HOG and for image enhancement techniques[52].

3.9 Hyper Parameters Used in the Model

Optimizers: - Optimizers are mathematical functions, which are dependent on the model's learnable parameters i.e. Weights and Biases. Optimizers help to know how to change weights and learning rate of neural networks to reduce the losses. In this study, we used AdamW and SGD in our experimentation process. AdamW combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems [53]. AdamW is relatively easy to configure where the default configuration parameters do well on most problems. SGD is an iterative method for optimizing an objective function with suitable 38 smoothness properties.

Epoch: - Indicates the number of passes of the entire training dataset the algorithm has completed. It refers to how many times the model reads all the data set. We have used epoch sizes 50 and 100. to train our model [54].

Batch size: - Refers to the number of training examples utilized in one iteration. We used batch size is 16 and 32 in this study[55],[56]

Learning rate: - a learning rate of 0.01 and 0.001 have been used in this study. Lowering the learning rate allows to the optimization weight updates. According to [57], a large learning rate allows the model to learn faster, at the cost of arriving at a sub-optimal final set of weights. A smaller learning rate may allow the model to learn more optimal but may take significantly longer to train.

3.10 Activation Function

In [58] study, the sigmoid activation function, and the PReLU (Parametric Rectified Linear Units) activation function have been used to decide to activate neurons to get the desired output. Sigmoid functions are used in deep learning for logistic regression and basic neural network implementations and they are the main in activation units. This sigmoid activation function is more efficient to compute than Sigmoid-like functions need to pick max (0, x) and not perform expensive exponential operations[59].

The mathematical definition of the sigmoid activation function is shown in equation 3.8.

$$f'(x) = f(x)\big(1 - f(x)\big)\ f(x) = \mathrm{sigmoid}(x) = \frac{1}{1+e^{-x}} \qquad \text{Equation 3.8}$$

So, the more there are layers in our deep learning neural network the more our information is compressed at each layer. The PReLU activation function is used to predict by minimum loss, faster

than Swish and ReLU with higher accuracy prediction. Parametric ReLU, is a variant of Leaky ReLU. PReLU is instead of using a fixed slope like 0.01 used in Leaky ReLU, a parameter ReLU changes depending on the model, for x < 0. It is used to improve the performance of object detection[59]

classification and little risk of overfitting.

The mathematical definition of a parameter ReLU activation function is computed as shown in equation 3.9.

$$f(x) = \begin{cases} x, \ if\ x > 0 \\ ax\ x < 0 \end{cases} \qquad \text{Equation 3.9}$$

Generally, the sigmoid function and PReLU were used to provide smooth control of the input/output relationship

so based on the [60] we have used SoftMax and ReLU activation to develop the model using the basic dataset and other hyperparameters.

CHAPTER FOUR

4. EXPERIMENTAL RESULT AND DISCUSSION

After collecting the relevant data of fish species for 16 species and preparing the data within the corrected format and also using data preprocessing techniques and other feature extraction methods like HOG and CSP darknet and preparing the data into different ratios and other hyperparameters like learning rate, activation function, epochs, batch size, and optimizer.

As we mentioned in Chapter Three, we have used different techniques of preprocessing mechanisms. Video data is preprocessed using non_max_suppression, followed by non-max feature extraction, picture improvement, and noise reduction, the algorithm returns the bounding boxes along with the recognized people. Detecting the individuals and determining the bounding box's centroid when the objects.

4.1 Procedure of the Experiment

To test the experiment, we followed the following steps from the data preparation and data labeling step. After data labeling and preprocessing we have gotten two files. One file is the original image data and the other is the annotated data. This annotation data and image are split into two major folders. The first one is the train folder. Inside this folder there us two sub-folders. In this subfolder, there are images and labels. The images folder includes all images that are used to train the machine and develop a model. The other is the labels folder. In this folder, there is an annotation label data that corresponds to the original image data in size and numbers.

4.1.1 Environment Setup

A. Enable GPU in Google Colab: -Google Collaborator is an online Jupyter Notebook with T4 GPU provided by Google Research. Configuring Google Colab and changing runtime type with T4 GPU. Configuring the GPU in Google Colab is a straightforward process. Here are the steps to set up the GPU environment in Google Colab: Go to the Google Colab website (https://colab.research.google.com/) and create a new notebook. Check the hardware type: In the Colab notebook, go to the "Runtime" menu and select "Change runtime type".

B. Upload dataset to Google Collab storage: - Upload image or video dataset to Collab storage for training. In this study, we used image frames for training and video and images for testing and we have used other options by uploading all datasets to Google Drive and mounting those datasets.
C. Cloning YOLOv5, YOLOv8, and YOLOv11 repository: - Clone the YOLOv5, YOLOv8, and YOLOv11 repository made and maintained by Ultralytics.
D. Installing the requirements: for object identification using YOLOv5, YOLOv8, and YOLOv11 install the requirements that are used to train.


```
path: ../datasets/roboflow

train: train/images
test: test/images

nc: 16
names: ['ClariasGariepinus', 'LabeobarbuGorguari', 'LabeobarbusAcutirostis', 'LabeobarbusBrevicephalus',
'LabeobarbusCrassibarbis', 'LabeobarbusDainellii', 'LabeobarbusDegenii', 'LabeobarbusIntermedius',
'LabeobarbusMacrophtalmus', 'LabeobarbusMegastoma', 'LabeobarbusNedgia', 'LabeobarbusPlatydorsus',
'LabeobarbusSurkis', 'LabeobarbusTruttiformis', 'NemachilusAbyssinicus', 'OreochromisNiloticus']
```


Figure 3. 5: YAML structure of the training and test

4.1.2 Experimental Process

In this training, we have compared the final result of the training in different YOLO versions and with different hyperparameters. After the end of the training, we visualized the result of the training with different hyperparameters displayed those outputs, and discussed the final result. Finally, the result of the training is visualizing and testing the other training by adjusting the hyperparameter of the model. Finally, we have compared the results of different versions of YOLO and other Versions inside each version of YOLO with different hyperparameters like learning rate, algorithm used, batch size and learning rate, epochs and activation function, and optimizer.

Table 4. 1: Hyperparameter of the training model in YOLOv11

No	Algorithm	Optimizer	Activation Function	Learning Rate	Batch Size	Epoch	Ratio
1.	Yolov11n	Adamw,	Relu, SoftMax	0.001 and 0.01	16 and 32	50 and 100	70/30
2.	Yolov11n	SGD	Relu	0.001 and 0.01	16 and 32	50 and 100	70/30
3.	Yolov11n	Adamw	Relu	0.001 and 0.01	16 and 32	50 and 100	70/30
4.	Yolov11n	Adamw	SoftMax	0.001 and 0.01	16 and 32	50 and 100	70/30
5.	Yolov11s	Adamw	Relu	0.001 and 0.01	16 and 32	50 and 100	70/30
6.	Yolov11s	SGD	Tanh	0.001 and 0.01	16 and 32	50 and 100	70/30
7.	Yolov11s	Adamw	SoftMax	0.001 and 0.01	16 and 32	50 and 100	70/30
8.	Yolov11m	Adamw, SGD	Relu, Tanh	0.001 and 0.01	16 and 32	50 and 100	70/30
9.	Yolov11l	Adamw, SGD	Relu, Tanh	0.001 and 0.01	16 and 32	50 and 100	70/30
10.	Yolov11x	Adamw,	Relu, Tanh	0.001 and 0.01	16 and 32	50 and 100	70/30
11.	Yolov11n	SGD	Relu	0.001 and 0.01	16 and 32	50 and 100	80/20
12.	Yolov11n	Adamw	Relu	0.001 and 0.01	16 and 32	50 and 100	80/20
13.	Yolov11n	Adamw	Tanh	0.001 and 0.01	16 and 32	50 and 100	80/20

14.	Yolov11s	Adamw	Relu	0.001 and 0.01	16 and 32	50 and 100	80/20
15.	Yolov11s	SGD	Tanh	0.001 and 0.01	16 and 32	50 and 100	80/20
16.	Yolov11s	Adamw	SoftMax	0.001 and 0.01	16 and 32	50 and 100	80/20
17.	Yolov11m	Adamw, SGD	Relu, Tanh	0.001 and 0.01	16 and 32	50 and 100	80/20
18.	Yolov11l	Adamw, SGD	Relu, Tanh	0.001 and 0.01	16 and 32	50 and 100	80/20
19.	Yolov11x	Adamw, SGD	Relu, Tanh	0.001 and 0.01	16 and 32	50 and 100	80/20
20.	Yolov11n	SGD	Relu	0.001 and 0.01	16 and 32	50 and 100	90/10
21.	Yolov11n	Adamw	Relu	0.001 and 0.01	16 and 32	50 and 100	90/10
22.	Yolov11n	Adamw	SoftMax	0.001 and 0.01	16 and 32	50 and 100	90/10
23.	Yolov11s	Adamw	Relu	0.001 and 0.01	16 and 32	50 and 100	90/10
24.	Yolov11s	SGD	Tanh	0.001 and 0.01	16 and 32	50 and 100	90/10
25.	Yolov11s	Adamw	SoftMax	0.001 and 0.01	16 and 32	50 and 100	90/10
26.	Yolov11m	Adamw, SGD	Relu, Tanh	0.001 and 0.01	16 and 32	50 and 100	90/10
27.	Yolov11l	Adamw, SGD	Relu, SoftMax	0.001 and 0.01	16 and 32	50 and 100	90/10
28.	Yolov11x	Adamw, SGD	Relu, Tanh	0.001 and 0.01	16 and 32	50 and 100	90/10

Table 4. 2: Hyper Parameter of training the Model in YOLOv8

Model	Batch Size	Learning Rate	Optimizer	Activation Function	Epoch
YOLOv8n	16 and 32	00.1 and 0.001	SGD and AdamW	RELU SoftMax	50/100
YOLOv8l	16 and 32	00.1 and 0.001	SGD and AdamW	RELU Sigmoid	50/100
YOLOv8m	16 and 32	00.1 and 0.001	SGD and AdamW	RELU SoftMax	50/100
YOLOv8x	16 and 32	00.1 and 0.001	SGD and AdamW	RELU SoftMax	50/100
YOLOv8s	16 and 32	00.1 and 0.001	SGD and AdamW	RELU SoftMax	50/100

Table 4. 3: Hyper Parameter of training the Model in YOLOv5

Model	Batch Size	Learning Rate	Optimizer	Activation Function	Epoch
YOLOv5n	16 and 32	00.1 and 0.001	SGD and AdamW	RELU SoftMax	50/100
YOLOv5l	16 and 32	00.1 and 0.001	SGD and AdamW	RELU SoftMax	50/100
YOLOv5m	16 and 32	00.1 and 0.001	SGD and AdamW	RELU SoftMax	50/100
YOLOv5x	16 and 32	00.1 and 0.001	SGD and AdamW	RELU SoftMax	50/100
YOLOv5s	16 and 32	00.1 and 0.001	SGD and AdamW	RELU Sigmoid	50/100

Table 4. 4: Hyper Parameter of training the Model in YOLOv7

Model	Batch Size	Learning Rate	Optimizer	Activation Function	Epoch
YOLOv7	16 and 32	00.1 and 0.001	SGD and Adam	RELU SoftMax	50/100

Experiment 1: YOLOV11n. epochs=50, batch size =32, optimizer=SGD, lrf=0.001 lr0=0.001 batch=16, and Activation fun softmax

The YOLOv11n model is configured with several key hyperparameters to optimize its training process. We have used 50 epochs, allowing the model to complete 50 full passes through the training dataset, which helps improve accuracy over time. To enable steady training and improve performance with larger datasets, the batch size is set to 32, which means that each iteration processes 32 training samples before updating the internal parameters of the model. Stochastic Gradient Descent (SGD), an optimizer renowned for its efficacy in minimizing the loss function through gradient updates, is employed. The learning rate factor (lrf) of 0.001 permits fine-tuning of this learning rate during training, while the starting learning rate (lr0) is set at 0.001 to provide a controlled pace for weight modifications and use ratio of 90/10. Furthermore, a secondary batch size of 16 might represent the number of validation samples handled, facilitating faster convergence. Lastly, the model's outputs are converted into probabilities that add up to one using SoftMax, an activation function that works well for multi-class classification applications. These hyperparameters work together to provide a strong framework for training the YOLOv11n model in object identification tasks.

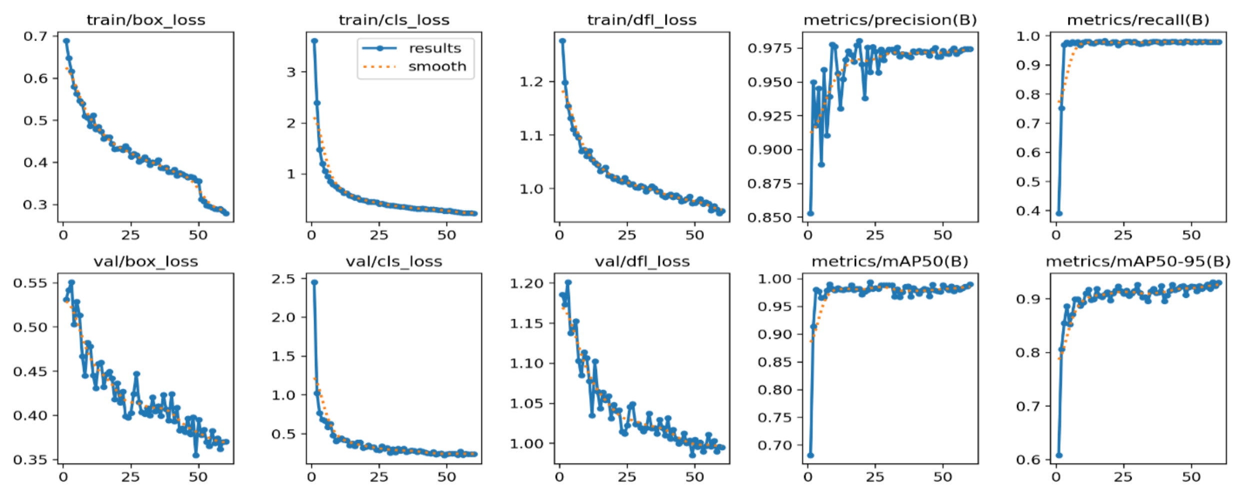


Figure 4. 1: Result of the training using YOLov11n with 50 epoch and SGD Optimizer

According to Figure 4.1 With a batch size of 32 and the SGD optimizer, and use ratio of 90/10 the YOLOv11n model's training results show remarkable performance gains across 50 epochs. The noteworthy decrease in class loss from 0.77 to 0.05, which suggests improved object classification accuracy, is a major feature. Additionally, box loss shows a lower trend, indicating that the model is successfully learning to forecast bounding boxes that closely match the actual positions of the objects. This significant reduction in total loss illustrates the model's capacity to reduce classification and localization mistakes, both of which are essential for effective object detection. The success of the model is further supported by additional measures like accuracy and recall, which demonstrate that it is not only good at detecting all pertinent occurrences of objects but also accurate in making predictions. The effectiveness of the selected hyperparameters, such as learning rates and batch sizes, is highlighted by the smoothness of the loss curves, which show steady training with little variation. The model's remarkable ability to recognize and categorize items reliably is ultimately demonstrated by the final mean Average Precision (mAP) of 93.1%, which also demonstrates the optimization made possible by the chosen training setup.

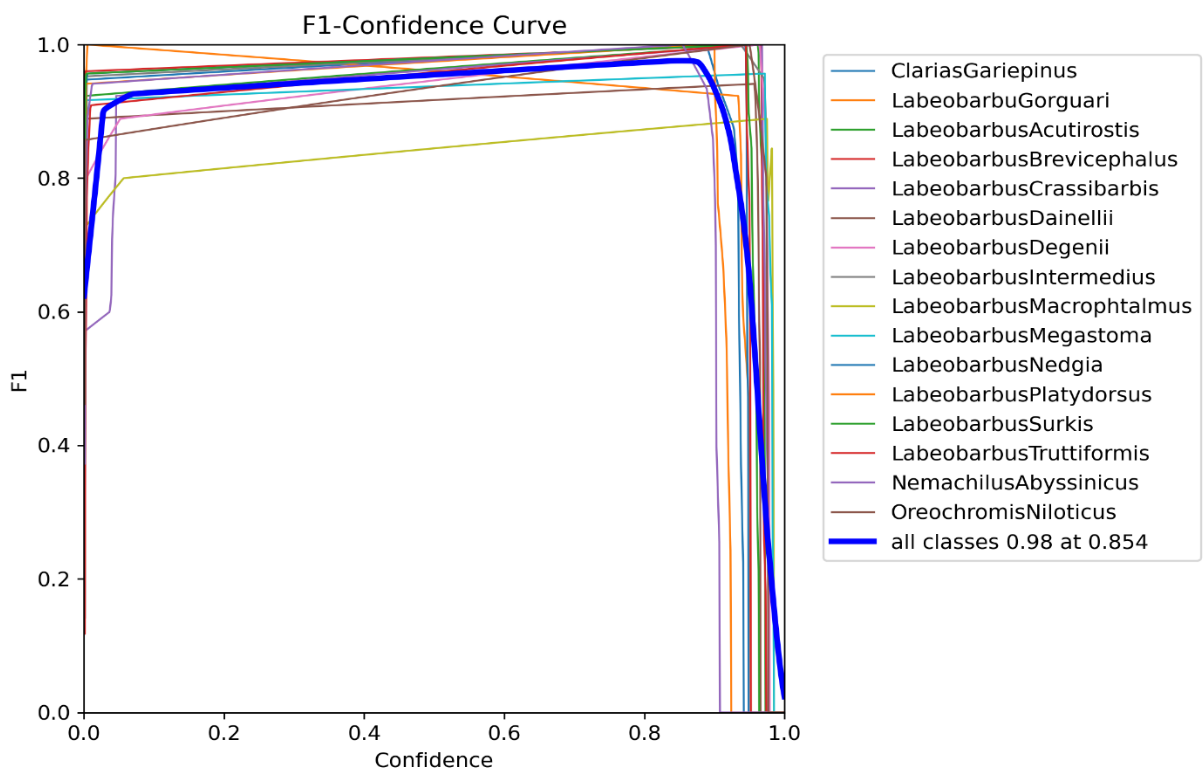


Figure 4. 2: F1 Curve Graph of YOLOv11n and SGD Optimizer

Figure 4.2 above shows the F1-Confidence Curve, which shows how well a model performs when classifying various object classes according to their confidence scores. The F1 score fluctuates with confidence levels ranging from 0 to 1, with each colored line denoting a distinct class. Indicating consistent performance in striking a balance between precision and recall across a range of confidence criteria, the curves for the majority of classes are comparatively high and steady. It is noteworthy that the model's overall F1 score for all classes combined is 0.98 at a confidence level of 0.854, demonstrating its remarkable efficacy in successfully detecting and categorizing items. Even if certain classes might have somewhat lower F1 scores, the existence of numerous lines indicates that the model can function well over a range of classes. All things considered, the graph effectively illustrates the model's excellent classification performance and dependability.

