An Efficient Object Detection and Localization Framework using Modified YOLOv5

Subrat Kumar Pradhan

Department of Computer Science and Engineering
National Institute of Technology Rourkela

An Efficient Object Detection and Localization Framework using Modified YOLOv5

Thesis submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Subrat Kumar Pradhan

(Roll Number: 119CS0550)

based on research carried out

under the supervision of

Prof. Anup Nandy

May, 2023

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Prof. Anup Nandy

May 11, 2023

Supervisor's Certificate

This is to certify that the work presented in the thesis entitled *An Efficient Object Detection and Localization Framework using Modified YOLOv5* submitted by *Subrat Kumar Pradhan*, Roll Number 119CS0550, is a record of original research carried out by him under my supervision and guidance in partial fulfillment of the requirements of the degree of *Bachelor of Technology* in *Computer Science and Engineering*. Neither this thesis nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

Anup Nandy

Dedication

I dedicate this thesis to all those who have supported and inspired me throughout my B.Tech journey. To my loving family, whose unwavering encouragement and belief in my abilities have been a constant source of strength. Your love and sacrifices have been the foundation upon which I built my dreams. To my incredible professors and mentors especially Prof. Anup Nandy sir whose guidance and expertise have shaped my intellectual growth and pushed me to strive for excellence. Your passion for knowledge and dedication to teaching have ignited my curiosity and fueled my pursuit of innovation. To my friends and colleagues, who have been my pillars of support and shared in both the challenges and triumphs of this research project. Your camaraderie and collaborative spirit have made this journey all the more rewarding. Finally, I dedicate this work to the countless individuals whose lives may be positively impacted by the findings and implications of this research. May it contribute to the advancement of knowledge and pave the way for a brighter future.

Signature

Declaration of Originality

I, *Subrat Kumar Pradhan*, Roll Number *119CS0550* hereby declare that this thesis entitled *An Efficient Object Detection and Localization Framework using Modified YOLOv5* presents my original work carried out as a undergraduate student of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the sections “Reference” or “Bibliography”. I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

May 11, 2023
NIT Rourkela

Subrat Kumar Pradhan

Acknowledgment

Presentation, inspiration and motivation have always played a crucial role in the success of my venture.

I'd like to thank Dr. Anup Nandy for providing me this wonderful opportunity to work under his supervision. Without his guidance, this venture would not have been possible. His direction and guidance in every part of the research made this effort fruitful. I'd also like to thank to the head of the department Dr. DP Mahapatra and all the esteemed instructors in the department of Computer Science and Engineering.

I feel to acknowledge my indebtedness and deep sense of gratitude to Archana Balmik for giving me kind support throughout the research work.

Finally I'd want to offer my heartfelt appreciation to my family and friends for their unwavering support and encouragement.

May 11, 2023
NIT Rourkela

Subrat Kumar Pradhan
Roll Number: 119CS0550

Abstract

Object detection is the process of identifying and classifying objects within an image or video stream, while localization involves determining the precise location and extent of the detected objects. The dependence on other computer vision techniques for support in many object detection systems, which results in slow and subpar performance, is a significant challenge. Image recognition, image generation, and object detection are just a few of the many components that make up computer vision. The major challenging task of object detection is to recognise items in a semi-structured environment with inadequate lighting, tiny, crowded objects, objects that are obscured by other objects, low light, reflecting surfaces, amorphous bodies and partial views of an object. Here, we've focused on single stage object detection techniques like YOLO and experimented on modification of YOLOv5 architecture to improve accuracy of the model as conventional YOLOv5 model fails to achieve it. In this project, we take an end-to-end approach to solving the object detection problem that is entirely based on deep learning. We have applied various image processing techniques and a proposed feature extraction method which is a hybrid of Autoencoder, Canny edge extraction and SIFT (Scale Invariant Feature Transform) to improve the accuracy of tiny and low light images. We have obtained best fit anchor box sets using genetic algorithm and kmean clustering which play a major role in enhancing efficient object detection with better precision. The suggested modified YOLOv5 model achieves mean Average Precision (mAP) of 95.8% and a total loss of 0.015% which is a very satisfying result. Modified YOLOv5 gives better mAP score, loss and fps (frames per second) compared to SSD Mobilenetv2 making it a better choice for efficient object detection.