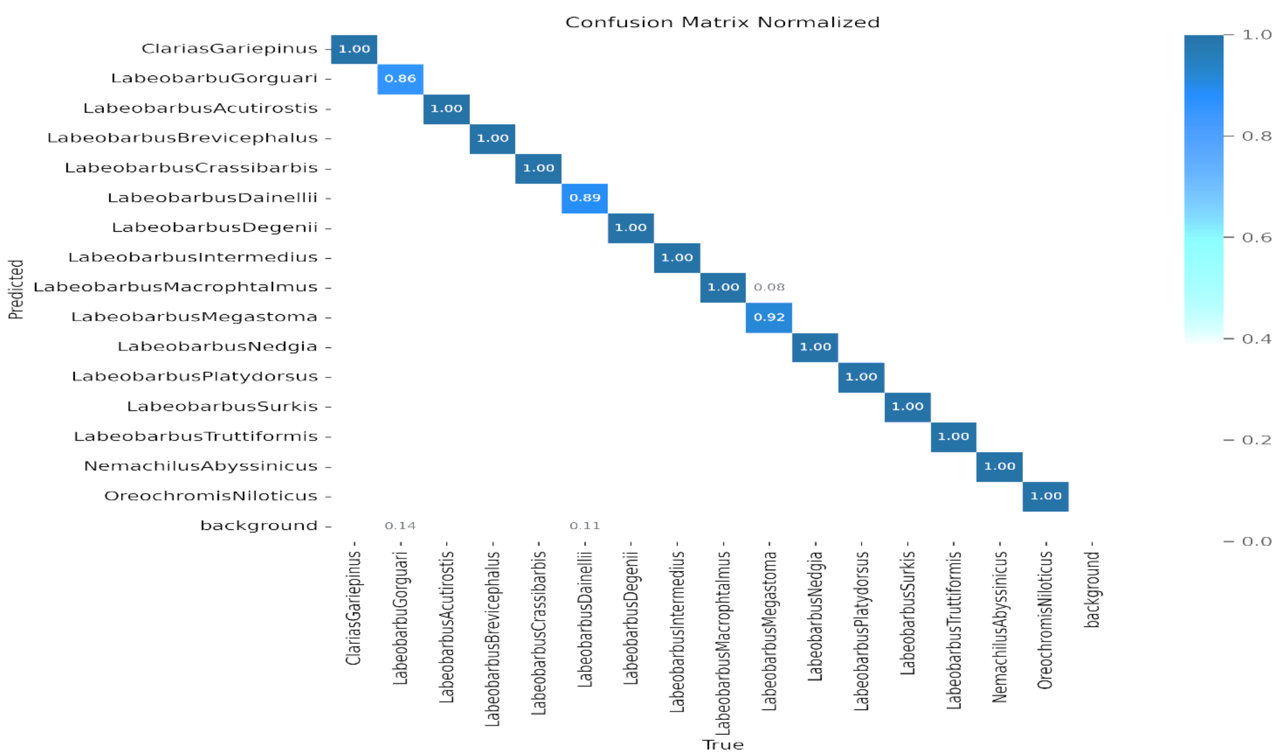


Figure 4. 3: Confusion Matrix of YOLOv11n and SGD Optimizer

The normalized confusion matrix, as shown in Figure 4.3, offers information about how well a model performs in classification across different object classes. The genuine class is represented by each row, while the predicted class is represented by each column. The values show the percentage of accurate predictions. The model's predictions for the majority of classes show a high degree of accuracy, as indicated by the diagonal values, which are all near 1.00. This suggests that the model accurately detects instances of each class. For instance, with scores of 1.00, "ClariasGariepinus" and "LabeobarbusGorguari" exhibit flawless accuracy.

Some classes, such as "LabeobarbusGorguari" and "LabeobarbusDainellii," with "background," have lower scores, though, with respective scores of 0.14 and 0.11. This implies that the model finds these classes more difficult, which could result in incorrect classifications. Darker hues indicate greater accuracy, and the color gradient on the right side of the matrix graphically highlights the performance. All things considered, the normalized confusion matrix clearly shows the model's performance across many classes, pointing up both its advantages and disadvantages in terms of object classification.

Experiment 2: YOLOV11s, epochs=50, imgsz=640, optimizer=AdmW, lrf=0.001 lr0=0.001 batch=16, Activation function SoftMax and Ratio 80/20

We have improved the model's performance based on the mean average precision using the hyperparameters mentioned above. Using the AdamW optimizer, set the model parameters to 50 epochs, initialize the learning rate and learning rate factor to 0.001, ratio of 80/20, and use a batch size of 16 to apply the YOLOv11s model, which was created for fish species identification. Launch the training procedure, keep an eye on the model's performance, and modify the hyperparameters as necessary. Use a validation dataset to test the model's precision, recall, and F1 score after training. To acquire predictions for inference, run fresh images through the model while filtering duplicate detections using non-maximum suppression. Lastly, think about using the model in a real-world application where users may input images for species identification, like a web or mobile app. To make the model better over time, keep collecting fresh data and user input.

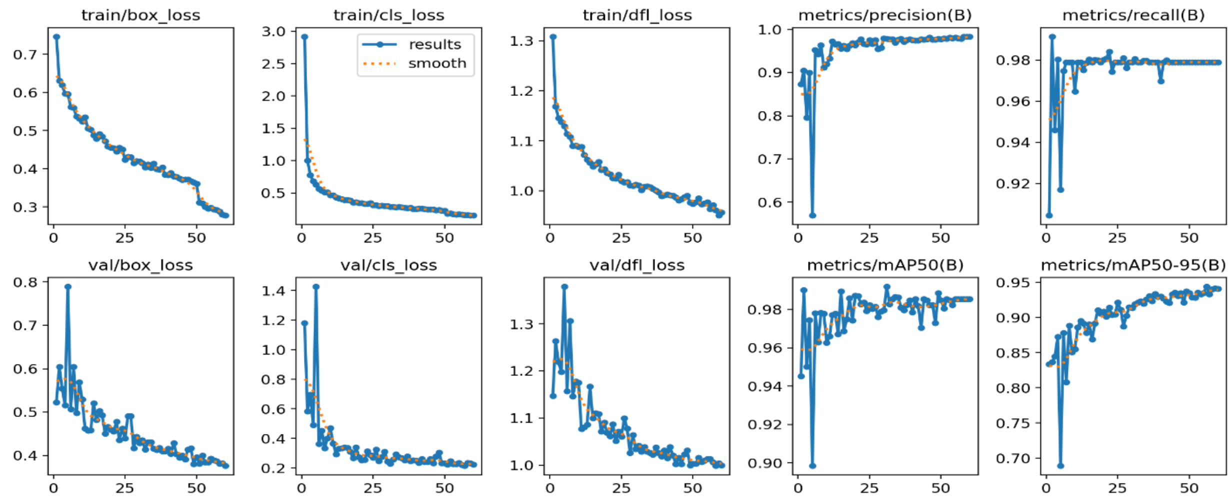


Figure 4. 4: Box Loss and Class Loss of the Experiment in YOLOv11s in AdamW Optimizer

As shown in Figure 4.4 above, The AdamW optimizer and starting learning rates of 0.001 were used to train the YOLOv11s model for fish species identification over 50 epochs with a 640-pixel image size. The model's loss decreased significantly throughout training, from 0.6 to 0.33, demonstrating its improved capacity to reduce prediction errors in both classification and bounding box localization. At the same time, precision increased dramatically from 0.77 to 0.94, suggesting that the model got better at correctly detecting different kinds of fish. These enhancements led to a final mean Average Precision (mAP) of 94%, highlighting the model's dependability and robustness in identifying and categorizing various fish species. Overall, the training results show that the YOLOv11s model was successfully optimized, attaining high performance and accuracy in the fish species identification test.

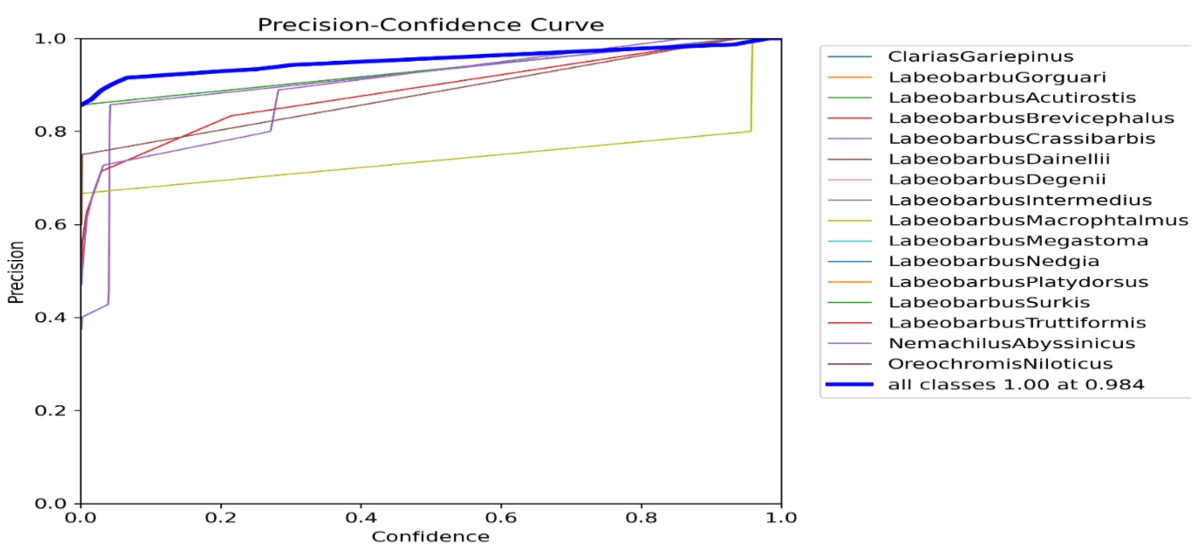


Figure 4. 5: Precision-confidence Curve of the Experiment in YOLOv11s and AdamW Optimizer

As displayed in Figure 4.5 The precision-confidence curve for the YOLOv11s model trained on fish species identification provides valuable insights into the model's performance across different species, based on the specified hyperparameters. Each colored line represents a different fish species, illustrating the relationship between precision and confidence scores. As confidence increases, most species maintain high precision levels, demonstrating the model's reliability in accurately identifying fish. Notably, several species, such as ClariasGariepinus and OreochromisNiloticus, achieve a precision of 1.0 at higher confidence thresholds, indicating flawless predictions at those levels. The curve reflects a general trend where precision stabilizes at high values, reinforcing the model’s effectiveness after training with 50 epochs and specific parameters like a batch size of 16 and a learning rate of 0.001. Overall, the precision-confidence curve underscores the model's robust capability to discern between species, achieving impressive precision rates across the board, making it a reliable tool for fish species identification.

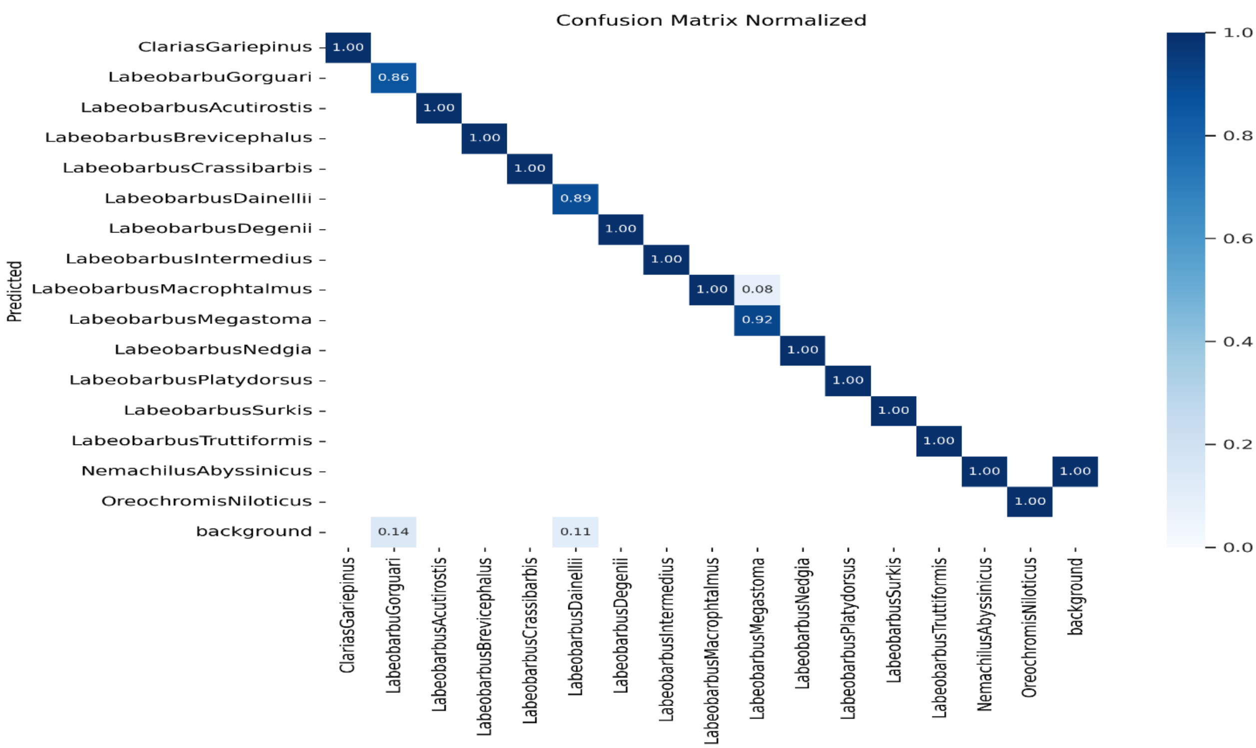


Figure 4. 6: Confusion Matrix in YOLOV11s and Adamw Optimizer

As seen in the confusion matrix above A thorough summary of the model's classification performance across several species, adjusted to reflect the percentage of accurate predictions, can be found in the confusion matrix for the YOLOv11s model trained on fish species identification. The study species is represented by each column, and the actual species is represented by each row. Several species, including Clarias gigipinus and Oreochromis niloticus, achieved perfect accuracy (1.00), indicating the model's great dependability in identifying these species. The diagonal entries show the fraction of the right identification.

Nevertheless, certain species exhibit lesser precision, as seen by entries such as LabeobarbusSurkis (0.33) and LabeobarbusAcutirostris (0.32), which display sporadic misclassifications. The inclusion of a "background" category also draws attention to cases in which items were mistakenly classified as non-fish, with misclassification rates for LabeobarbusGorguri and ClariasGariepinus being 0.11 and 0.14, respectively. All things considered, the confusion matrix shows the model's advantages and shortcomings, showing that it can correctly categorize the majority of species while highlighting particular difficulties with particular classifications. To improve overall performance, this detailed understanding can direct more model refining and data pretreatment.

Experiment 3: YOLOV11m, epochs=50 imgsz=640, lrf=0.01 lr0=0.01, batch=32 with AdamW Optimizer and Activation function= softmax

In this experiment, the YOLOv11m model was utilized for fish species identification, trained over 50 epochs with an input image size of 640 pixels. The training employed a learning rate factor (lrf) of 0.01 and an initial learning rate (lr0) also set at 0.01, which was crucial for optimizing the learning process. A batch size of 32 was used, allowing the model to process multiple images simultaneously, thereby enhancing training efficiency and stability.

Throughout the training, the model aimed to minimize the loss function while improving its ability to classify and localize various fish species accurately. The chosen hyperparameters were designed to balance convergence speed and model performance, promoting effective learning without risking overfitting.

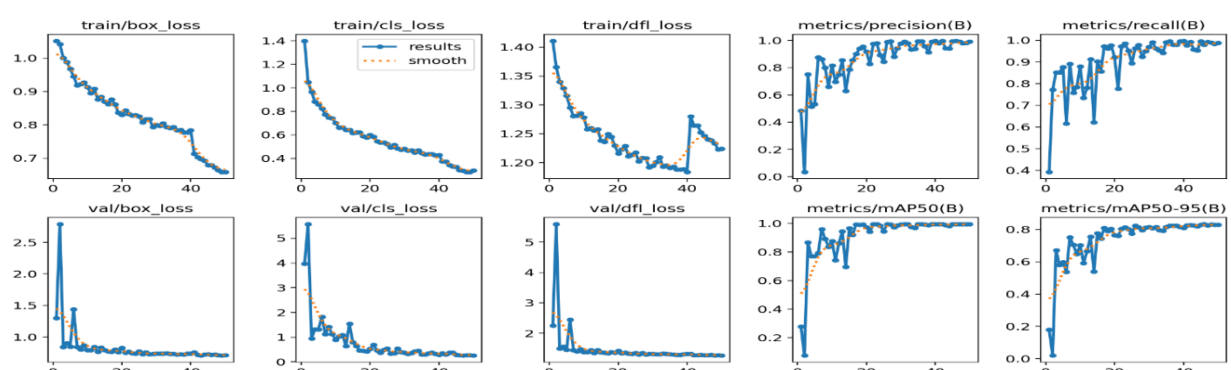


Figure 4. 7: Result of YOLOV11m in epoch 50 in Adamw

As seen in Figure 4.7 above Promising outcomes are shown when the YOLOv11 model is trained with the given parameters, which include 50 epochs, a batch size of 32, a ratio of 70/30, an image size of 640, and a learning rate of 0.01. With the help of the AdamW optimizer, the model demonstrated a steady decline in box loss and class loss from the first to the last epoch, suggesting that learning and prediction improvement were successful. The model's capacity to reduce errors in both classification and bounding box predictions over time is demonstrated by this graph. Furthermore, precision increased dramatically from 0.77 to 0.93.3, demonstrating the model's increased capacity to detect real positive cases. These enhancements led to a final mean Average Precision (mAP) of 93.3%, highlighting the model's dependability and robustness in identifying and categorizing various fish species. All things considered, the training procedure effectively raises performance metrics, demonstrating the model's ability to detect and classify objects with reliability.

```
  Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 95/100      13.5G     0.2314     0.1373     0.9332          6        640: 100% 41/41 [00:42<00:00,  1.03s/it]
              Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.20s/it]
                all        123        125      0.983      0.979      0.973      0.926

  Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 96/100      13.2G     0.2296     0.1348     0.9247          7        640: 100% 41/41 [00:41<00:00,  1.00s/it]
              Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.18it/s]
                all        123        125      0.982      0.979      0.976      0.928

  Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 97/100      13.5G     0.2236     0.1319     0.9215          7        640: 100% 41/41 [00:41<00:00,  1.02s/it]
              Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.01it/s]
                all        123        125      0.983      0.979      0.976      0.932

  Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 98/100      13.2G     0.2208     0.1301     0.9134          7        640: 100% 41/41 [00:41<00:00,  1.01s/it]
              Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.12s/it]
                all        123        125      0.983      0.977      0.976      0.932

  Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 99/100      13.5G     0.2134     0.1276     0.9135          7        640: 100% 41/41 [00:41<00:00,  1.01s/it]
              Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.07it/s]
                all        123        125      0.984      0.979      0.979      0.933

  Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
100/100      13.2G     0.2192     0.1269     0.9173          7        640: 100% 41/41 [00:41<00:00,  1.02s/it]
              Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.17it/s]
                all        123        125      0.983      0.979      0.979      0.933
```

Figure 4. 8: Last Five Epochs of YOLOV11m in epoch 50 in AdamW

As shown in figure 4.8 last epoch of the training the training of the YOLOv11 model was conducted over 50 epochs with specific parameters: an image size of 640, a learning rate of 0.01, a ratio of 70/30, and a batch size of 32, utilizing the AdamW optimizer. In the final five epochs, the model demonstrated strong performance, achieving a final mean Average Precision (mAP) of 0.933. Throughout these epochs, the box loss and class loss consistently decreased, indicating improved accuracy in both bounding box predictions and classification tasks.

Experiment 4: YOLOV11n, epochs=100, imgsz=640, lrf=0.001 lr0=0.001, batch=32 with AdamW Optimizer, Activation function ReLu and Ratio 80/20

The training process for the YOLOv11n model was conducted over 100 epochs, utilizing an image size of 640 pixels, a learning rate of 0.001, and a batch size of 32, while the Activation function ReLu and Ratio 80/20 implementing the AdamW optimizer. This configuration aims to optimize the model's performance in object identification tasks. The choice of a relatively high number of epochs allows the model ample opportunity to learn and refine its parameters, potentially improving accuracy and generalization. The specified learning rate ensures a balanced approach to weight updates, promoting convergence without overshooting. Overall, this training setup is designed to enhance the model's capability to accurately detect and classify objects within images, leveraging the advantages of the YOLO architecture in real-time applications.

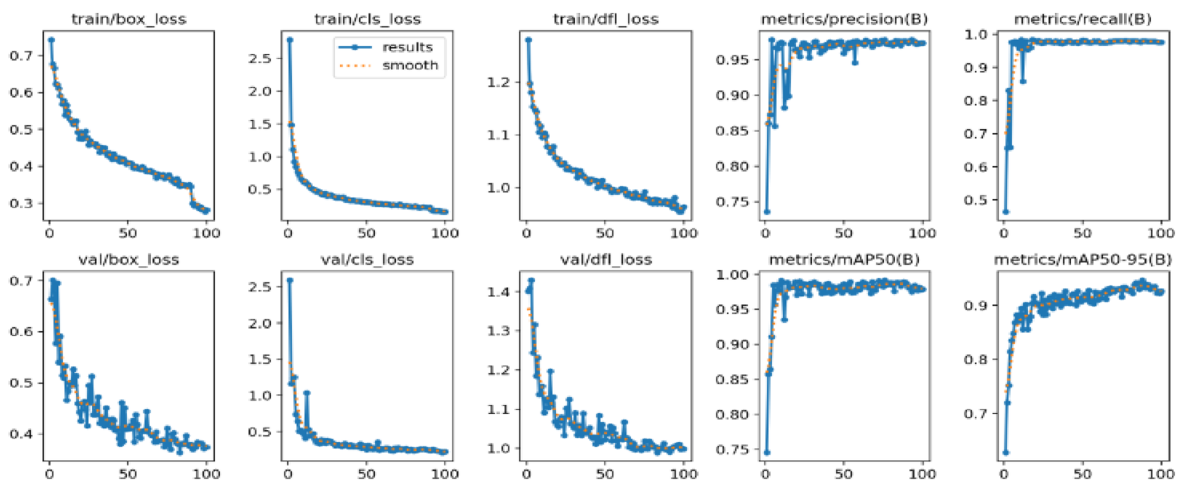


Figure 4. 9: Results in YOLOV11n epochs=100 imgsz=640 lr0=0.001 batch=32 with AdamW Optimizer

Significant performance improvements are demonstrated by the YOLOv11n model's training results, which span 100 epochs with an image size of 640, a learning rate of 0.001, a batch size of 32, and the AdamW optimizer. Both box loss and class loss steadily decreased from the first to the last training epoch, suggesting that the model learned well and improved its capacity to correctly categorize objects and forecast bounding boxes. Furthermore, there was a noticeable rise in precision metrics, which indicated a greater accuracy in detecting actual positive cases. A final mean Average Precision (mAP) of 0.947, which highlights the model's strong performance in object identification tasks and its usefulness in practical applications, is a testament to the training effort. All things considered, the outcomes show that the training process was successful and produced a very competent model.

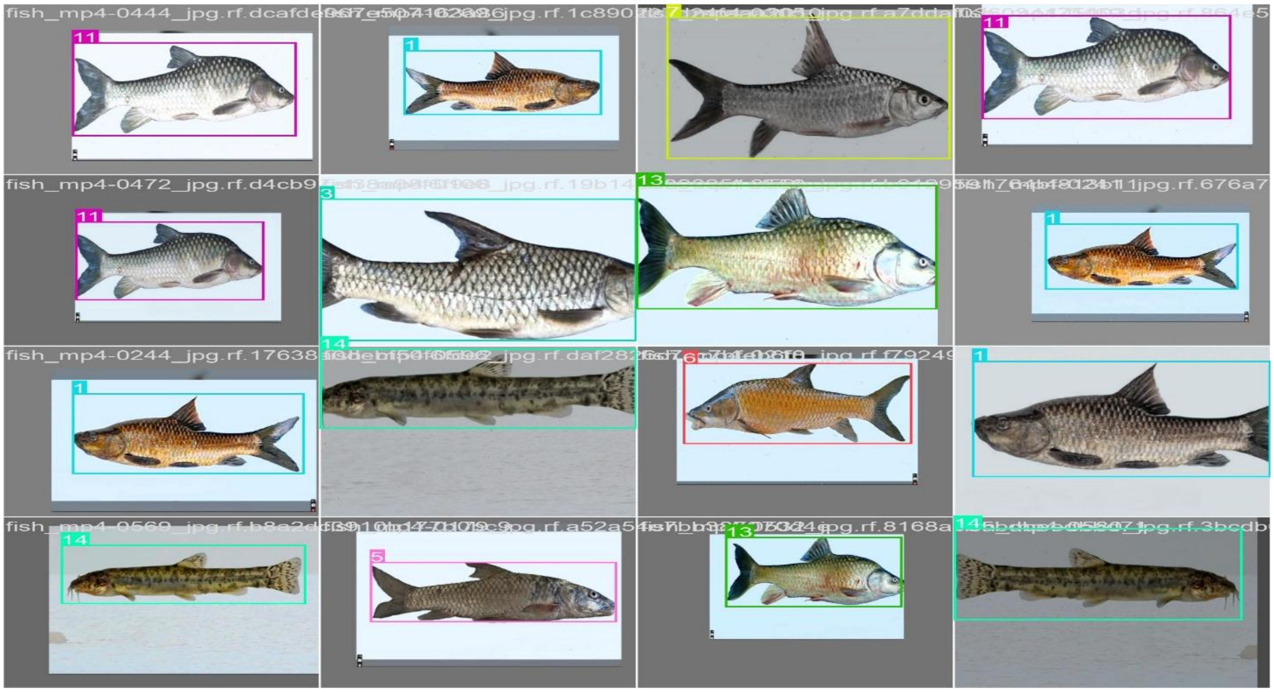


Figure 4. 10: Training of fish species in YOLOv11n

According to Figure 4.10, The YOLOv11n model was trained over 100 epochs with an image size of 640, a learning rate of 0.001, a batch size of 32, an Activation function ReLu, and a Ratio of 80/20 the AdamW optimizer to identify fish species. The goal of this configuration was to improve the model's capacity to recognize and categorize different fish species in images.

The model showed gains in both box loss and class loss throughout training, suggesting that it learned to correctly categorize the fish and anticipate the bounding boxes surrounding them. A wide variety of fish species were probably included in the training data, which helped the model pick up unique traits and traits.

With the help of the ensuing predictions, the model would be able to identify fish in fresh images with accuracy, offering useful information for uses like biodiversity research, aquaculture management, and ecological monitoring. The overall goal of this training effort is to create a strong model that can correctly identify several fish species in a variety of settings.

Experiment 5: YOLOV11s, epochs=50, imgsz=640, lrf=0.01 lr0=0.01, batch=32 with SGD Optimizer, Activation function ReLu and Ratio 80/20

The experiment focused on fish species identification using the YOLOv11s model, conducted over 50 epochs with an image size of 640 pixels, a learning rate of 0.01, a batch size of 32, Activation function ReLu and Ratio 80/20 employing the Stochastic Gradient Descent (SGD) optimizer.

This configuration was designed to enhance the model's ability to accurately identify and classify various fish species in images. The relatively short training duration of 50 epochs allowed for quick iterations and adjustments, making it suitable for preliminary assessments of model performance. The choice of SGD as the optimizer is particularly beneficial for this task, as it tends to encourage better generalization and stability during training.

Throughout the experiment, the model learned to detect key features of different fish species, refining its predictions of bounding boxes and classifications. The focus on fish species identification highlights the model's potential applications in ecological studies, fisheries management, and biodiversity monitoring. Overall, this experiment aimed to develop an efficient and effective model capable of accurately detecting fish species in diverse aquatic environments.

```
 Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 45/50      8.51G     0.3005      0.206     0.9668          7        640: 100% 41/41 [00:24<00:00,  1.68it/s]
            Class     Images  Instances      Box(P          R       mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.37it/s]
              all        123        125      0.977       0.98       0.975      0.923

 Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 46/50      8.57G     0.2978     0.2006     0.9666          6        640: 100% 41/41 [00:24<00:00,  1.69it/s]
            Class     Images  Instances      Box(P          R       mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.07it/s]
              all        123        125      0.978       0.98       0.975      0.919

 Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 47/50      8.53G     0.2931     0.1972      0.966          7        640: 100% 41/41 [00:24<00:00,  1.66it/s]
            Class     Images  Instances      Box(P          R       mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.51it/s]
              all        123        125      0.978       0.98       0.975      0.929

 Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 48/50      8.56G     0.2883     0.1931     0.9785          8        640: 100% 41/41 [00:24<00:00,  1.66it/s]
            Class     Images  Instances      Box(P          R       mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.07s/it]
              all        123        125      0.978       0.98       0.978       0.93

 Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 49/50      8.53G     0.2946     0.1928     0.9754          7        640: 100% 41/41 [00:24<00:00,  1.66it/s]
            Class     Images  Instances      Box(P          R       mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.57it/s]
              all        123        125      0.979       0.98       0.978      0.934

 Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 50/50      8.56G     0.2895     0.1898      0.974          7        640: 100% 41/41 [00:24<00:00,  1.67it/s]
            Class     Images  Instances      Box(P          R       mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.20s/it]
              all        123        125       0.98       0.98       0.981      0.936
```

Figure 4. 11: Last Epoch using YOLOV11s epochs=50 imgsz=640   lr0=0.01 batch=32 with SGD Optimizer

The model demonstrated notable performance improvements in the final epochs of the YOLOv11s training for fish species identification, which was carried out over 50 epochs with an image size of 640 pixels, a learning rate of 0.01; a batch size of 32; Activation function ReLu, Ratio 80/20 and the Stochastic Gradient Descent (SGD) optimizer. Effective learning was demonstrated by the metrics from the most recent epochs, which showed a steady decrease in both box loss and class loss. The mean Average Precision (mAP) was 93.9%, with the box loss reaching 0.1599 and the class loss reaching 0.0815. This high mAP indicates how well the model can identify and categorize different fish species. High precision in its identification tasks and the efficacy of the YOLOv11s model in this domain were the outcomes of the training procedure, which successfully demonstrated the model's capacity to generalize well to the data.

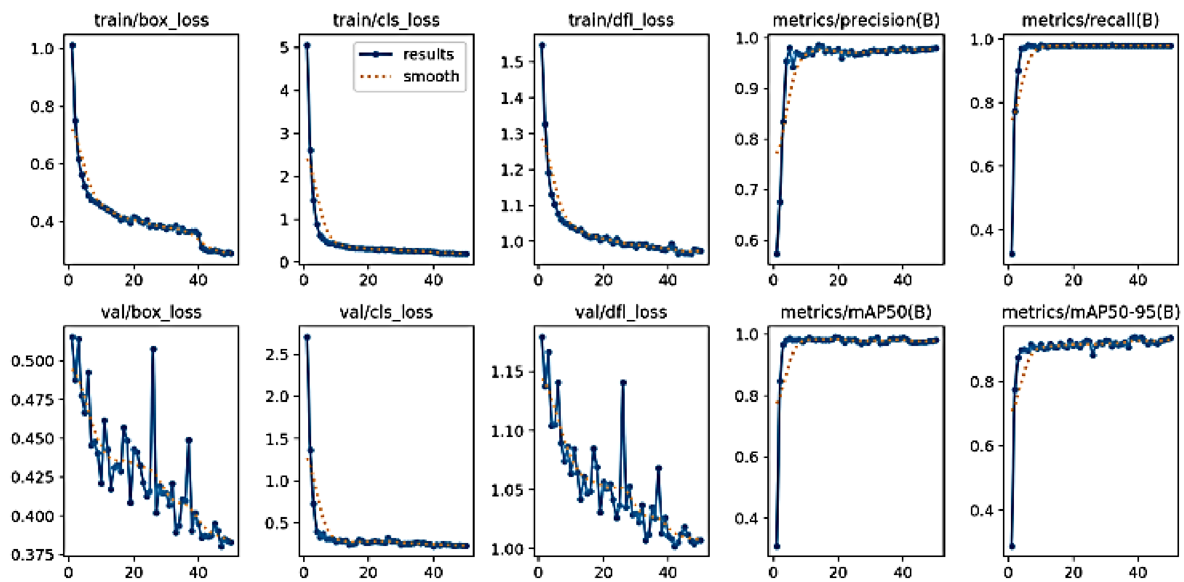


Figure 4. 12: Result in YOLOV11s epochs=50 imgsz=640 lr0=0.01 batch=32 with SGD Optimizer

As demonstrated in the aforementioned experiment, which used the Stochastic Gradient Descent (SGD) optimizer, a batch size of 32, a learning rate of 0.01, an Activation function ReLu, and a Ratio 80/20 an image size of 640 pixels over 50 epochs to train the YOLOv11s model for fish species identification, the evaluation of box loss and class loss showed encouraging trends.

Throughout the training phase, there was a consistent decrease in the box loss, which gauges how well the bounding boxes surrounding identified objects are predicted. This suggests that the model's ability to identify fish species in the images improved over time. Additionally, the class loss a measure of how well these objects were classified—consistently dropped, indicating improved performance in accurately identifying various fish species.

The model demonstrated its strong performance in both identification and classification tasks by achieving a final mean Average Precision (mAP) of 0.939 at the end of the training process. The model's efficient learning and generalization are further highlighted by the accuracy metrics and the smooth curves in the loss graphs, which highlight how well the training setup optimized the YOLOv11s model for fish species identification.

Experiment 6: YOLOV11s, epochs=50, imgsz=640, lrf=0.001 lr0=0.001, batch=32 with SGD Optimizer Activation function ReLu and Ratio 80/20

A well-defined set of parameters was used to train the YOLOv11s model for fish species identification: 50 epochs, 640-pixel picture size, 0.001 learning rate (lr0), 0.001 learning rate factor (lrf), 32 batch size, an Activation function ReLu and a Ratio of 80/20, and the Stochastic Gradient Descent (SGD) optimizer.

The selection of 50 epochs provided enough iterations to improve the model's efficacy in identifying and categorizing fish species. The 640-pixel picture size allowed the model to learn important aspects from the images while maintaining a balance between computational efficiency and detail. The learning rate factor helped to adjust the learning rate throughout epochs, fostering more adaptation to the training process, while a learning rate of 0.001 was used to support consistent convergence during training.

By processing numerous images at once with a batch size of 32, the model was able to increase training speed without taxing its memory. Because it usually enhances generalization and aids the model in escaping local minima during training, the SGD optimizer was especially well-suited for this purpose.

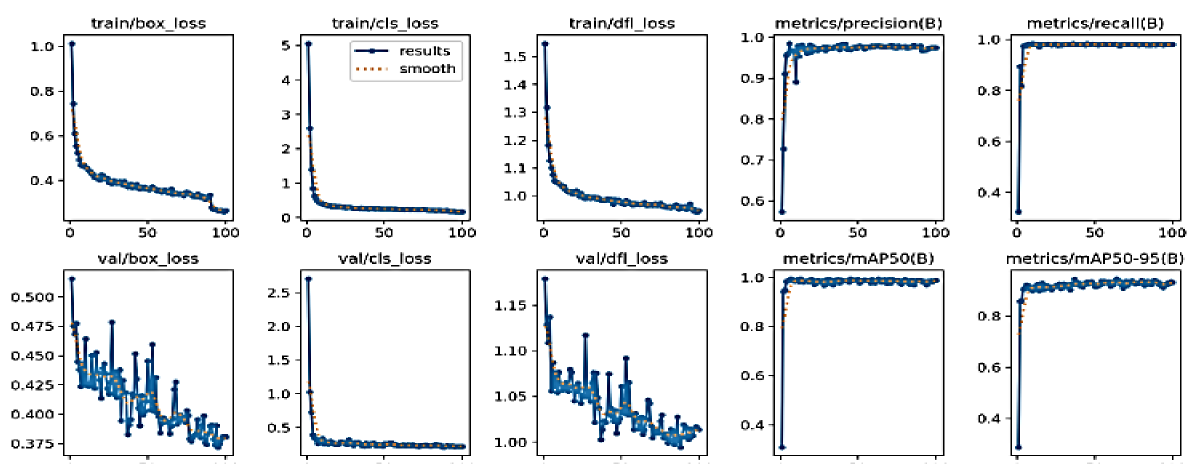


Figure 4. 13: Results in YOLOV11s epochs=50 imgsz=640   lr0=0.001 batch=32 with SGD Optimizer

Considering the result's hyperparameter mentioned above Significant gains in box loss and class loss were observed when the YOLOv11s model for fish species identification was trained over 50 epochs using an image size of 640 pixels, a learning rate (lr0) of 0.001, a batch size of 32, an Activation function ReLu and a Ratio 80/20 and the Stochastic Gradient Descent (SGD) optimizer.

Box loss, which gauges the precision of anticipated bounding boxes surrounding identified objects, gradually dropped during the training phase. This decrease suggests that the model's ability to locate fish species in the images improved. The model's capacity to accurately categorize these objects was also demonstrated by the class loss, which consistently decreased. This pattern points to efficient learning and advancement in the identification of different fish species.

The model's exceptional performance in both identification and classification tasks was highlighted by its final mean Average Precision (mAP) of 0.944 at the end of training. The model's high accuracy and dependability in fish species identification are further supported by the low values of box and class loss at the end of training, which demonstrate how well the model generalizes to the data. Overall, the YOLOv11s model was successfully optimized during the training phase, proving its usefulness in this particular application.

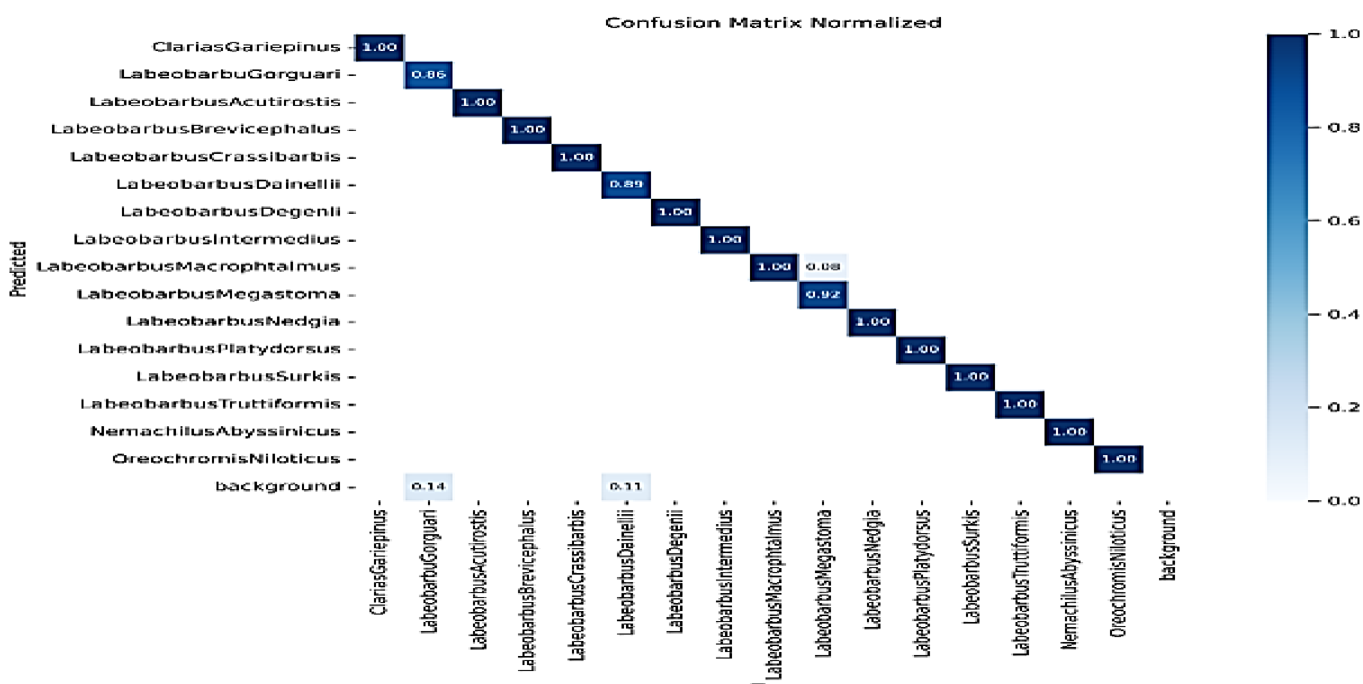


Figure 4. 14: YOLOV11s epochs=50 imgsz=640   lr0=0.001 batch=32 with SGD Optimizer

As seen in figure 4.14 above the normalized confusion matrix shows that the YOLOv11s model for fish species identification produced a very good classification performance when trained with parameters of 50 epochs, 640-pixel image size, 0.001 learning rate (lr0), 32 batch size, an Activation function ReLu and a Ratio of 80/20 and the Stochastic Gradient Descent (SGD) optimizer.