***Keywords*: *Object Detection*; *YOLOv5*; *Image Processing*; *Canny Edge Extraction*; *SIFT*; *Autoencoder*.**

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Object detection and localization are integral components of computer vision that enable the identification and precise positioning of objects within images or video streams. Object detection involves detecting the presence and classifying multiple objects of interest, while localization focuses on accurately delineating their spatial boundaries through bounding boxes or pixel-level segmentation. By combining advanced techniques such as deep learning and convolutional neural networks, object detection and localization algorithms have achieved remarkable progress in accurately and efficiently locating objects in diverse environments. These technologies find applications in fields such as autonomous vehicles, surveillance systems, image analysis, and robotics, facilitating tasks like real-time tracking, scene understanding, and interactive object manipulation. The ongoing research and development in this field continuously refine the accuracy, robustness, and speed of object detection and localization methods, paving the way for numerous innovative applications and advancements in computer vision [3].

Object detection has a rich history that spans several decades [4], with significant advancements in algorithms and techniques leading to the development of single-stage and two-stage detectors [5]. These detectors have revolutionized the field of computer vision, enabling accurate and efficient object detection in a wide range of applications. The history of object detection can be traced back to the early days of computer vision research when traditional computer vision techniques [6], such as edge detection, template matching, and feature extraction, were employed to identify objects in images. These methods relied on handcrafted features and rule-based approaches, which often struggled with variations in object appearance, scale, and viewpoint.

In the 2000s, researchers began exploring machine learning techniques for object detection. Boosted classifiers, such as AdaBoost, were introduced to learn discriminative features and classify objects. However, these methods were limited to detecting specific object categories and were not robust to variations in object appearance. The next major advancement in object detection came with the introduction of the two-stage detector, known as the Region-based Convolutional Neural Network (R-CNN), in 2013. R-CNN employed selective search to generate region proposals and then used a CNN to classify and refine these proposals. This approach significantly improved detection accuracy, but it was computationally expensive due to the need to process a large number of region proposals. To address the speed issue, researchers introduced the Fast R-CNN in 2015, which shared the convolutional features across all proposals [7], eliminating redundant computations. This approach improved both speed and accuracy. Building upon this, the Faster R-CNN was proposed in 2015, which introduced a Region Proposal Network (RPN) to generate region proposals directly from the convolutional features, further boosting the speed of object detection [8].

Single-stage and double-stage detectors are two popular approaches in the field of object detection, each with its strengths and weaknesses. These methods differ in their architecture and how they handle the task of detecting objects within images or video frames. Single-stage detectors, also known as one-stage detectors, aim to detect objects in a single pass of the network without the need for an additional region proposal step. These detectors directly predict the bounding box coordinates and class probabilities for all potential objects present in an image. Prominent single-stage detectors includes SSD (Single Shot Detection) and You Only Look Once (YOLO) family, which includes YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5 etc [9].

Single-stage detectors are known for their simplicity and efficiency [10]. They can achieve real-time object detection due to their direct approach of predicting bounding boxes and class probabilities in a single shot. These detectors are particularly suitable for scenarios that require fast inference times, such as video analysis and real-time applications. However, single-stage detectors often struggle with accurately detecting small objects or objects with complex shapes due to their limited spatial resolution and receptive fields. On the other hand, double-stage detectors, also referred to as two-stage detectors, follow a multi-step approach for object detection. In the first stage, these detectors generate a set of potential object proposals or regions of interest (RoIs). These proposals act as candidate bounding boxes that might contain objects of interest. In the second stage, these proposed regions are classified and refined to obtain accurate object detection. Examples of double-stage detectors include the Region-based Convolutional Neural Network (R-CNN) family, such as Faster R-CNN and Mask R-CNN. The choice of which method to use depends on the specific requirements of the application, balancing factors such as speed, accuracy, and computational resources. Ongoing research continues to explore new techniques and hybrid approaches to further advance the field of object detection.

Low light and small object detection are challenging tasks in computer vision that require specialized techniques to overcome the difficulties posed by limited illumination and the small size of the objects of interest [11]. These tasks find applications in surveillance systems, nighttime imaging, robotics, and many other fields where capturing and analyzing objects in challenging conditions is essential. In low light conditions, images suffer from reduced illumination, resulting in low contrast, increased noise, and loss of details. Detecting objects accurately in such environments becomes challenging due to the limited visual information available. Traditional object detection algorithms designed for well-lit conditions may fail to perform adequately in low light scenarios.