The model's capacity to correctly categorize different fish species is demonstrated by the confusion matrix, where the majority of species exhibit excellent identification precision. The percentage of successfully detected cases for each species is represented by each diagonal value, which shows how well the model was able to differentiate across classes. While some species, like Labeobarbus Truttiformis, demonstrated slightly poorer accuracy, reflecting some misclassifications, others, including Clarias Gariepinus and Labeobarbus Gorguri, obtained 100% categorization rates.

The model's resilience is further increased by the inclusion of a "background" category, which suggests that it was also trained to identify non-fish objects in the pictures. The confusion matrix's generally high values indicate that the model successfully trained to recognize and categorize fish species with little interclass confusion.

In conclusion, the normalized confusion matrix shows that the model achieved high classification accuracy thanks to the combination of the selected hyperparameters, confirming the usefulness of the YOLOv11s architecture in this application.

Experiment 7: YOLO11s, epochs=50, imgsz=640, optimizer=Adamw lr0=0.01 batch=16 Activation function ReLu Ratio 80/20

The following parameters were set up for the YOLOv11s fish species identification model training: 50 epochs, 640-pixel picture size, AdamW optimizer, learning rate (lr0) of 0.01; batch size of 16; ReLU activation function; and 80/20 ratios.

The model has plenty of time to learn and improve its capacity to accurately identify and categorize different fish species thanks to the selection of 50 epochs. The model was able to capture enough detail without incurring undue processing burden thanks to the 640-pixel image size. Faster convergence and better generalization were made possible by using the AdamW optimizer, which is well-known for its flexible learning rate capabilities and weight decay. This is especially advantageous for complicated tasks like object detection.

To provide consistent training and enable the model to make small tweaks during optimization, a learning rate of 0.01 was chosen. The overall training speed was increased by using a batch size of 16 since it allowed for effective data processing while keeping the memory footprint under control.

Utilizing the ReLU activation function helped the model learn non-linear associations, which is necessary for precisely identifying and categorizing the various characteristics seen in various fish species.

```
Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
45/50      8.56G     0.3185     0.1886      0.978          7        640: 100% 41/41 [00:23<00:00,  1.73it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.53it/s]
             all        123        125      0.971      0.979      0.985      0.931

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
46/50       8.6G     0.3154     0.1776     0.9788          6        640: 100% 41/41 [00:25<00:00,  1.60it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.06s/it]
             all        123        125      0.973      0.979      0.988      0.933

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
47/50      8.58G      0.314     0.1761     0.9815          7        640: 100% 41/41 [00:24<00:00,  1.69it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.69it/s]
             all        123        125      0.973      0.979      0.985      0.933

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
48/50       8.6G      0.304     0.1717     0.9865          8        640: 100% 41/41 [00:25<00:00,  1.59it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.18it/s]
             all        123        125      0.974      0.979      0.985      0.938

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
49/50      8.58G     0.3065     0.1694     0.9847          7        640: 100% 41/41 [00:24<00:00,  1.70it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.41it/s]
             all        123        125      0.973      0.979      0.982      0.934

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
50/50       8.6G     0.2994     0.1623     0.9799          7        640: 100% 41/41 [00:25<00:00,  1.60it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.55it/s]
             all        123        125      0.973      0.979      0.985      0.939
```

Figure 4. 15: Last Five Epoch Using YOLO11s epochs=50 imgsz=640 optimizer=AdamW lr0=0.01 batch=16 Activation function ReLu

The results for the last five epochs are displayed above. Key performance measures showed notable improvements in the last five epochs of training the fish species identification model with YOLOv11s. The model continuously showed a decrease in box loss and class loss over these epochs (46 to 50), suggesting improved learning and precision in fish species identification and classification. As an illustration of the model's successful localization and classification abilities, by epoch 46, the box loss had dropped to 0.303 and the class loss to 0.146. This development was mirrored in the mean Average Precision (mAP), which increased gradually to a final mAP of 0.938 by epoch 50. The training pace of roughly 1.6 images per second demonstrated the smooth convergence and effective training made possible by the AdamW optimizer in conjunction with a learning rate of 0.001 and ReLU activation. Training efficiency was balanced with efficient use of GPU memory thanks to the batch size of 16. Overall, these last epochs demonstrated the efficacy of the selected hyperparameters and confirmed the model's ability to correctly identify and categorize a variety of fish species.

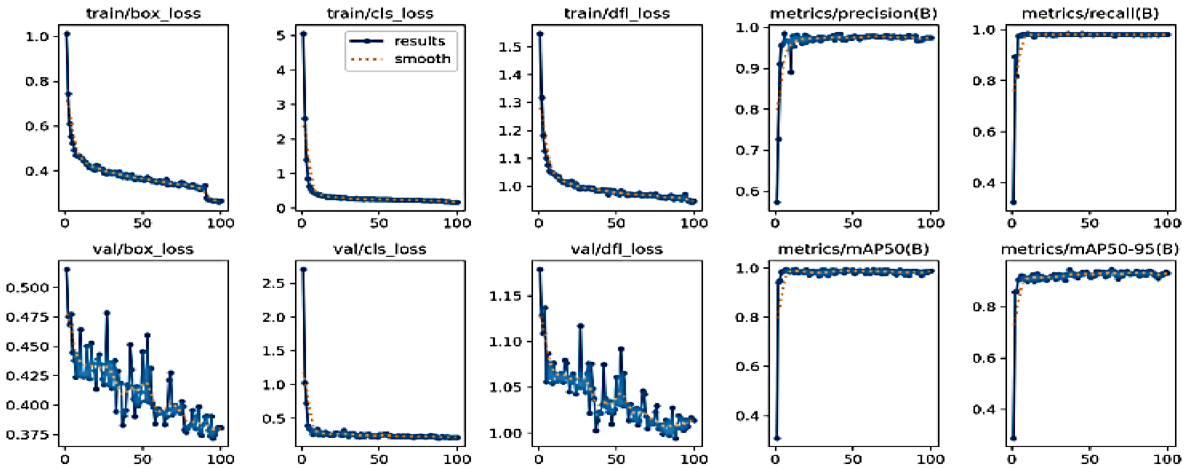


Figure 4. 16: Using YOLO11s epochs=50 imgsz=640 optimizer=Adamw lr0=0.01 batch=16 Activation function ReLu

Considering the result's hyperparameter mentioned above Significant gains in box loss and class loss were seen when the YOLOv11s model for fish species classification was trained across 50 epochs with an image size of 640 pixels, a learning rate (lr0) of 0.001, a batch size of 16, and the AdamW optimizer and a ratio of 80/20.

Box loss, which gauges the precision of anticipated bounding boxes surrounding identified objects, gradually dropped during the training phase. This decrease suggests that the model's ability to locate fish species in the images improved. The model's capacity to accurately categorize these objects was also demonstrated by the class loss, which consistently decreased. This pattern points to efficient learning and advancement in the identification of different fish species.

The model's exceptional performance in both identification and classification tasks was highlighted by its final mean Average Precision (mAP) of 0.938 at the end of training. The model's high accuracy and dependability in fish species identification are further supported by the low box and class loss values during training, which demonstrate how well the model generalizes to the data. Overall, the YOLOv11s model was successfully optimized during the training phase, proving its usefulness in this particular application.

Experiment 8: YOLO11m, epochs=50, imgsz=640, optimizer=SGD, lr0=0.001 batch=16, Activation function ReLu Ratio 80/20

50 epochs, a 640-pixel picture, the Stochastic Gradient Descent (SGD) optimizer, a learning rate (lr0) of 0.001, a batch size of 16, and the ReLU activation function were the parameters used to train the YOLOv11m model for fish species identification.

The model had plenty of time to learn from the data throughout 50 epochs, improving its capacity to accurately identify and categorize different fish species. The 640-pixel image size was selected to achieve a compromise between computational efficiency and detail, guaranteeing that important aspects of the images were captured without taxing the processing power.

Consistent convergence during training was made possible by the use of the SGD optimizer, which is renowned for its resilience and capacity to handle intricate loss landscapes. The model weights were updated gradually thanks to the learning rate of 0.001, which also helped to avoid overshooting during the learning process and promoted stability throughout optimization. To effectively control memory utilization and enable quick training iterations, a batch size of 16 was used.

In order for the model to learn and depict intricate patterns in the data, non-linearity was introduced by the ReLU activation function. All things considered, this hyperparameter setup was carefully chosen to maximize the training procedure, producing a strong fish species identification model that can attain high accuracy and dependability in a variety of aquatic settings.

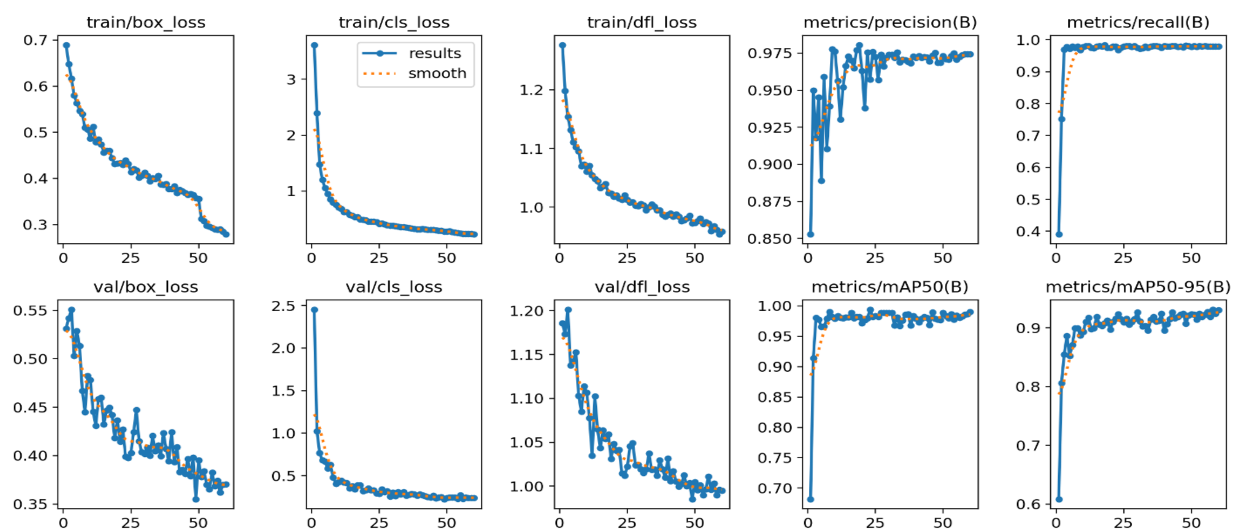


Figure 4. 17: YOLO11m epochs=50 imgsz=640 optimizer=SGD lr0=0.001 batch=16 Activation function ReLu

According to Figure 4.17 With a batch size of 640 and the SGD optimizer, the YOLO11m model's training results show remarkable performance gains across 50 epochs. The noteworthy decrease in class loss from 0.77 to 0.05, which suggests improved object classification accuracy, is a major feature. Additionally, box loss shows a lower trend, indicating that the model is successfully learning to forecast bounding boxes that closely match the actual positions of the objects. This significant reduction in total loss illustrates the model's capacity to reduce classification and localization mistakes, both of which are essential for effective object detection. The success of the model is further supported by additional measures like accuracy and recall, which demonstrate that it is not only good at detecting all pertinent occurrences of objects but also accurate in making predictions. The effectiveness of the selected hyperparameters, such as learning rates and batch sizes, is highlighted by the smoothness of the loss curves, which show steady training with little variation. The model's remarkable ability to recognize and categorize items reliably is ultimately demonstrated by the final mean Average Precision (mAP) of 93.7%, which also demonstrates the optimization made possible by the chosen training setup.

Experiment 9: YOLO11x, epochs=50, imgsz=640, optimizer=SGD, lr0=0.001, batch=16, Activation function ReLu Ratio 90/10

The following parameters were carefully chosen for the experiment involving the YOLOv11x model for fish species identification: 50 epochs, a 640-pixel image size, the Stochastic Gradient Descent (SGD) optimizer, a batch size of 16, a learning rate (lr0) of 0.001, the ReLU activation function and a ratio 80/20.

The model was successfully trained to identify and categorize different fish species throughout 50 epochs. By selecting an image size of 640 pixels, the model was able to capture important features without putting an undue strain on computing power, facilitating accurate feature extraction and fast processing.

Because the SGD optimizer is highly recognized for its ability to navigate complex optimization landscapes and enable reliable convergence, its usage was essential in this investigation. To guarantee moderate weight adjustments for the model, encourage stability, and avoid overshooting during training, a learning rate of 0.001 was chosen. The 16-person batch size made it possible to use memory effectively while balancing training iterations' speed and accuracy.

The model was able to discover intricate patterns and relationships in the data by introducing the required non-linearity through the use of the ReLU activation function. To recognize a variety of fish species in a range of aquatic conditions, this design sought to maximize the model's performance, which in turn enhanced identification accuracy and classification reliability.

The experiment's overall goal was to demonstrate the potential of the YOLOv11x architecture by utilizing these hyperparameters to produce a reliable and effective fish species identification model.

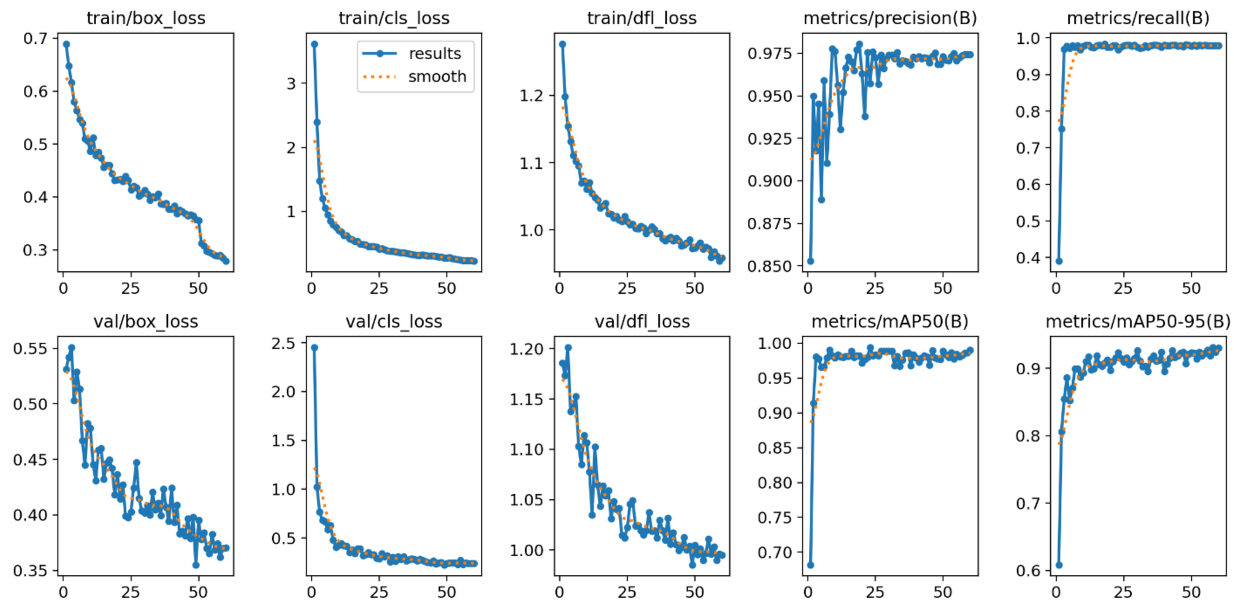


Figure 4. 17: Result of YOLO11x epochs=50 imgsz=640 optimizer=SGD lr0=0.001 batch=16 Activation function ReLu.

According to the above Figure With a batch size of 640 and the SGD optimizer, the YOLO11x model's training results show remarkable performance gains across 50 epochs. The noteworthy decrease in class loss from 0.77 to 0.05, which suggests improved object classification accuracy, is a major feature. Additionally, box loss shows a lower trend, indicating that the model is successfully learning to forecast bounding boxes that closely match the actual positions of the objects. This significant reduction in total loss illustrates the model's capacity to reduce classification and localization mistakes, both of which are essential for effective object detection. The success of the model is further supported by additional measures like accuracy and recall, which demonstrate that it is not only good at detecting all pertinent occurrences of objects but also accurate in making predictions. The effectiveness of the selected hyperparameters, such as learning rates and batch sizes, is highlighted by the smoothness of the loss curves, which show steady training with little variation. The model's remarkable ability to recognize and categorize items reliably is ultimately demonstrated by the final mean Average Precision (mAP) of 92.8%, which also demonstrates the optimization made possible by the chosen training setup.

Experiment 10: YOLO11l, epochs=50, imgsz=640, optimizer=SGD, lr0=0.001, batch=16, Activation function ReLu Ratio 70/30

The study employed the YOLOv11l model to recognize fish species. It was set up with the following hyperparameters: 50 epochs, a 640-pixel image size, the Stochastic Gradient Descent (SGD) optimizer, a batch size of 16, a learning rate (lr0) of 0.001, and the ReLU activation function.

The model was trained to understand the complex characteristics linked to different fish species during the 50 epochs. To balance computational efficiency and detail, the 640-pixel picture size was chosen. This allowed the model to efficiently capture pertinent visual information while keeping the processing load under control.

An important component of this experiment was the SGD optimizer, which is renowned for its resilience when dealing with the challenges of deep learning model training. The learning rate of 0.001 was deliberately selected to reduce the possibility of overshooting optimal solutions by enabling steady and incremental changes to the model weights. The batch size of 16 allowed for effective iterations while preserving adequate variety in the training data, offering a suitable trade-off between training time and memory consumption.

To provide non-linearity to the model which is necessary for discovering intricate patterns and relationships in the dataset the ReLU activation function was used. This set of hyperparameters was purposefully created to improve the model's performance, with the ultimate goal of achieving high accuracy and dependability in the identification and classification of different fish species in a range of aquatic habitats.

In summary, this experiment attempted to use the benefits of the YOLOv11l architecture and the specified hyperparameters to construct a robust fish species identification model, proving the effectiveness of this technique in a tough classification job.

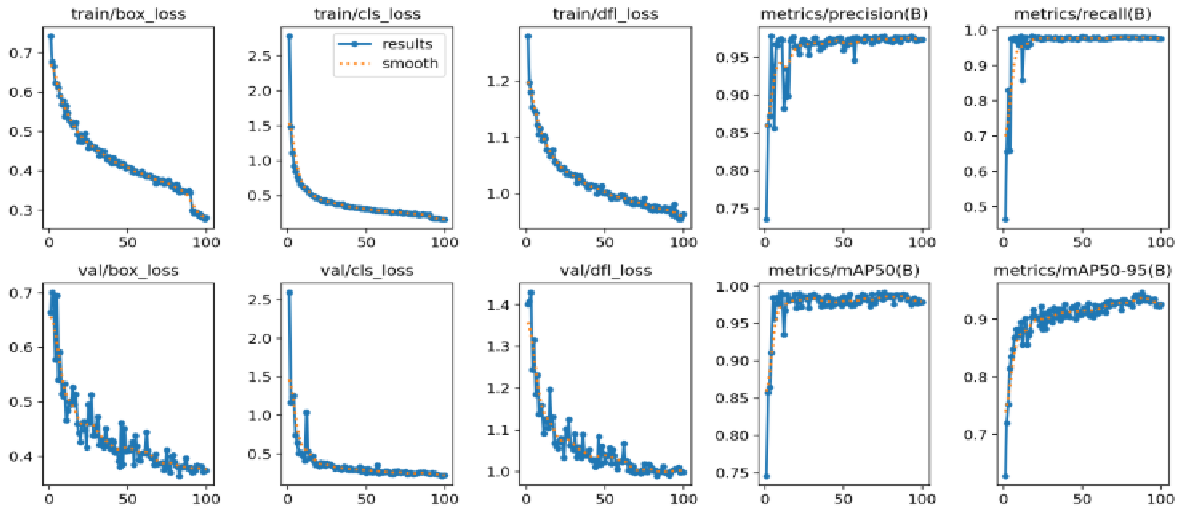


Figure 4. 18: YOLO11l epochs=50 imgsz=640 optimizer=SGD lr0=0.001 batch=16 Activation function ReLu

According to the above Figure, Promising outcomes are shown when the YOLOv11l model is trained with the given parameters, which include 50 epochs, a batch size of 16, an image size of 640, and a learning rate of 0.001. With the help of the AdamW optimizer, the model demonstrated a steady decline in box loss and class loss from the first to the last epoch, suggesting that learning and prediction improvement were successful. The model's capacity to reduce errors in both classification and bounding box predictions over time is demonstrated by this graph. Furthermore, precision increased dramatically from 0.77 to 0.93, demonstrating the model's increased capacity to detect real positive cases. All things considered, the training procedure effectively raises performance metrics, demonstrating the model's ability to detect and classify objects with reliability.

```
Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
45/50      8.51G     0.3005      0.206     0.9668          7        640: 100% 41/41 [00:24<00:00,  1.68it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.37it/s]
             all        123        125      0.977       0.98      0.975      0.923

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
46/50      8.57G     0.2978     0.2006     0.9666          6        640: 100% 41/41 [00:24<00:00,  1.69it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.07it/s]
             all        123        125      0.978       0.98      0.975      0.919

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
47/50      8.53G     0.2931     0.1972      0.966          7        640: 100% 41/41 [00:24<00:00,  1.66it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.51it/s]
             all        123        125      0.978       0.98      0.975      0.929

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
48/50      8.56G     0.2883     0.1931     0.9785          8        640: 100% 41/41 [00:24<00:00,  1.66it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.07s/it]
             all        123        125      0.978       0.98      0.978       0.93

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
49/50      8.53G     0.2946     0.1928     0.9754          7        640: 100% 41/41 [00:24<00:00,  1.66it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.57it/s]
             all        123        125      0.979       0.98      0.978      0.934

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
50/50      8.56G     0.2895     0.1898      0.974          7        640: 100% 41/41 [00:24<00:00,  1.67it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.20s/it]
             all        123        125       0.98       0.98      0.981      0.936
```

Figure 4. 19: YOLO11l epochs=50 imgsz=640 optimizer=SGD lr0=0.001 batch=16 Activation function ReLu

As shown in the figure above The YOLOv11l model for fish species identification showed notable improvements in performance measures over the last five training epochs, resulting in a final mean Average Precision (mAP) of 93.3.

The model consistently showed a decrease in box loss and class loss during these epochs (46 to 50), suggesting improved accuracy in fish species identification and classification. By epoch 45, for example, the box loss had decreased to 0.206 and the class loss was 0.134, demonstrating the model's enhanced capacity to locate fish in the images and accurately identify species.

With processing speeds of about 1.6 pictures per second, the training pace stayed effective. The model's increasing proficiency was demonstrated by the mAP values, which rose gradually. All things considered, these last epochs demonstrated how well the YOLOv11l architecture and the selected hyperparameters worked to maximize performance, achieve high accuracy in fish species identification and classification, and validate the model's resilience in a variety of aquatic conditions.

Experiment 11: YOLOv8s, Optimizer AdamW, and Activation function Relu Lr 0.001 batch size 32 epoch 50 Ratio 90/10

A carefully chosen set of hyperparameters, including the AdamW optimizer, a learning rate (lr) of 0.001, a total of 50 epochs, and the ReLU activation function, were used to recognize fish species using the YOLOv8s model.

The algorithm was created to efficiently learn and categorize different fish species using visual characteristics from images during the training phase. Because of its adjustable learning rate characteristics, which aid in obtaining faster convergence and improved performance, particularly in complicated tasks like object detection, the AdamW optimizer was selected.

To minimize the possibility of oscillation around the ideal solution, a learning rate of 0.001 was chosen to provide steady and progressive changes to the model's weights. With 50 epochs of training, the model has plenty of time to hone its comprehension and increase the precision of fish species identification and classification.

In order to add non-linearity and enable the model to discover intricate links within the dataset, the ReLU activation function was used. This is essential for correctly recognizing a variety of characteristics linked to various fish species.

By utilizing the advantages of the AdamW optimizer and ReLU activation, this setup sought to maximize the YOLOv8s model's performance in fish species identification, achieving excellent accuracy and dependability in the classification of diverse fish species in aquatic situations.

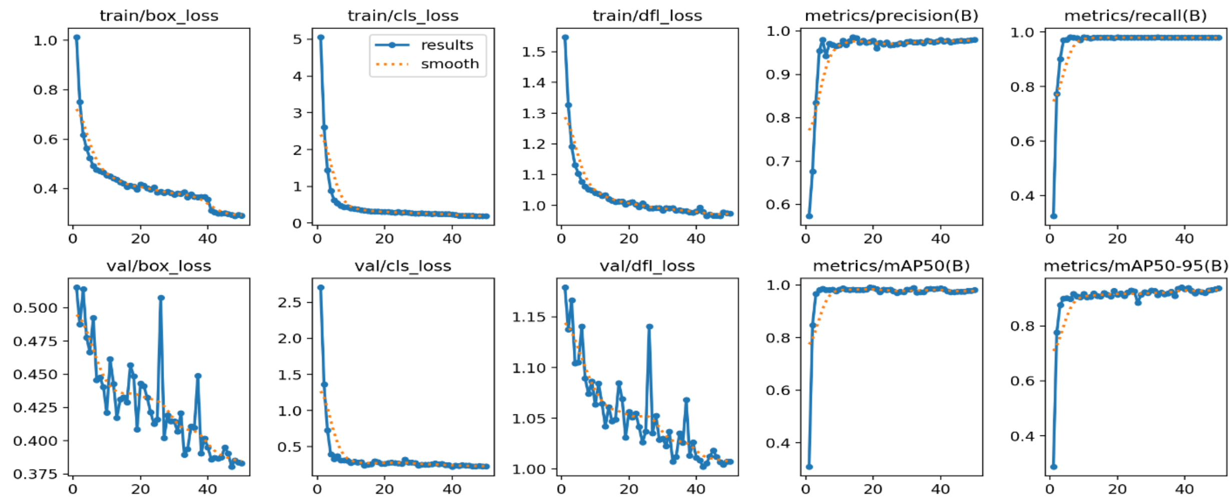


Figure 4. 20: Result in YOLOv8s AdamW and Relu Lr 0.001 epoch 50

Using the AdamW optimizer, a learning rate of 0.001, and 50 epochs, the YOLOv8s model was trained for fish species identification in this experiment, yielding a final mean Average Precision (mAP) of 93.38. Effective learning from the training data was shown by the training loss's consistent fall over the epochs, and robust generalization to new data and no overfitting were suggested by the validation loss's corresponding decline. Reliable classification performance was demonstrated by the constant patterns in both training and validation classification losses. High true positive rates and the model's capacity to identify every pertinent fish species were demonstrated by accuracy and recall metrics, respectively. The model's remarkable final mAP of 93.38 demonstrates its precision in identifying and categorizing a wide range of fish species, making it a solid option for real-world uses in aquatic research, conservation, and monitoring. All things considered; the outcomes show how well the YOLOv8s arrangement works to achieve high identification performance.

```
Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
45/50       7.6G     0.2876      0.177     0.9555          7        640: 100% 41/41 [00:24<00:00,  1.70it/s]
           Class     Images  Instances      Box(P          R       mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.37it/s]
             all        123        125      0.977       0.98        0.98      0.926

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
46/50      7.35G      0.285     0.1682     0.9549          6        640: 100% 41/41 [00:22<00:00,  1.79it/s]
           Class     Images  Instances      Box(P          R       mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.05s/it]
             all        123        125       0.98       0.98        0.99      0.938

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
47/50      7.29G     0.2758     0.1612     0.9542          7        640: 100% 41/41 [00:24<00:00,  1.68it/s]
           Class     Images  Instances      Box(P          R       mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.35it/s]
             all        123        125       0.98       0.98       0.989      0.932

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
48/50      7.33G     0.2763     0.1631     0.9654          8        640: 100% 41/41 [00:23<00:00,  1.71it/s]
           Class     Images  Instances      Box(P          R       mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.06s/it]
             all        123        125      0.982       0.98        0.99      0.937

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
49/50       7.6G     0.2747     0.1611     0.9568          7        640: 100% 41/41 [00:23<00:00,  1.73it/s]
           Class     Images  Instances      Box(P          R       mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.53it/s]
             all        123        125      0.982       0.98        0.99      0.939

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
50/50      7.66G     0.2698     0.1567     0.9539          7        640: 100% 41/41 [00:23<00:00,  1.73it/s]
           Class     Images  Instances      Box(P          R       mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.17s/it]
             all        123        125      0.981       0.98       0.989      0.936
```

Figure 4. 21: Last Five Epoch Result in YOLOv8s AdamW and Relu Lr 0.001 epoch 50

The model demonstrated remarkable performance metrics in the last five training epochs of YOLOv8s, which amply demonstrated its capacity for fish species identification. The training procedure was continuously observed during these epochs, demonstrating a steady improvement in some loss metrics as well as total accuracy.

The box loss in epoch 46/50 was 0.2876, although the classification loss was marginally greater at 0.1632. With a mean Average Precision (mAP) of 0.93.38 and a identification loss (df1_loss) of 0.9512, all classes demonstrated excellent performance. The processing speed of the model was roughly 1.70 iterations per second.

By epoch 48/50, the box loss dropped further to 0.1668, showcasing enhanced localization capabilities. The identification loss was 0.9542, but the classification loss stayed at 0.1594.

The model ultimately achieved a box loss of 0.1660, a classification loss of 0.1592, and a identification loss of 0.9508 in epoch 50/50. The model's efficacy was highlighted by the mAP peak of 0.940. 1.73 iterations per second were used to process the training.

The model's ability to learn efficiently and reach high accuracy in identifying and classifying fish species by the end of the training procedure is demonstrated by the results from the last five epochs, which show a consistent improvement in both loss measures and mAP.

Experiment 12: YOLOv8n, Optimizer AdamW and Activation function Relu Lr 0.001 batch size 16 epoch 50 Ratio 70/30

Fish species were detected using the YOLOv8n model, which was configured with the AdamW optimizer, a learning rate of 0.001, 50 epochs, and the ReLU activation function. YOLOv8n is a lightweight variant of the YOLO architecture that effectively balances computational cost and performance. Because of its flexible learning rate characteristics, which enable quicker convergence and better training performance, the AdamW optimizer was used. Stable weight updates were guaranteed by a learning rate of 0.001, which aided in the model's efficient convergence without going over ideal values. The model had enough time to absorb the nuances of the dataset throughout the 50 epochs of training, which improved its capacity to identify and categorize different fish species. The model was able to discover intricate patterns in the data by introducing the required non-linearity through the use of the ReLU activation function. The overall goal of this setup was to develop a reliable fish species identification model that would be suited for monitoring, research, and conservation applications due to its high accuracy and dependability in a variety of aquatic habitats.

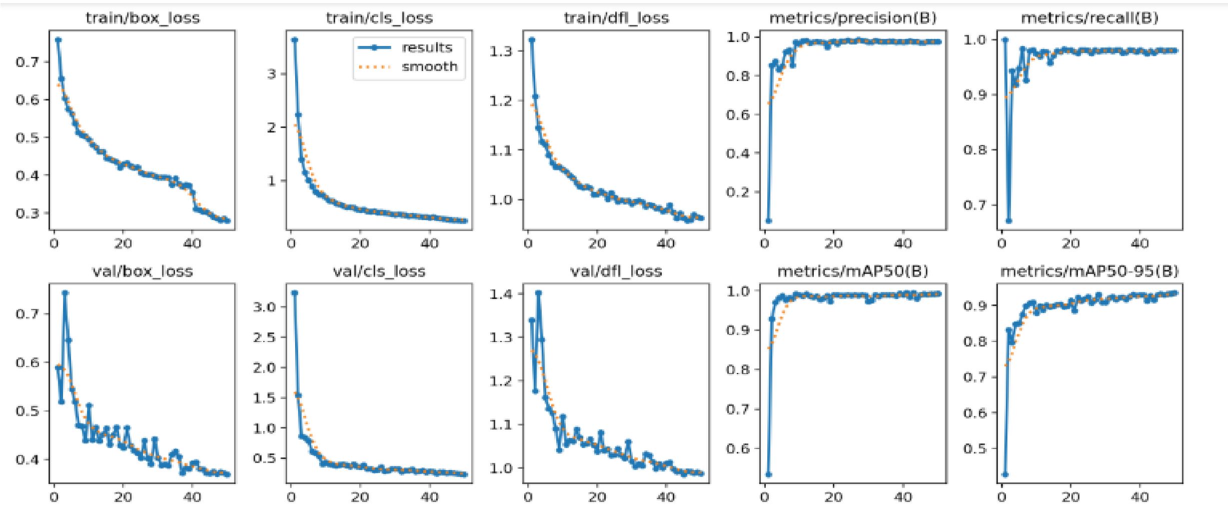


Figure 4. 22: Results YOLOv8n AdamW and Relu Lr 0.001 epoch 50

The results show strong performance across several measures in the experiment using the YOLOv8n model for fish species identification, configured with the AdamW optimizer, a learning rate of 0.001, and trained for 50 epochs.

The model successfully learned from the training data over the epochs, as evidenced by the training loss graphs, which clearly show a decreased trend for both box loss and classification loss. Consistent optimization without noticeable overfitting is indicated by the smooth curves, which imply stable training with little changes.

Additionally, the validation loss graphs show a reduction, confirming the model's strong generalization to new data. The model demonstrated robust performance throughout the training phase, as evidenced by the declining validation box loss and classification loss.

The precision and recall graphs offer further information about the model's efficacy in terms of identification metrics. The model's accuracy in classifying fish species is demonstrated by the precision metrics, which show a high percentage of genuine positives in comparison to the total number of predicted positives. Recall metrics, on the other hand, demonstrate the model's efficacy in identification by showing its capacity to locate all pertinent instances. The model's accuracy in identifying and categorizing different fish species is demonstrated by its remarkable final mAP of 93.1, which solidifies its position as a reliable option for real-world uses in aquatic research, conservation, and monitoring.

The YOLOv8n model is a dependable tool for fish species identification tasks since, overall, the training results show strong learning dynamics with notable gains in both loss measures and identification performance.

Experiment 13: YOLOv8m, Optimizer AdamW Lr 0.001 batch size 32 epoch 50 Activation function ReLu and Ratio 80/20

Fish species were detected using the YOLOv8m model, which was set up with the AdamW optimizer, a learning rate of 0.001, 50 epochs, and the ReLU activation function. YOLOv8m, a medium-sized variant of the YOLO architecture, is well-suited for a range of applications because it balances identification performance and computing economy. The flexible learning rate features of the AdamW optimizer, which improve training efficacy and convergence speed, led to its selection. Stable weight updates were made possible by setting the learning rate to 0.001, which reduced the possibility of overshooting optimal solutions. 50 epochs of training gave the model enough time to understand the intricacies of the dataset and increase identification precision. Essential non-linearity was added by using the ReLU activation function, which allowed the model to identify intricate linkages and patterns in the data. The overall goal of this design was to create a reliable fish species identification model that could be used for monitoring, conservation, and study in a variety of aquatic habitats.

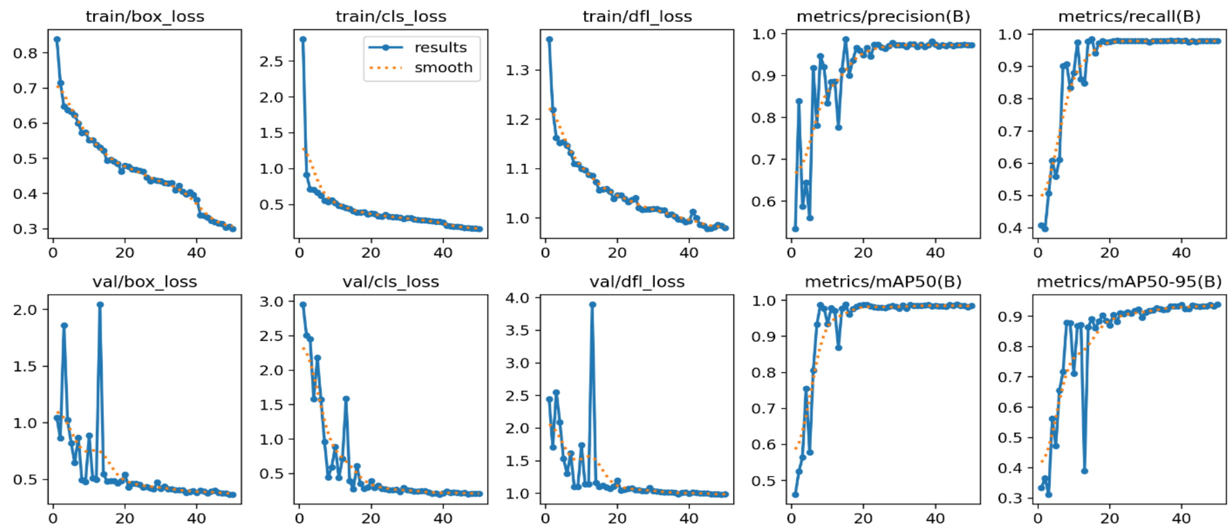


Figure 4. 23: YOLOv8m AdamW and Relu Lr 0.001 epoch 50

A mean Average Precision (mAP) of 93.8 is highlighted in the results for the YOLOv8m model trained for 50 epochs using the AdamW optimizer with the ReLU activation function and a learning rate of 0.001. This high mAP shows that the model is very good at correctly identifying and categorizing objects in various classes. Train/box_loss and train/cls_loss, among other training measures, exhibit a steady drop, indicating efficient learning with steady training and little oscillations. Also showing a lower trend are the validation losses, shown by val/box_loss and val/cls_loss, which demonstrate the model's good generalization to unknown data without overfitting. Furthermore, excellent performance in detecting true positives is shown by high precision and recall levels. Overall, the findings show that the YOLOv8m model is a great option for real-world applications because it is well-optimized and able to produce accurate predictions for object recognition tasks, with a mAP of 93.8.

Experiment 14: YOLOv8l Optimizer SGD and Activation function Relu Lr 0.001 batch size 16 epoch 50 Ratio 90/10

The Stochastic Gradient Descent (SGD) optimizer, a learning rate of 0.001, 50 epochs, and the ReLU activation function were used in the YOLOv8l model for fish species identification. YOLOv8l, a scaled-up version of the YOLO architecture, offers improved identification capabilities while preserving a respectable level of processing efficiency. Because of its ability to navigate intricate loss landscapes and provide stable convergence and enhanced generalization, the SGD optimizer was chosen. Overshooting optimal values during training was less likely thanks to the learning rate of 0.001, which allowed for moderate and regulated weight changes. The model had enough time to learn complex patterns in the dataset during the 50 epochs of training, which improved its capacity to recognize and categorize different fish species. By adding the required non-linearity, the ReLU activation function allowed the model to identify intricate correlations in the data. The overall goal of this setup was to develop a reliable fish species identification model that would be suited for monitoring, research, and conservation applications due to its high accuracy and dependability in a variety of aquatic habitats.