In general for efficient object detection YOLO is preferred over SSD due to it's accuracy and speed [12]. To address the challenges of low light and small object detection, several techniques have been developed. One approach is to enhance the image quality by applying image processing techniques specifically designed for low light conditions. These techniques include denoising algorithms, contrast enhancement, and illumination normalization. By improving the image quality, it becomes easier for object detection algorithms to operate effectively. Other approach is to modify the architecture of default model to enhance better object detection with high precision and speed. In this project we focused on modifying default YOLOv5 model to enhance it's mAP score and inference speed.

1.1 Objective

1. To design an efficient and improved model for object detection with high precision and fast execution.

2. To investigate and analyze the impact of different images processing and feature extraction techniques on the proposed model's performance.

.

Chapter 2

Literature Review

Huang et al. [13] proposed a technique for the detection of tiny objects that is based on an upgraded version of YOLOv5. This was done in light of the fact that the majority of object identification algorithms have a low degree of accuracy when it comes to detecting very minute objects. The basic feature extraction network of YOLOv5 has been updated in such a way that it now creates four feature pictures. This change was made in order to improve the results obtained while extracting features from the first photos used as input.It is possible to increase the effectiveness of microscopic item recognition by making modifications to the YOLOv5 Neck component, combining it with FPN and PANet, and performing feature fusing on four feature maps that include a range of semantic information. The fundamental procedure was altered such that it now makes use of the GIoU loss function in addition to the IoU loss function. This adjustment was implemented in order to increase the placement accuracy of extremely tiny objects, which prompted the need for the change. In order to better retain target qualities, the traditional ReLU activation function was swapped out for the Swish activation function. The modified YOLOv5 algorithm, the learning rate cosine annealing attenuation training approach, and the Mosaic data enhancement method were all applied in conjunction with one another to enrich the object recognition background, respectively. The improved YOLOv5 method was used so that the degree of freedom that could be exercised over the learning rate parameters could be increased. During the course of this inquiry, a comparison test using the CityPersons data collection and the first iteration of the YOLOv5 algorithm will be carried out.

Teng et al. [14] proposed a real-time AUV underwater detection strategy to enhance the ability of AUVs to independently differentiate rubbish while operating underwater using a modified YOLOv5s architecture. The upgraded network architecture includes a re-clustering of the anchor box using the KMeans++ clustering method to improve the ground truth feature information. To improve the accuracy of the box loss regression function and the fit of the forecast box to the ground truth box, CIoU was swapped for GIoU in the original model. The improvement in the forecast box's fit to the ground truth box allowed us to achieve these two aims. The test set training results showed that the improved network model could accurately and quickly identify plastic waste located at the bottom of the ocean. Minimum acceptable identification recall, accuracy and overall accuracy are 97.8%, 87.2%, and 90.6% respectively. Recognising each picture takes around 0.29 seconds on average. The revised YOLOv5s model utilised in this study had a positive impact on the detection of underwater real-time garbage tracking, as shown by the experimental results presented here.

Benjumea et al. [15] investigates how the well-known YOLOv5 object detector can be improved to better identify smaller objects, with a focus on automated racing. Start using the popular YOLOv5 object detector if you want a good jumping-off point. An fantastic place to begin your inquiry, this innovative piece of technology use AI to identify objects by their unique signals. This is done by measuring how changing individual parts of the model's structure (together with their linkages and other features) affects the speed with which inferences may be made and the total amount of time they take to complete. To do this, it proposes a collection of models over a variety of dimensions that it calls ”YOLO-Z”. These models show up to 6.9% more mAP than the original YOLOv5 model when differentiating smaller objects at 50% IOU, and they do so with just 3ms more inference time. Adding more time to the inference process led to this improvement. These findings will be used to enhance current systems so they can better recognise very small things in scenarios when current models are entirely ineffectual. This might significantly benefit a driverless racing automobile since it would likely boost the detection range and perceptual resilience of an automated vehicle, leading to enhanced planning and decision-making.

Jung et al. [16] evaluated a model for object detection in challenging situations. This photo was taken with a drone under settings that made it hard to distinguish individual items. Poor weather, heights, and backgrounds were these circumstances. It also improved detection and identified things in these circumstances. Testing uses the YOLOv5 architecture. The upgraded YOLOv5 model was compared to the original. Training picked the best YOLOv5 modified model. After that, the YOLOv5 modified model is tested with the best validation weight. The improved YOLOv5 model's accuracy has grown to 0.9% due to the mAP's improvements. To compare more accurately, the most relevant indicators were calculated using YOLOv3 and YOLOv4. The initial YOLOv5 model was more than 1.6 percent off from mAP, YOLOv3, and YOLOv4.When training started, the loss function closure speed of the YOLOv5 modified model was slower than the original. The two models differed greatly.