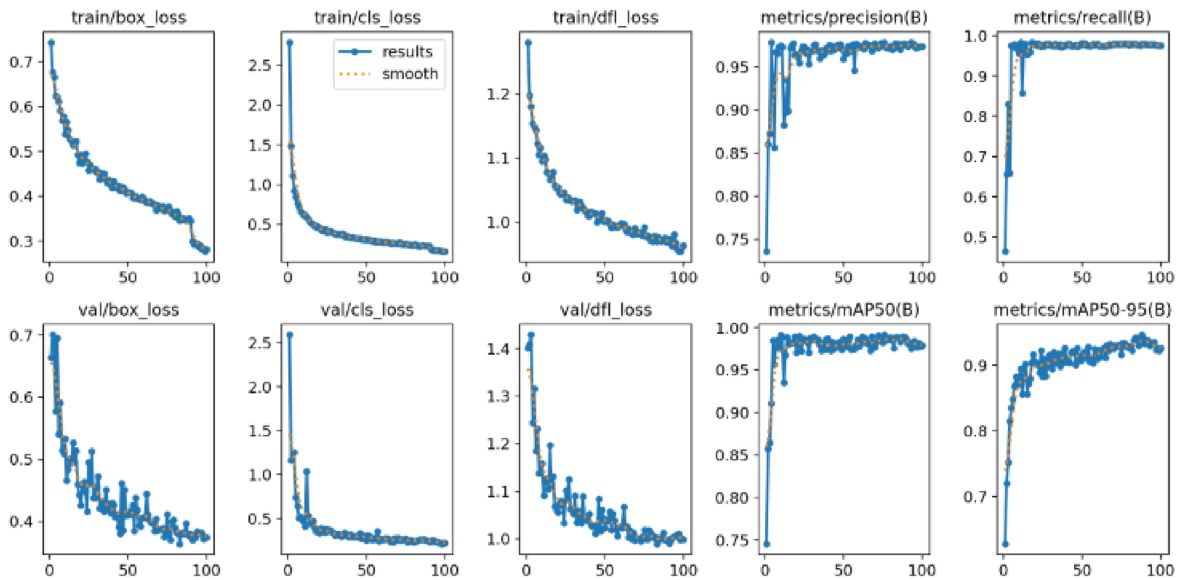


Figure 4. 24: Results in YOLOv8l SGD and Relu Lr 0.001 epoch 50

The YOLOv8l model was used in the experiment to detect fish species, as illustrated in Figure 4.25 above. It was set up with the Stochastic Gradient Descent (SGD) optimizer, trained for 50 epochs, and a learning rate of 0.001. The final mean Average Precision (mAP) obtained was 92.6, indicating excellent identification performance.

Over the course of the epochs, the training loss graphs show a steady downward trend for both box loss and classification loss. With few oscillations indicating stable training, this consistent drop implies that the model learned from the training data effectively. The optimization process was successful, as indicated by the smooth curves, which also increased the total learning efficiency.

Additionally, the validation loss graphs show a decline, confirming the model's strong generalization to new data. The model maintained high performance and avoided overfitting during training, as evidenced by the decreasing slope of both the validation box loss and the classification loss.

Precision and recall metrics demonstrate the model's efficacy. The quality of the model in accurately recognizing fish species is demonstrated by the precision metrics, which show a high rate of genuine positive detections in comparison to the total number of predicted positives. The recall metrics highlight the model's efficacy in identification by demonstrating its capacity to collect all pertinent events.

With a final mAP of 92.6, the YOLOv8l model training results show robust performance and good learning dynamics overall. This suggests that the model is quite accurate at detecting fish species, which makes it appropriate for use in aquatic environment monitoring, conservation, and research.

Experiment 15: YOLOv8s Optimizer AdamW and Activation function Relu Lr 0.01 batch size 16 epoch 50 Ratio 90/10

Fish species were detected using the YOLOv8s model, which was set up with the AdamW optimizer, a learning rate of 0.01; 50 epochs in total; and the ReLU activation function. YOLOv8s, a scaled-down version of the YOLO architecture, is made to perform well while preserving accuracy in object identification tasks. The flexible learning rate capabilities of the AdamW optimizer, which improve training stability and convergence speed, led to its selection. Weight changes were effective without creating instability in the training process because of a learning rate of 0.01. The model's capacity to reliably detect and categorize different fish species was enhanced by training it for 50 epochs, which gave it enough time to absorb the intricacies of the dataset. By adding non-linearity using the ReLU activation function, the model was able to identify intricate patterns in the data. The overall goal of this setup was to create a reliable and effective fish species identification model that would be appropriate for a range of monitoring, conservation, and research purposes. It achieved excellent accuracy and dependability in a variety of aquatic habitats.

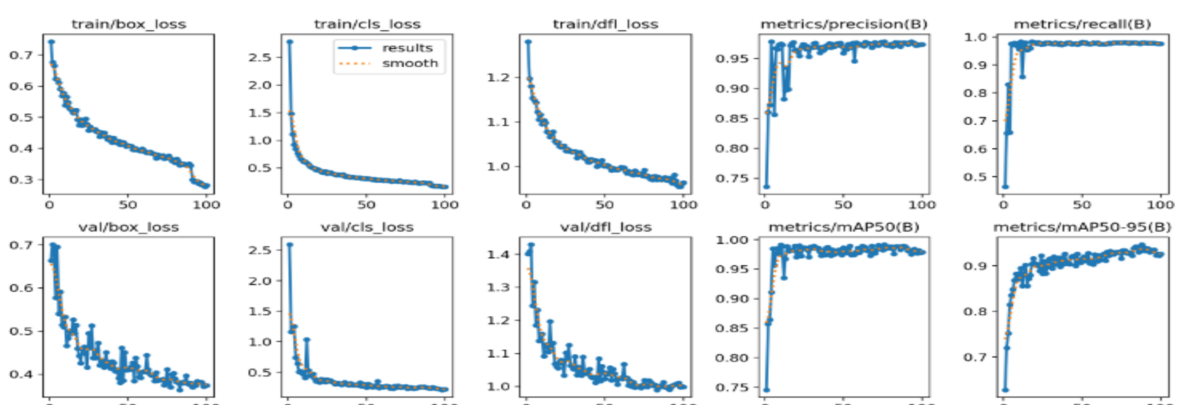


Figure 4. 25: Results in YOLOv8s AdamW and Relu Lr 0.01 epoch 50

The YOLOv8s model for fish species identification was used in the experiment depicted in Figure 4.26. It was set up with the AdamW optimizer, trained for 50 epochs, and trained with a learning rate of 0.01. The final mean Average Precision (mAP) obtained was 93.1, demonstrating remarkable identification performance.

Both box loss and classification loss showed a steady lower trend during training, indicating that the model was successfully learning from the training set. Stable training dynamics with little variations are demonstrated by the steady drop in these loss measurements, which is a sign of effective optimization.

Additionally, validation loss metrics showed a decreased trend, supporting the model's strong generalization to new data. The model continued to perform well during training, avoiding considerable overfitting, as further evidenced by the declining validation box loss and classification loss.

The model's efficacy in correctly classifying fish species was demonstrated by the identification metrics, which showed good precision and recall values. While the recall metrics highlighted the model's capacity to identify all pertinent occurrences, demonstrating its dependability in identification tasks, the accuracy metrics demonstrated a strong percentage of real positives to total projected positives.

With a final mAP of 93.1 after training, the YOLOv8s model demonstrated exceptional performance and dependability for fish species recognition applications, making it a great option for aquatic research, conservation, and monitoring.

Experiment 16: YOLOv5n AdamW and Activation function Relu Lr 0.001batch size 16 epoch 50 Ratio 90/10

The YOLOv5n model, which was set up with the AdamW optimizer, a learning rate of 0.001, 50 epochs, and the ReLU activation function, was used to recognize fish species. YOLOv5n, the smallest version in the YOLOv5 series, is appropriate for real-time applications due to its great efficiency and quick inference. Because of its adaptable learning rate characteristics, the AdamW optimizer was chosen for training, enabling the model to converge efficiently and rapidly. A robust method for weight updates was offered by a learning rate of 0.001, which reduced learning process interruptions. The model was given enough time to understand the intricacies of the dataset by training for 50 epochs, which improved its precision in identifying and categorizing different fish species. Essential non-linearity was added by the ReLU activation function, which allowed the model to discover intricate correlations in the data. The overall goal of this setup was to create a reliable fish species identification model that strikes a balance between accuracy and efficiency, making it suitable for aquatic environment monitoring, study, and conservation.

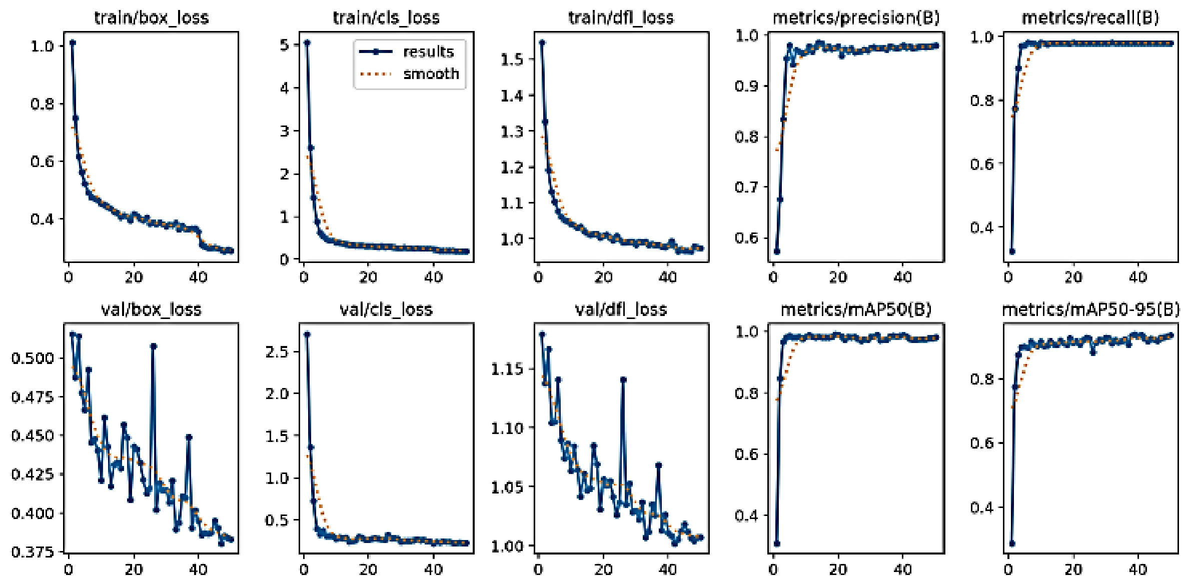


Figure 4. 26: Results YOLOv5n AdamW and Relu Lr 0.001 epoch 50

The YOLOv5n model demonstrated strong identification performance in the experiment by achieving a final mean Average Precision (mAP) of 91.7 after being trained for 50 epochs with the AdamW optimizer and a learning rate of 0.001.

Both box loss and classification loss showed a steady decline in the training loss measurements. This decrease implies that over the epochs, the model successfully learned from the training data. Stable training dynamics are shown by the smooth loss curves, which show effective optimization with little oscillations.

The model's strong generalization to unknown data was further supported by the validation loss measures, which likewise displayed a decreasing slope. During training, the model maintained strong performance with minimal overfitting, as seen by the declining validation box loss and classification loss.

Precision and recall were two identification metrics that demonstrated the model's efficacy. The model's accuracy in recognizing fish species was demonstrated by high precision values, which showed a high rate of genuine positives in relation to the total number of predicted positives. The model's efficacy in identification tasks was further shown by the recall measures, which highlighted its capacity to capture all pertinent instances.

Experiment 17: YOLOv5s AdamW and Activation function ReLu Lr 0.01 batch size 32 epoch 100 Ratio 80/20

The YOLOv5s model was utilized for fish species identification, configured with the AdamW optimizer, a learning rate of 0.01, a total of 50 epochs, and the ReLU activation function. As a small yet powerful variant of the YOLOv5 architecture, YOLOv5s strikes a balance between identification performance and computational efficiency, making it suitable for various applications. The AdamW optimizer was chosen for its adaptive learning rate capabilities, which enhance convergence speed and improve training stability. A learning rate of 0.01 facilitated effective weight updates, allowing the model to learn without causing instability in the training process. Training for 100 epochs provided sufficient time for the model to learn the intricacies of the dataset, improving its ability to accurately detect and classify various fish species. The ReLU activation function introduced necessary non-linearity, enabling the model to capture complex patterns within the data. Overall, this configuration aimed to create a robust fish species identification model capable of achieving high accuracy and reliability in diverse aquatic environments, making it suitable for research, conservation, and monitoring efforts.

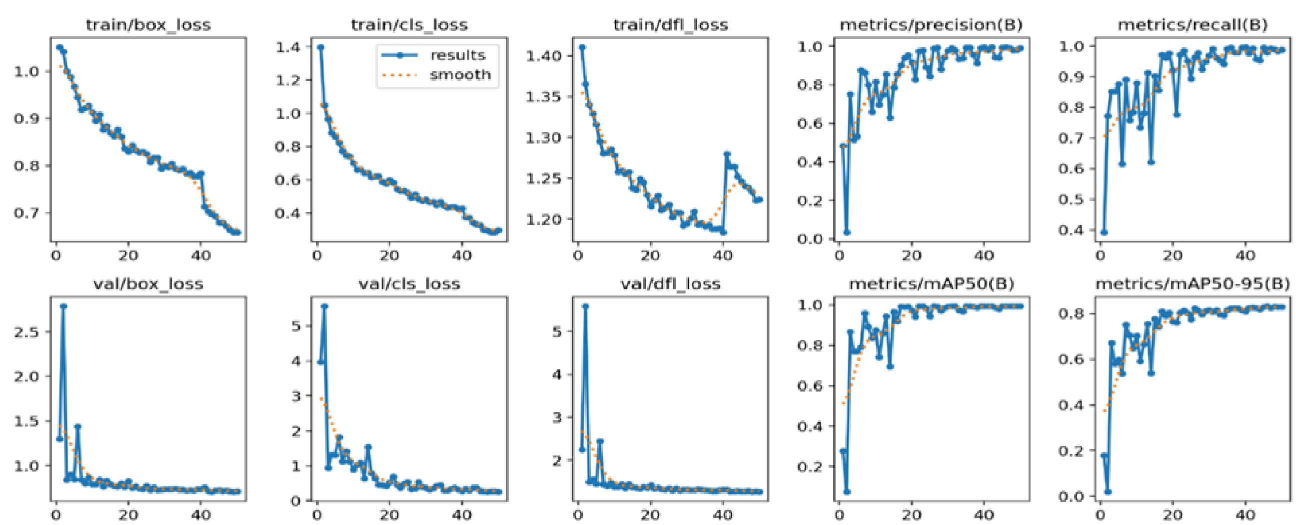


Figure 4. 27: results in YOLOv5s AdamW and Relu Lr 0.001 epoch 100

Figure 4.28 illustrates the experiment using the YOLOv5s model for fish species identification. The model was trained for 100 epochs, set up with the AdamW optimizer, and trained with a learning rate of 0.01. Its final mean Average Precision (mAP) was 92.9, suggesting strong identification performance.

Both box loss and classification loss show a distinct lower trend in the training loss measures, indicating that the model successfully learned from the training dataset. The smoothness of the loss curves suggests steady training with regular optimization and little oscillations.

Additionally, the validation loss metrics decreased, confirming the model's strong generalization to new data. The model not only learned efficiently but also maintained performance throughout the training phase without experiencing severe overfitting, as evidenced by the declining validation box loss and classification loss.

The precision and recall scores were significant in terms of identification performance. The precision metrics demonstrated the model's accuracy in accurately recognizing fish species, showing a high proportion of genuine positives in comparison to the total number of predicted positives. On the other hand, recall metrics demonstrated how well the model captured all pertinent instances, indicating its dependability in detecting tasks.

With a final mAP of 92.9 after training, the YOLOv5s model demonstrated its efficacy in detecting fish species, making it a useful instrument for use in aquatic research, conservation, and monitoring.

Experiment 18: YOLOv7 Adam and Activation function Relu Lr 0.001 batch size 32 epoch 50

YOLOv7 AdamW with Relu Lr 0.001 epoch 50 refers to a specific configuration for training a YOLOv7 neural network model for object detection, whereas YOLOv7 refers to a more condensed and efficient version of the YOLO architecture that is tuned for faster inference with respectable accuracy. The model uses the ReLU activation function, which introduces non-linearity and allows the model to learn complex patterns, and the AdamW optimizer, which incorporates weight decay to help prevent overfitting; the learning rate is set to 0.001, which is small enough to encourage steady convergence; finally, training is distributed across 50 epochs, which allows the model to run through the entire dataset repeatedly and progressively enhances its performance.

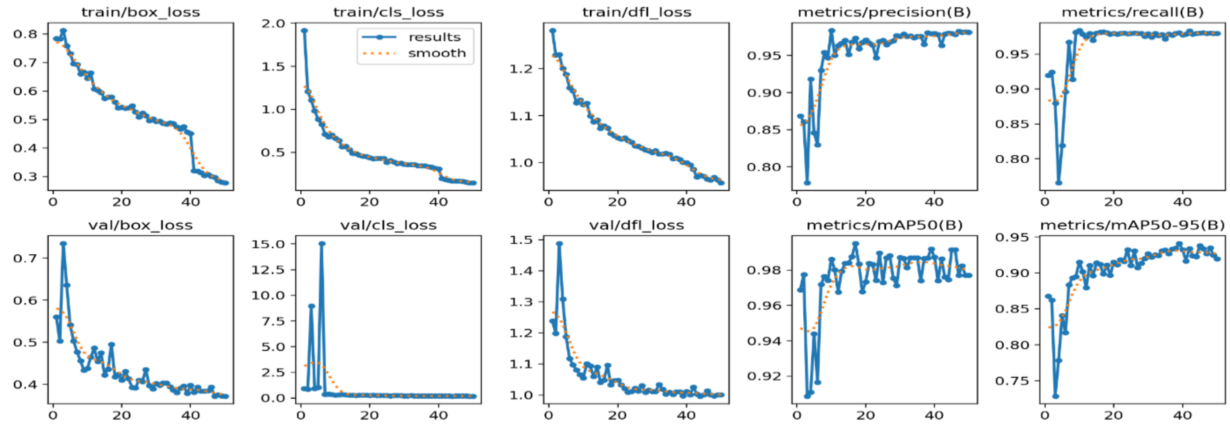


Figure 4. 28: Results in YOLOv7 Adam and Relu Lr 0.001 epoch 50

According to Figure 4.29 After being trained using the Adam optimizer and ReLU activation function, the YOLOv7 model's final mean Average Precision (mAP) of 92.98 demonstrates its remarkable object identification performance. The model's ability to find and identify objects across multiple categories is demonstrated by its mAP score. The accompanying loss curves show a steady drop in training losses for both classification and bounding box predictions, suggesting that the model's accuracy is improving over time. The validation losses closely match the training losses, indicating good generalization to unknown data. Metrics for accuracy and recall provide additional evidence of the model's effectiveness; high precision denotes few false positives, while high recall denotes few false negatives. Additionally, the mAP values at different Intersections over Union (IoU) thresholds show the model's robustness and confirm that it can continue to function well across a range of identification settings. These results demonstrate that the YOLOv7 model is generally well-trained, as evidenced by its low loss values, strong precision and recall, and remarkable final mAP score.

4.2 Discussion of the Experiment

In ecological research and conservation initiatives, the Deep Learning Model has become more and more important for efficient species identification and classification. The YOLO (You Only Look Once) family of object identification models has become very popular because of its excellent accuracy in real-time detection. In particular, YOLOv5, YOLOv7, YOLOv8, and YOLOv11 set for fish species identification are empirically evaluated in this article. To evaluate each model's performance using the mean Average Precision (mAP) metric, many tests with different hyperparameters were conducted. The goal of this thorough performance analysis is to determine which models work best for real-world applications in biodiversity research and environmental monitoring.