Kim et al. [17] proposed a method for improving the annotations of the SMD (Singapore Maritime Dataset) dataset and give an updated version of the dataset, which they have nicknamed SMD-Plus, in order to serve as a benchmark for DNN methods. The authors of this study are Kim and his colleagues. In addition to this, it suggests several individualised improvement strategies for the SMD-Plus. If we put this information to use and apply it to the issue at hand, we may be able to come up with a solution that is not only practical but also successful. YOLO-V5 employs not just the conventional methods of augmentation, but also an additional method known as the mix-up method. The modified YOLOv5 that was fitted with the SMD-Plus performed far better than the original YOLOv5 in terms of both detection and classification, as shown by the results of the experiments.

Emine et al. [18] focused on categorising poisonous species of mushrooms, and the results of their labour can be seen in the dataset in the form of the most dangerous mushroom varieties. Prior to the image being converted to the YOLO format, the labelImg tool was used in order to assign information to the picture. The purpose of the study was to distinguish between eight separate varieties of poisonous mushrooms. The names of these varieties are as follows: "Autumn Skullcap" (Galerina marginata), "Destroying Angels", "Conocybe Filaris", "Deadly Dapperling", "Death Cap", "Podostroma Cornu-damae", "Fly Agaric" and "Webcaps". These mushroom species images are trained utilising the cutting-edge fine-tuning tool, YOLOv5x, that is included with the YOLOv5 platform. According to the findings of our investigation into the academic literature, there have been no attempts made to identify and categorise poisonous mushrooms. Because of this, it has not been possible to compare its results to those of other research and serve as a standard for comparison. Despite this, it has been shown to be beneficial to use a precision-recall curve in conjunction with other measurements such as average precision and mean average accuracy. The mean and average accuracy across all courses is 0.77, with individual AP scores are 0.818, 0.825, 0.61, 0.737, 0.826, 0.854, 0.993 and 0.556 respectively.

Wu et al. [19] focused in a computer simulation on a CARLA vehicle equipped with distance detecting equipment. To improve upon the current YOLOv5s neural network architecture, this research proposes a new architecture called YOLOv5-Ghost. Yolov5s' network layer structure has been revamped. Since the suggested neural network design requires less computing power, it is more suited for embedded devices. Detection accuracy for the YOLOv5s is 83.36 percent mean average precision (mAP) after testing the updated network topology, and the detection rate is 28.57 frames per second. When compared, the YOLOv5-Ghost has a detection speed of 47.62 frames per second and an accuracy of 80.76 percent mean average precision. By analysing the images taken by the monocular camera inside the CARLA simulated environment, the study also calculates the distance covered by each vehicle. On average, there is a 5% disparity in length.

Ting et al. [20] proposes an upgraded version of the YOLOv5 network as a solution to the issue of insufficient feature extraction in the ship identification systems that are currently in use, which is caused by uneven feature distribution. In particular, the Ghostbottle module was employed to replace and fuse the feature extraction component of the YOLOv5 network, which ultimately led to enhanced experimental outcomes in comparison to those that were produced with the initial YOLOv5 network. The mean value of the loss function exhibited a more consistent pattern of reduction, and the maximum achievable percentage (mAP) reached 99.8%, which is 2.2 percentage points higher than the baseline network. The most essential YOLOv5 measures, mAP and GIoU, have been significantly improved upon as a result of enhancements made to the original YOLOv5 network, which have shown that these enhancements have been made.

Yan et al. [21] proposed a lightweight fruit target real-time identification based on enhanced YOLOv5 for use by an automated apple picker. This method was proposed so that a picking robot could automatically identify which apples were reachable and which were not from a photograph of an apple tree. In order to achieve the lightweight enhancement of the network, the BottleneckCSP module, which was utilised to replace BottleneckCSP in the backbone architecture of the original YOLOv5s network, was better built to become the BottleneckCSP-2 module in the improved designed network architecture. This action was taken to guarantee the network's optimal performance. In order to better capture the functioning of apple targets in a wide range of settings, the SE module has been integrated to the newly built backbone network. Using feature maps in bonded fusing mode, the target detection layer of the target detection system more precisely identifies apple targets. Based on the test set's detection findings, it's clear that the proposed improved network model can successfully implement the identification of fruits that a picking robot can grip but which are now out of reach in the displayed picture of an apple tree. Overall, the F1 score was 87.5%, the mAP was 86.75%, the recall was 91.48%, and the accuracy was 83.83%. Each picture took an average of 0.015 seconds to be recognised.

Based on the papers reviewed above we concluded that low light and small object detection are challenging tasks in computer vision that require specialized techniques to overcome the difficulties posed by limited illumination and the small size of the objects of interest.In low light conditions, images suffer from reduced illumination, resulting in low contrast, increased noise, and loss of details. Detecting objects accurately in such environments becomes challenging due to the limited visual information available. Traditional object detection algorithms designed for well-lit conditions may fail to perform adequately in low light scenarios. To overcome this challenge the default architecture of YOLOv5 model is modified with fine tuning of hyperparameters. In addition, various feature extraction techniques are necessary to enhance better object detection and localization. Hence in this project we are focusing on improving the architecture of YOLOv5 model which are trained with our prepared custom dataset of low light and other environmental conditions.

Chapter 3

Methodology

In this research, we implemented modified YOLOv5 model having certain architectural changes and fine tuning of hyperparameters. Figure 3.1 shows the flow chart of our suggested model, which is followed by a description to further explain our approach.

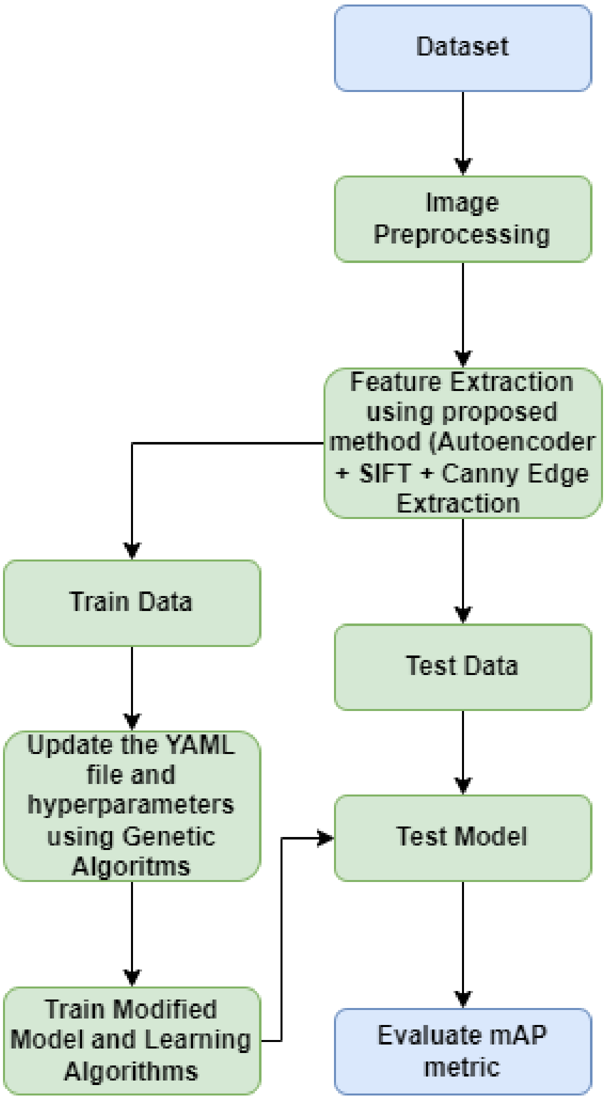

Dataset
Image Preprocessing
Feature Extraction using proposed method (Autoencoder + SIFT + Canny Edge Extraction
Train Data
Test Data
Update the YAML file and hyperparameters using Genetic Algoritms
Test Model
Train Modified Model and Learning Algorithms
Evaluate mAP metric

Figure 3.1: Proposed Model Workflow

The images were first acquired, then a custom dataset with eight different classifications was constructed. Finally, we employ a variety of data augmentation strategies to increase the number of images, hence enhancing the dataset's enrichment and the precision of the suggested model. To improve the size and quality of our dataset so that our model can produce better results, we also use feature extraction and data/image preparation techniques. The data is then split into training and testing groups. The learning model is then supplied with the training data as an input. We are doing supervised learning since we are using labelled data. The evaluation of the detection procedure is then done using the test data. The model is then tested using the test data, and the results are shown as mean average precision (mAP).