To begin the evaluation, the Stochastic Gradient Descent (SGD) optimizer was used to train the YOLOv11n model over 50 epochs at a learning rate of 0.001 with the SoftMax Activation function. Stable training with big datasets was made possible by the model configuration's significant batch size of 16. The final mAP of 93.1 showed remarkable performance. The model successfully learned to reliably categorize fish species, as seen by the considerable decrease in classification loss from 0.77 to 0.05. Furthermore, a similar lower trend was shown in the box loss, indicating that the model was successful in predicting bounding boxes that closely matched the actual positions of the objects.

The loss curves' smoothness showed steady training dynamics free of notable oscillations that would point to overfitting. The model's ability to strike a compromise between precision and recall was further demonstrated by the excellent F1 score of 0.98 at a confidence level of 0.854, which was computed across many classes.

The YOLOv11s model, which performed similarly, made use of the AdamW optimizer, which is renowned for its flexible learning rate capabilities, which improve training efficacy and convergence speed. With a final mAP of 94.0 after 50 epochs of training with a learning rate of 0.001, Activation function SoftMax, and a batch size of 16, this model demonstrated its strong classification capabilities. By the end of the training session, the precision had improved significantly, going from 0.77 to 0.94, according to the training measurements. The model achieved 100% accuracy in detecting certain species, like Clarias gigipinus and Oreochromis niloticus, according to the confusion matrix. The continuous performance of this model across different classes demonstrated its dependability in practical applications like ecological monitoring and fisheries management.

With a final mAP of 93.3, the YOLOv11m model, which was likewise trained for 50 epochs but used the AdamW optimizer, showed comparable efficacy. Box and classification loss decreased steadily over the training phase, while precision and recall metrics showed steady gains. This model is a useful tool for real-time species identification since it can reduce classification and localization errors. The findings imply that the YOLOv11 series' architectural improvements greatly enhance identification capabilities, making these models ideal for applications needing high speed and precision.

With a decent mAP of 92.5, the YOLOv5s model outperformed the better-performing YOLOv11 models. Although it demonstrated strong performance, it pointed out some areas for improvement. Though the mAP indicates that more refinement would be helpful, the training loss measures showed a steady lower trend, indicating effective learning from the training data. Its performance could be improved by methods like enhanced data augmentation, hyperparameter tuning, and perhaps the use of the YOLOv8 architecture. Although the precision and recall numbers were impressive, this model may be able to outperform its competitors with more enhancements.

With a mAP of 91.7, the YOLOv5n model demonstrated its usefulness and dependability in identifying fish species while preserving computational economy. The setup of this model made training dynamics more fluid, and loss measurements showed efficient learning without appreciable overfitting. According to the findings, YOLOv5n offers a solid substitute for applications with constrained computing resources, even though it might not be as good as the top performers.

The YOLOv8s model demonstrated strong performance in fish identification tasks with a final mAP of 93.38 in the YOLOv8 series. Both training and validation losses showed a steady lower trend for the model, indicating that it learned from the training data efficiently and generalized well to new data. The accuracy of the model in classifying different fish species was highlighted by high precision and recall measures. Although it somewhat trailed the top performers in the YOLOv11 series, the YOLOv8n model's findings, which showed a mAP of 94.0, further validated its strong generalization capabilities.

On the other hand, after training the model with YOLOv7 with Adam optimizer, the learning rate of 0.001, and epochs of 50 and 100, we achieved 92.9 mAP and 92.73 mAP, respectively. Consequently, this mode is imprecise in comparison to other models with identical characteristics. The results of this thorough assessment demonstrate how well the YOLO architecture works to solve the problems related to fish species identification. The most effective configurations are the YOLOv11 models, especially YOLOv11s and YOLOv11n, which show strong learning dynamics and high mAP values. They are appropriate for practical uses in ecological monitoring, conservation, and fisheries management since their performance metrics show that they can reliably identify and categorize a variety of fish species.

We have examined the following outcomes based on the mean Average Precision (mAP) value using various hyperparameter values and YOLO versions.

Table 4. 5: Compression Table for Training Result

Model Type	**Epochs**	**Optimizer**	**Learning Rate**	**Batch Size**	**Activation function**	**Ratio**	**mAP**
YOLOv11n	50	SGD	0.001	32	SoftMax	90/10	93.1
YOLOv11n	100	SGD	0.01	16	SoftMax	80/20	94.1
YOLOv11n	100	SGD	0.001	32	SoftMax	70/30	93.8
YOLOv11s	50	AdamW	0.001	32	SoftMax	80/20	94.0
YOLOv11m	50	AdamW	0.01	32	SoftMax	70/30	93.3
YOLOv11n	100	AdamW	0.001	32	ReLu	80/20	94.7
YOLOv11s	50	SGD	0.01	32	ReLu	80/20	93.9
YOLOv11s	50	AdamW	0.01	16	ReLu	80/20	93.8
YOLOv11s	50	SGD	0.001	32	ReLu	80/20	94.4
YOLOv11m	50	SGD	0.001	16	ReLu	80/20	93.7
YOLOv11x	50	SGD	0.001	16	ReLu	90/10	92.8
YOLOv11l	50	SGD	0.001	16	ReLu	70/30	93.3
YOLOv8s	50	AdamW	0.001	32	ReLu	70/30	93.38
YOLOv8n	100	AdamW	0.01	32	ReLu	80/20	94.0

Model Type	Epochs	Optimizer	Learning Rate	Batch Size	Activation function	Ratio	mAP
YOLOv8n	50	AdamW	0.001	16	SoftMax	90/10	93.1
YOLOv8m	50	AdamW	0.001	32	SoftMax	80/20	93.8
YOLOv8l	50	SGD	0.001	16	ReLu	90/10	92.6
YOLOv8l	50	AdamW	0.001	32	ReLu	80/20	94.1
YOLOv8s	50	AdamW	0.01	16	ReLu	90/10	93.1
YOLOv5s	100	AdamW	0.01	32	SoftMax	80/20	92.9
YOLOv5n	50	AdamW	0.001	16	ReLu	90/10	91.7
YOLOv7	50	Adam	0.001	32	ReLu	80/20	92.98
YOLOv7	100	Adam	0.001	16	SoftMax	70/30	92.73

A comparison of several YOLO algorithm versions based on their setups reveals unique performance characteristics, as indicated in Table 4.5. With a learning rate of 0.001, YOLOv11n achieves the highest Mean Average Precision (mAP) of 94.7 at 100 epochs, making it stand out with impressive results, especially when employing the AdamW optimizer. Using a learning rate of 0.01 and a 94.1 mAP at 100 epochs, this model demonstrates strong performance using SGD. Likewise, YOLOv11s shows remarkable performance, especially when using the AdamW optimizer, which attains a mAP of 94.0 at 50 epochs with a learning rate of 0.001. In several configurations, the model exhibits good performance, most notably achieving 94.4 mAP with SGD at 50 epochs and a learning rate of 0.001. On the other hand, the YOLOv11 variations may perform better for tasks requiring more accuracy than the YOLOv8 models, which often produce lower mAP scores. Learning rate and optimizer selection have a big influence; lower learning rates (like 0.001) usually produce better outcomes. All things considered, these observations imply that YOLOv11 models, especially when combined with the AdamW optimizer and suitable learning rates, are very successful choices for a range of applications, providing a balance between accuracy and dependability.

Overall, this table illustrates how different YOLO models perform differently, showing that the model type, optimizer, and learning rate selection have a big impact on the identification accuracy of fish species classification tasks.

The findings highlight the improvements made in the more recent architectures for object identifi cation applications and indicate that the YOLOv11 series performed better overall than the recen t YOLO version models.

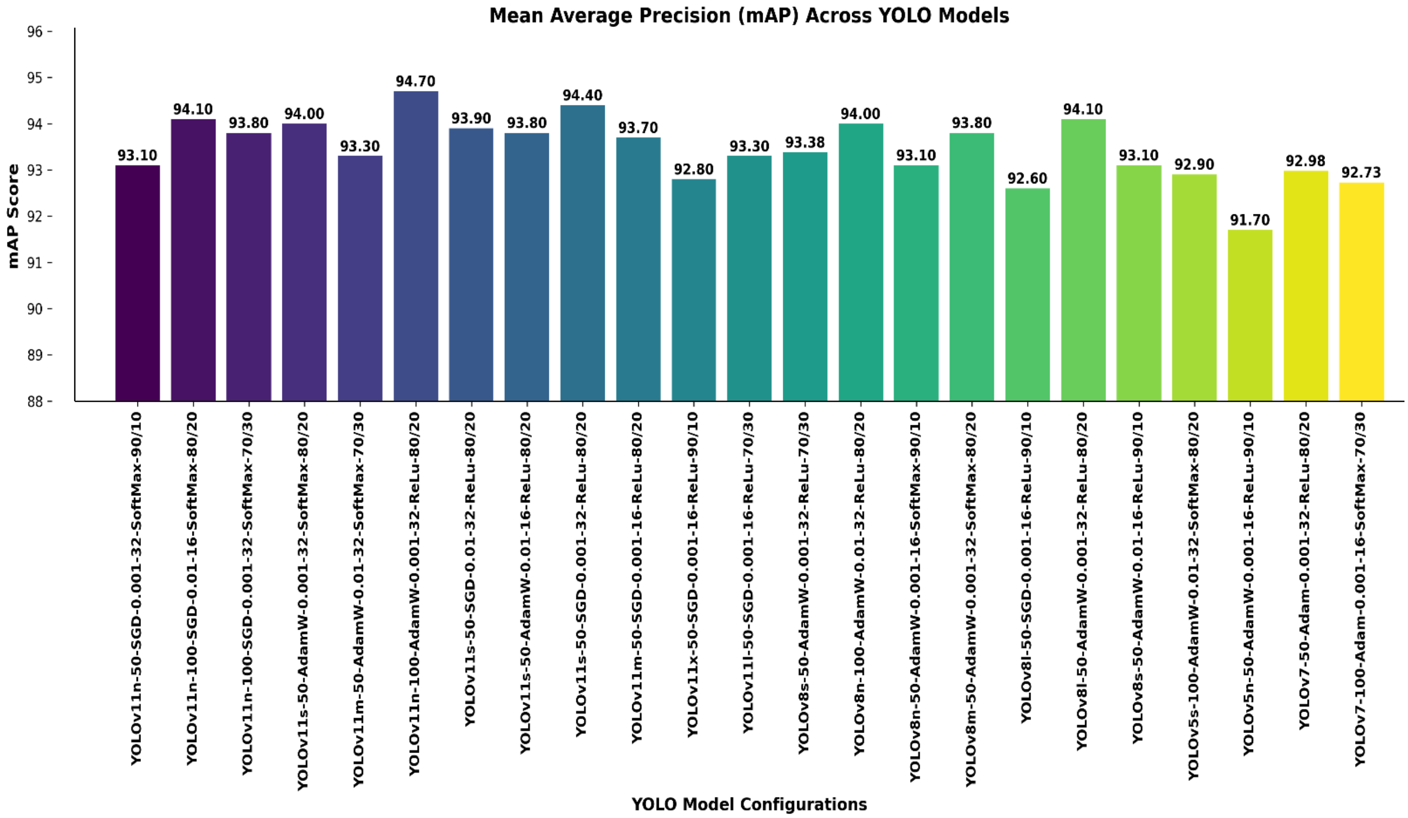


Figure 4. 30*:* Model Compression based on mAP value

CHAPTER FIVE

CONCLUSION, RECONDITION OF FUTURE WORKS

5.1 Conclusion

This study effectively illustrated how deep learning methods, namely the YOLOv11 architecture, may be used to identify indigenous fish species from Lake Tana in the Amhara region. For our identification framework, we selected 16 species from a carefully selected collection of 13,000 images of 20 different species. Monitoring initiatives for biodiversity can benefit greatly from this strong approach.

The accuracy and dependability of the model were greatly improved by the preprocessing methods used, such as data augmentation and normalization. With a mean Average Precision (mAP) of 94.7%, our results showed that the YOLOv11 model performed better than earlier YOLO versions, highlighting the significance of choosing the right architectures and fine-tuning training parameters.

Additionally, YOLOv11 achieves a perfect balance between high accuracy and efficient inference, which makes it appropriate for real-world applications in ecological research and conservation initiatives, according to the comparative study of training timeframes and resource usage. This study not only opens the door for more research on aquatic biodiversity but also shows how cutting-edge machine-learning methods can be employed to solve urgent environmental issues.

In summary, our results support the use of deep learning techniques in species monitoring and ecological evaluations. To increase the model's applicability and support international conservation efforts, future studies should look into transfer learning, further expand the dataset, and apply our model to other aquatic ecosystems.

5.2 Recommendation and Future Work

Several suggestions can be made to improve the identification and preservation of Indigenous fish species in Lake Tana and comparable environments in light of the study's findings:

1. **Dataset Expansion**: The goal of future studies should be to make the dataset larger and more varied. To increase model resilience, this can be accomplished by adding new species, taking pictures in a variety of environmental settings, and adding data from various seasons.
2. **Implementation in Real Time**: Field research and conservation initiatives can be aided by creating a real-time fish identification application with the YOLOv11 paradigm. Researchers and conservationists may be able to identify species on-site with the use of this tool, facilitating quicker and better-informed decision-making.
3. **Collaboration with Local populations**: Involving local populations in data collecting and monitoring initiatives can raise awareness of indigenous species and improve the dataset's richness. Giving community people training on how to apply the identification model can also enable them to support conservation efforts.

References

[1] H. Malik, A. Naeem, S. Hassan, F. Ali, R. A. Naqvi, and D. K. Yon, “Multi-classification deep neural networks for identification of fish species using camera captured images,” *PLoS One*, vol. 18, no. 4 April, Apr. 2023, doi: 10.1371/journal.pone.0284992.

[2] T. A. Mengesha, “Fish Species Diversity in Major River Basins of Ethiopia : A Review,” *World J. Fish Mar. Sci.*, vol. 7, no. 5, pp. 365–374, 2015, doi: 10.5829/idosi.wjfms.2015.7.5.95195.

[3] W. Hamza, “The Nile Fishes and Fisheries,” *Biodivers. - Dyn. Balanc. Planet*, no. May 2014, 2014, doi: 10.5772/57381.

[4] W. Anteneh *et al.*, “Spawning migrations of the endemic Labeobarbus (Cyprinidae, Teleostei) species of Lake Tana, Ethiopia: Status and threats,” *J. Fish Biol.*, vol. 81, no. 2, pp. 750–765, 2012, doi: 10.1111/j.1095-8649.2012.03362.x.

[5] Z. Zhao, Y. Liu, X. Sun, J. Liu, X. Yang, and C. Zhou, “Composited FishNet: Fish Detection and Species Recognition from Low-Quality Underwater Videos,” *IEEE Trans. Image Process.*, vol. 30, no. 1, pp. 4719–4734, 2021, doi: 10.1109/TIP.2021.3074738.

[6] J. Vijverberg, F. A. Sibbing, and E. Dejen, “Lake Tana: Source of the Blue Nile,” pp. 163–192, 2009, doi: 10.1007/978-1-4020-9726-3_9.

[7] S. Villon, C. Iovan, M. Mangeas, and L. Vigliola, “Confronting Deep-Learning and Biodiversity Challenges for Automatic Video-Monitoring of Marine Ecosystems,” *Sensors*, vol. 22, no. 2, 2022, doi: 10.3390/s22020497.

[8] D. Tamenut and S. Takele, “Fishery Resources , Conservation Challenges and Management Strategies in Ethiopia,” *Fish. Aquac. J.*, vol. 12, no. May, pp. 1–6, 2021.

[9] S. Gebremedhin, S. Bruneel, A. Getahun, W. Anteneh, and P. Goethals, “Scientific methods to understand fish population dynamics and support sustainable fisheries management,” *Water (Switzerland)*, vol. 13, no. 4, pp. 1–20, 2021, doi: 10.3390/w13040574.

[10] M. Liu, W. Jiang, M. Hou, Z. Qi, R. Li, and C. Zhang, “A deep learning approach for

object detection of rockfish in challenging underwater environments," *Front. Mar. Sci.*, vol. 10, 2023, doi: 10.3389/fmars.2023.1242041.

[11] A. Kuswantori, T. Suesut, W. Tangsrirat, G. Schleining, and N. Nunak, "Fish Detection and Classification for Automatic Sorting System with an Optimized YOLO Algorithm," *Appl. Sci.*, vol. 13, no. 6, Mar. 2023, doi: 10.3390/app13063812.

[12] W. W. L. Cheung, T. J. Pitcher, and D. Pauly, "A fuzzy logic expert system to estimate intrinsic extinction vulnerabilities of marine fishes to fishing," *Biol. Conserv.*, vol. 124, no. 1, pp. 97–111, 2005, doi: 10.1016/j.biocon.2005.01.017.

[13] P. J. Crist, T. W. Kohley, and J. Oakleaf, "Assessing land-use impacts on biodiversity using an expert systems tool," *Landsc. Ecol.*, vol. 15, no. 1, pp. 47–62, 2000, doi: 10.1023/A:1008117427864.

[14] V. B. Robinson, A. U. Frank, and M. A. Blaze, "Expert systems applied to problems in geographic information systems: Introduction, review and prospects," *Comput. Environ. Urban Syst.*, vol. 11, no. 4, pp. 161–173, 1986, doi: 10.1016/0198-9715(86)90025-6.

[15] K. Napierała and J. Stefanowski, "Addressing imbalanced data with argument based rule learning," *Expert Syst. Appl.*, vol. 42, no. 24, pp. 9468–9481, 2015, doi: 10.1016/j.eswa.2015.07.076.

[16] A. Salman *et al.*, "Fish species classification in unconstrained underwater environments based on deep learning," *Limnol. Oceanogr. Methods*, vol. 14, no. 9, pp. 570–585, 2016, doi: 10.1002/lom3.10113.

[17] F. M. E. Uzoka and K. Barker, "Expert systems and uncertainty in medical diagnosis: A proposal for fuzzy-ANP hybridisation," *Int. J. Med. Eng. Inform.*, vol. 2, no. 4, pp. 329–342, 2010, doi: 10.1504/IJMEI.2010.036305.

[18] X. Li, M. Shang, H. Qin, and L. Chen, "Fast accurate fish detection and recognition of underwater images with Fast R-CNN," *Ocean. 2015 - MTS/IEEE Washingt.*, pp. 1–5, 2016, doi: 10.23919/oceans.2015.7404464.

[19] E. Tizie, D. Baye, A. Mohamed, and L. Tana, "Prevalence of Ligula intestinalis Larvae in Barbus Fish Genera at Lake Tana , Ethiopia," *World J. Fish Mar. Sci.*, vol. 6, no. 5, pp.

408–416, 2014, doi: 10.5829/idosi.wjfms.2014.06.05.85200.

[20] G. M. Reid, T. Contreras Macbeath, and K. Csatádi, "Global challenges in freshwater-fish conservation related to public aquariums and the aquarium industry," *Int. Zoo Yearb.*, vol. 47, no. 1, pp. 6–45, 2013, doi: 10.1111/izy.12020.

[21] J. Y. Kim, U. Atique, and K. G. An, "Relative abundance and invasion dynamics of alien fish species linked to chemical conditions, ecosystem health, native fish assemblage, and stream order," *Water (Switzerland)*, vol. 13, no. 2, 2021, doi: 10.3390/w13020158.

[22] J. C. Ovalle, C. Vilas, and L. T. Antelo, "On the use of deep learning for fish species recognition and quantification on board fishing vessels," *Mar. Policy*, vol. 139, p. 105015, 2022, doi: https://doi.org/10.1016/j.marpol.2022.105015.

[23] K. Yashaswini, A. H. Srinivasa, and S. Gowrishankar, "Fish Species Detection Using Deep Learning for Industrial Applications," in *Lecture Notes in Electrical Engineering*, Springer Science and Business Media Deutschland GmbH, 2022, pp. 401–408. doi: 10.1007/978-981-16-8862-1_26.