3.1 Dataset

The major objective of the proposed research project is to recognise items in a semi-structured environment with inadequate lighting, crowded objects, objects that are obscured by other objects, low light, reflecting surfaces, amorphous bodies, and partial views of an object. This was one of the biggest difficulties we ran across when carrying out this experiment. For this project, a custom dataset has been prepared. The dataset consists of eight different object classes: spoon, orange, apple, cup, cricket bat, bottle, wrist watch and banana. By taking pictures of the things and obtaining the photographs from Google, we were able to get 90 positive examples from each class with aggregating total of 720 images. The Dataset is divided into train and test set of 80 : 20 ratio.

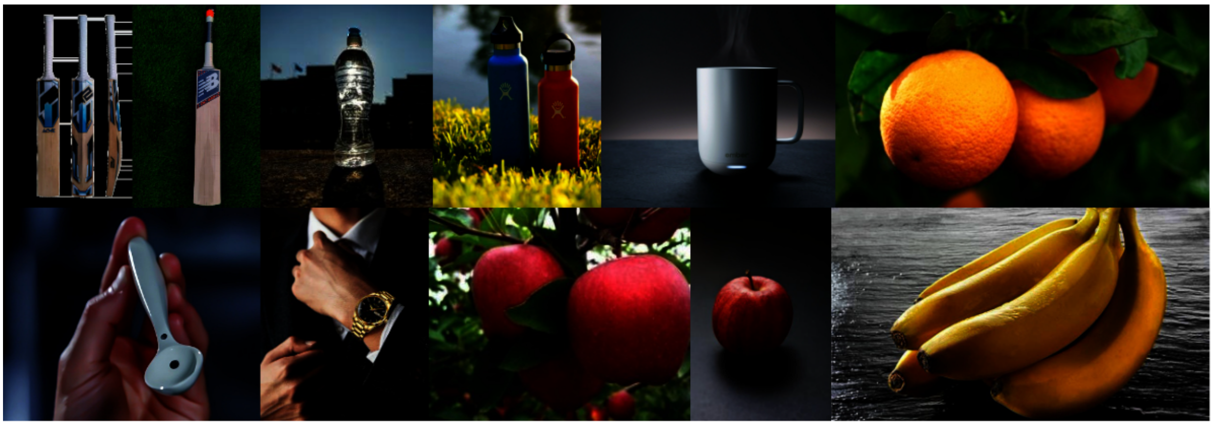


Figure 3.2: Sample Images of Proposed Dataset

The sample representation of the custom dataset proposed in this research is shown in Figure 3.2. It displays some representative images from the dataset that we have compiled. We must resize the data because the images we have acquired are of various sizes or resolutions. The entire dataset's image is then scaled to 640 X 640 pixels.

3.1.1 Data Annotation

Annotations means labelling of images. Depending on the various techniques used, many annotation shapes may be used to annotate a picture. Annotation methods like lines, splines,landmarking ,polygons and rectangular boxes are used for image annotations. We have used rectangular box annotation approach to label the data. Figure 3.3 represents how the annotation is done in this proposed method.

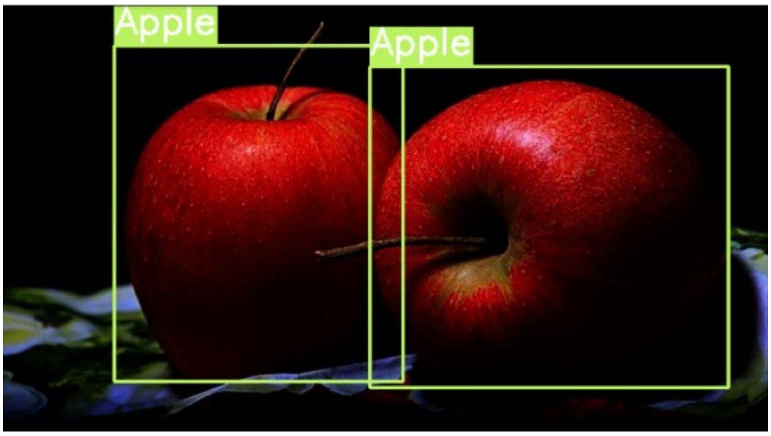


Figure 3.3: Data Annotation

For YOLOv5 model , annotations are done in YOLO format and for SSD MobileNetv2, PASCAL VOC format is used for labelling images.With the YOLO labelling format, a .txt file with the same name is produced for each picture file in the same directory. Each .txt file provides the annotations for the matching picture file, including the object class, coordinates, height, and width. In PASCAL VOC format an XML file for every image is generated that has all the informations regarding the bounding box as well as their class labels. The annotation here is done in PASCAL VOC format.