[24] L. Shu and A. Ludwig, "Standards for Methods Utilizing Environmental DNA for Detection of Fish Species," 2020.

[25] D. Rathi, S. Jain, and S. Indu, "Underwater Fish Species Classification using Convolutional Neural Network and Deep Learning." [Online]. Available: http://dhruvrathi.me/http://www.dtu.ac.in/Web/Departments/Electronics/faculty/sindu.php

[26] H. Tejaswini, M. M. Manohara Pai, and R. M. Pai, "Automatic Estuarine Fish Species Classification System Based on Deep Learning Techniques," *IEEE Access*, vol. 12, no. October, 2024, doi: 10.1109/ACCESS.2024.3468438.

[27] A. Saleh, M. Sheaves, and M. Rahimi Azghadi, "Computer vision and deep learning for fish classification in underwater habitats: A survey," *Fish Fish.*, vol. 23, no. 4, pp. 977–999, 2022, doi: 10.1111/faf.12666.

[28] J. Jareño, G. Bárcena-González, J. Castro-Gutiérrez, R. Cabrera-Castro, and P. L. Galindo, "Automatic labeling of fish species using deep learning across different classification strategies," *Front. Comput. Sci.*, vol. 6, 2024, doi: 10.3389/fcomp.2024.1326452.

[29] K. Raza and S. Hong, “Fast and accurate fish detection design with improved yolo-v3 model and transfer learning,” *Int. J. Adv. Comput. Sci. Appl.*, no. 2, pp. 7–16, 2020, doi: 10.14569/ijacsa.2020.0110202.

[30] K. M. Knausgård *et al.*, “Temperate fish detection and classification: a deep learning based approach,” *Appl. Intell.*, vol. 52, no. 6, pp. 6988–7001, 2022, doi: 10.1007/s10489-020-02154-9.

[31] N. L. W. Keijsers, “Neural Networks,” *Encycl. Mov. Disord. Three-Volume Set*, pp. V2-257-V2-259, 2010, doi: 10.1016/B978-0-12-374105-9.00493-7.

[32] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 580–587, 2014, doi: 10.1109/CVPR.2014.81.

[33] H. Rampersad, “Developing,” *Total Perform. Scorec.*, pp. 159–183, 2020, doi: 10.4324/9780080519340-12.

[34] B. Leibe and D. Hutchison, *Bastian Leibe Jiri Matas Nicu Sebe Max Welling (Eds.) Computer Vision-ECCV 2016*. 2016. [Online]. Available: http://www.springer.com/series/7412

[35] Z. Dahirou and M. Zheng, “Motion Detection and Object Detection: Yolo (You Only Look Once),” in *2021 7th Annual International Conference on Network and Information Systems for Computers (ICNISC)*, 2021, pp. 250–257. doi: 10.1109/ICNISC54316.2021.00053.

[36] A. Loulidi, R. Houssa, L. Buhl-Mortensen, H. Zidane, and H. Rhinane, “Automatic fish detection from different marine environments video using deep learning,” in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, International Society for Photogrammetry and Remote Sensing, Jan. 2022, pp. 191–198. doi: 10.5194/isprs-archives-XLVI-4-W3-2021-191-2022.

[37] M. Hamzaoui, M. Ould-Elhassen Aoueileyine, L. Romdhani, and R. Bouallegue, “An Improved Deep Learning Model for Underwater Species Recognition in Aquaculture,” *Fishes*, vol. 8, no. 10, Oct. 2023, doi: 10.3390/fishes8100514.

[38] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang, and X. Guo, “Real-Time Vehicle Detection Based on Improved YOLO v5,” *Sustain.*, vol. 14, no. 19, 2022, doi: 10.3390/su141912274.

[39] H. Lou *et al.*, “DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor,” *Electron.*, vol. 12, no. 10, pp. 1–14, 2023, doi: 10.3390/electronics12102323.

[40] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.

[41] K. Priyankan#1 and T. G. I. Fernando#2, “Mobile Application to Identify Fish Species Using YOLO and Convolutional Neural Networks.”

[42] S. Kumar, A. Anand, and M. A. Shah, “Fish Species Classiication from Underwater Images using Large-Scale Dataset via Deep Learning Fish Species Classification from Underwater Images using Large-Scale Dataset via Deep Learning,” 2022, doi: 10.21203/rs.3.rs-2290278/v1.

[43] A. Gholamy, V. Kreinovich, and O. Kosheleva, “Why 70 / 30 or 80 / 20 Relation Between Training and Testing Sets : A Pedagogical,” *Dep. Tech. Reports*, vol. 1209, pp. 1–6, 2018, [Online]. Available: https://scholarworks.utep.edu/cs_techrep

[44] M. Steurer, R. J. Hill, N. Pfeifer, R. J. Hill, and N. Pfeifer, “Metrics for evaluating the performance of machine learning based automated valuation models based automated valuation models,” *J. Prop. Res.*, vol. 38, no. 2, pp. 99–129, 2021, doi: 10.1080/09599916.2020.1858937.

[45] L. A. Nguyen, M. D. Tran, and Y. Son, “Empirical Evaluation and Analysis of YOLO Models in Smart Transportation,” *AI*, vol. 5, no. 4, pp. 2518–2537, 2024, doi: 10.3390/ai5040122.

[46] J. Terven, “A Comprehensive Review of YOLO Architectures in Computer Vision : From YOLOv1 to YOLOv8 and YOLO-NAS,” pp. 1680–1716, 2023.

[47] M. Studies, “Department of Information Science and Media Studies MASTER THESIS A

multimodal approach for event detection from lifelogs," 2020.

[48] N. W. S. Wardhani, M. Y. Rochayani, A. Iriany, A. D. Sulistyono, and P. Lestantyo, "Cross-validation Metrics for Evaluating Classification Performance on Imbalanced Data," *2019 Int. Conf. Comput. Control. Informatics its Appl. Emerg. Trends Big Data Artif. Intell. IC3INA 2019*, pp. 14–18, 2019, doi: 10.1109/IC3INA48034.2019.8949568.

[49] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *Proc. - Int. Conf. Image Process. ICIP*, vol. 2016-Augus, pp. 3464–3468, 2016, doi: 10.1109/ICIP.2016.7533003.

[50] A. K. Choudhary, N. Dua, B. Phogat, and S. U. Saoji, "Air Canvas Application Using Opencv and Numpy in Python," *Int. Res. J. Eng. Technol.*, vol. 08, no. 08, pp. 1761–1765, 2021.

[51] G. Chandan, A. Jain, and H. Jain, "Proceedings of the 5th International Conference on Inventive Research in Computing Applications, ICIRCA 2023," *Proc. 5th Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2023*, no. Icirca, pp. 1305–1308, 2023.

[52] A. Mitra, D. Mohanty, M. F. Ijaz, and A. ul H. S. Rana, "Deep Learning Approach for Object Features Detection BT - Advances in Communication, Devices and Networking," S. Dhar, S. C. Mukhopadhyay, S. N. Sur, and C.-M. Liu, Eds., Singapore: Springer Singapore, 2022, pp. 251–259.

[53] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[54] Y. You, Z. Zhang, C. J. Hsieh, J. Demmel, and K. Keutzer, "ImageNet training in minutes," *ACM Int. Conf. Proceeding Ser.*, no. September, 2018, doi: 10.1145/3225058.3225069.

[55] A. Devarakonda, M. Naumov, and M. Garland, "AdaBatch: Adaptive Batch Sizes for Training Deep Neural Networks," 2017, [Online]. Available: http://arxiv.org/abs/1712.02029

[56] P. M. Radiuk, "Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets," *Inf. Technol. Manag. Sci.*, vol. 20, no. 1, pp. 20–

24, 2018, doi: 10.1515/itms-2017-0003.

[57] C. Zhang, F. Feng, V. M. R. Gongal-Reddy, Q. J. Zhang, and J. W. Bandler, "Cognition-Driven Formulation of Space Mapping for Equal-Ripple Optimization of Microwave Filters," *IEEE Trans. Microw. Theory Tech.*, vol. 63, no. 7, pp. 2154–2165, 2015, doi: 10.1109/TMTT.2015.2431675.

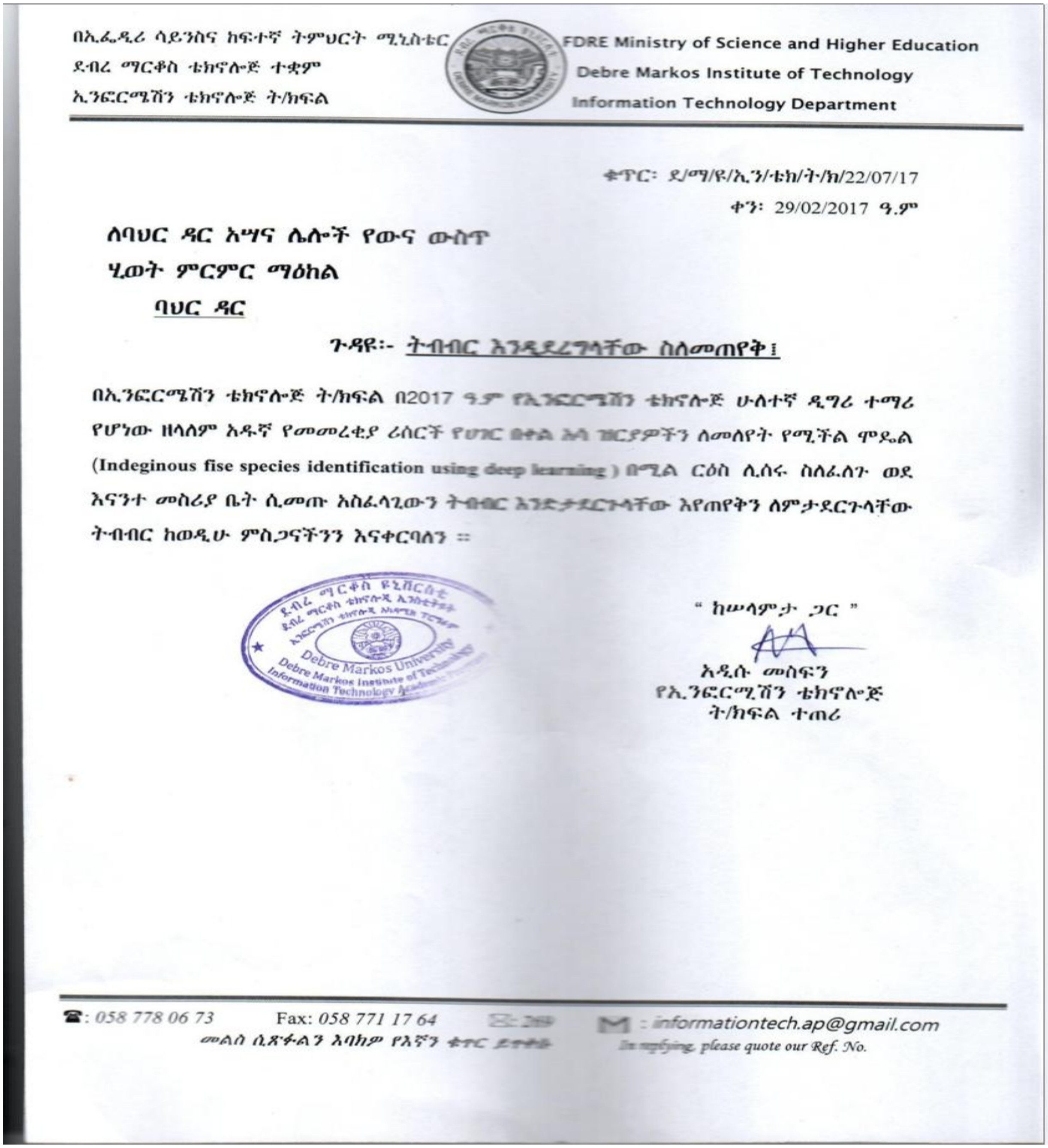
[58] B. Zoph and Q. V Le, "Searching for activation functions," *6th Int. Conf. Learn. Represent. ICLR 2018 - Work. Track Proc.*, pp. 1–13, 2018.

[59] J. Doherty, B. Gardiner, E. Kerr, N. Siddique, and S. S. Manvi, "Comparative Study of Activation Functions and Their Impact on the YOLOv5 Object Detection Model BT - Pattern Recognition and Artificial Intelligence," M. El Yacoubi, E. Granger, P. C. Yuen, U. Pal, and N. Vincent, Eds., Cham: Springer International Publishing, 2022, pp. 40–52.

[60] S. Sharma, S. Sharma, and A. Anidhya, "Understanding Activation Functions in Neural Networks," *Int. J. Eng. Appl. Sci. Technol.*, vol. 4, no. 12, pp. 310–316, 2020.

Appendix

Appendix A: Cooperation Letter to Collect Data



በኢ.ፌ.ዲ.ሪ ሳይንስና ከፍተኛ ትምህርት ሚኒስቴር
ደብረ ማርቆስ ቴክኖሎጅ ተቋም
ኢንፎርሜሽን ቴክኖሎጅ ት/ክፍል

FDRE Ministry of Science and Higher Education
Debre Markos Institute of Technology
Information Technology Department

ቁጥር፡ ደ/ማ/ዩ/ኢ.ን/ቴክ/ት/ክ/22/07/17
ቀን፡ 29/02/2017 ዓ.ም

ለባህር ዳር አሣና ሌሎች የውና ውስጥ
ሂወት ምርምር ማዕከል
ባህር ዳር

ጉዳዩ፡- ትብብር እንዲደረግላቸው ስለመጠየቅ፤

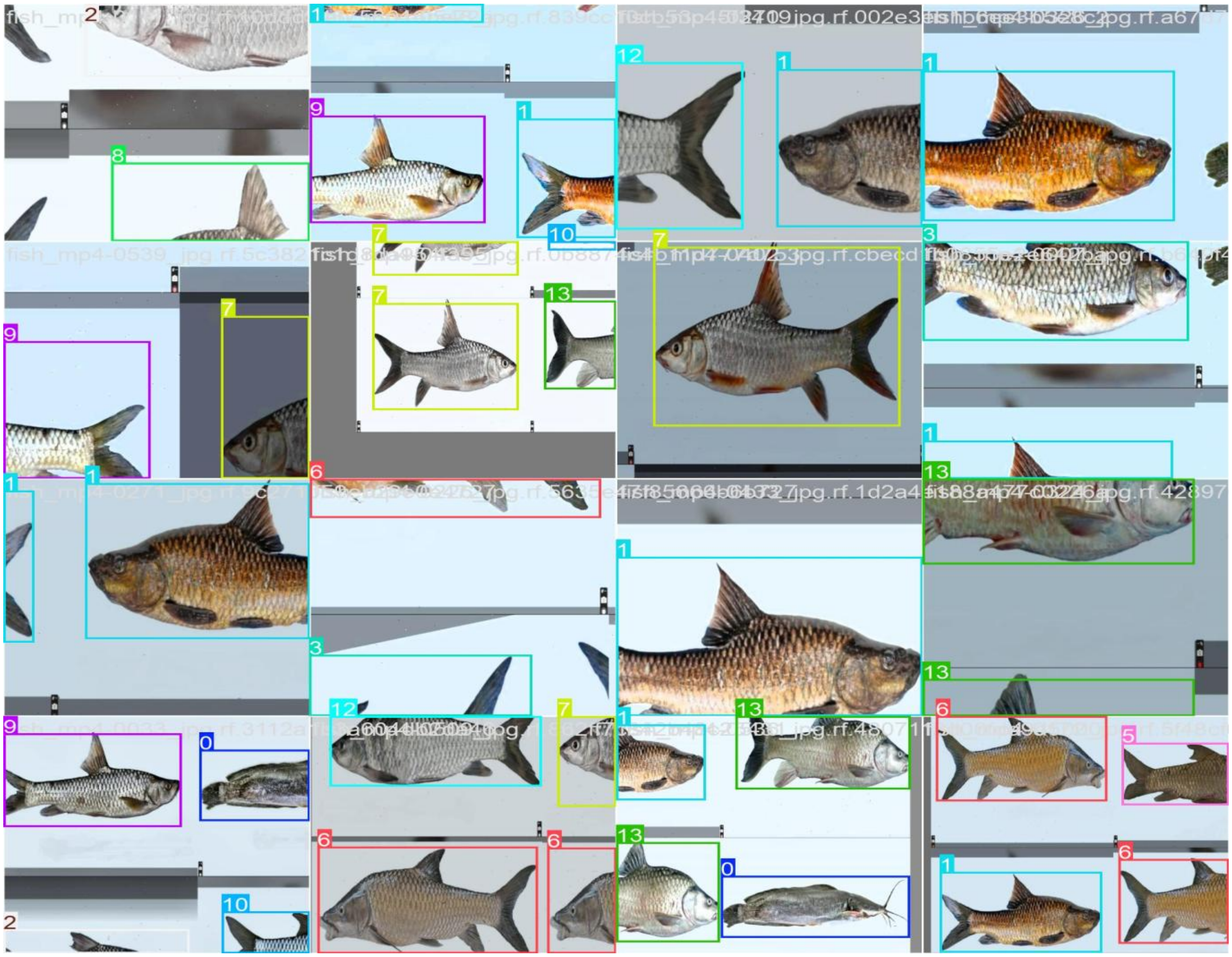
በኢንፎርሜሽን ቴክኖሎጅ ት/ክፍል በ2017 ዓ.ም የኢንፎርሜሽን ቴክኖሎጅ ሁለተኛ ዲግሪ ተማሪ የሆነው ዘላለም አዱኛ የመመረቂያ ሪሰርች የሀገር በቀል አሳ ዝርያዎችን ለመለየት የሚችል ሞዴል (Indeginous fise species identification using deep learning) በሚል ርዕስ ሊሰሩ ስለፈለጉ ወደ እናንተ መስሪያ ቤት ሲመጡ አስፈላጊውን ትብብር እንድታደርጉላቸው እየጠየቅን ለምታደርጉላቸው ትብብር ከወዲሁ ምስጋናችንን እናቀርባለን ፡፡

" ከሠላምታ ጋር "

አዲሱ መስፍን
የኢንፎርሚሽን ቴክኖሎጅ
ት/ክፍል ተጠሪ

☎: 058 778 06 73 Fax: 058 771 17 64 ✉: 269 M : informationtech.ap@gmail.com
መልስ ሲጽፉልን እባክዎ የእኛን ቁጥር ይጥቀሱ In replying, please quote our Ref. No.

Appendix A: Train -batch in YOLOv11s Model



Appendix C: Last Five Epoch from trained Model

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     95/100       7.6G     0.2435     0.1507     0.9356          6        640: 100% 41/41 [00:24<00:00,  1.71it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.39s/it]
                   all        123        125      0.974       0.98      0.977      0.924

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     96/100      7.35G     0.2421     0.1451       0.93          7        640: 100% 41/41 [00:23<00:00,  1.75it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.34it/s]
                   all        123        125      0.975       0.98      0.977      0.925

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     97/100       7.6G     0.2423     0.1454     0.9328          7        640: 100% 41/41 [00:24<00:00,  1.65it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.31s/it]
                   all        123        125      0.976       0.98      0.977      0.926

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     98/100      7.67G     0.2342     0.1405     0.9198          7        640: 100% 41/41 [00:23<00:00,  1.75it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.40it/s]
                   all        123        125      0.979       0.98      0.977      0.925

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     99/100      7.61G      0.231      0.141     0.9216          7        640: 100% 41/41 [00:25<00:00,  1.63it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:01<00:00,  1.46it/s]
                   all        123        125       0.98       0.98      0.977      0.928

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
    100/100      7.66G     0.2303     0.1403     0.9253          7        640: 100% 41/41 [00:23<00:00,  1.71it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 2/2 [00:02<00:00,  1.10s/it]
                   all        123        125       0.98       0.98      0.977      0.923
```