3.2 Pre-Processing Techniques

3.2.1 Data Augmentation

Data augmentation refers to a range of techniques for increasing the quantity and quality of training datasets so that efficient Deep Learning models may be developed. To create viable Deep Learning models, the training error and validation error must both decline simultaneously. The best method to do this is via data augmentation. The gap between the training and validation sets, as well as any future testing sets, will be reduced by the enriched data's inclusion of a larger variety of data points.The network performs better at classification and runs less of a chance of overfitting as data is added. Use of data augmentation may be accomplished in two ways. The first strategy entails enhancing the dataset as a whole using data-augmentation methods before storing the photos inside [22]. Moreover, you may employ the data augmentation phase after each training phase.In this proposed method we have implemented following data augmentation methodologies:

- Horizontal and Vertical Flipping
- Rotation, scaling and resizing of images
- Translation of images
- Cropping and padding of images
- Adding random gaussian noise to images
- Altering the color attributes of an image, such as brightness, contrast, saturation, can help models become more tolerant to changes in lighting conditions
- Applying elastic transformation to images which simulates the elastic properties of objects

3.2.2 Image Processing

Image processing is the process of applying operations on a picture in order to enhance or retrieve significant information from it. Signal processing is a subset of image processing. A picture serves as the input, and the result might be that image, or elements of it. The picture in the dataset includes noise, thus image processing is crucial. So, by using different image processing techniques, we can eliminate the unnecessary information and only retain the valuable information. It aids in dimension reduction as well.

Histogram Equalization

The Histogram Equalization method is used to boost picture contrast in order to improve image readability, which will enable our model to train more successfully and provide results that are more effective. It is the most often used strategy due to its simplicity and effectiveness. It works by remapping the grey levels of the picture in accordance with their probability distribution. The main assumptions of histogram equalisation is to re-map the provided intensity values of pixels to a target value in order to enhance the contrast of the image or the quality of its information. This is done by using picture intensity statistics, or more specifically, the probability density function (pdf). A cumulative density function may be used to do this by controlling the migration of intensities (cdf).Figure 3.4 will show us the difference between two images before and after the implementation of histogram equalization [23].

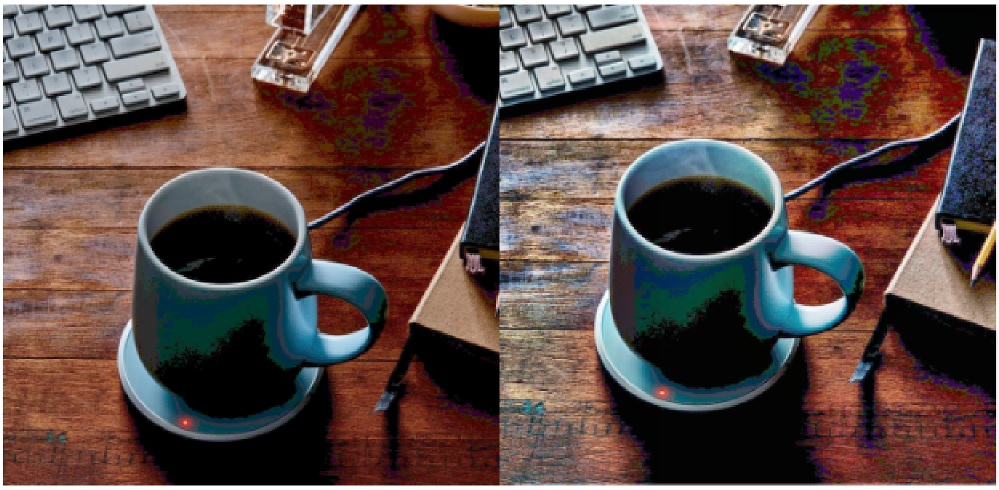


Figure 3.4: Before and after Histogram Equalization

Gamma Correction

Gamma correction is a non-linear modification to each pixel's value. Although we performed linear operations on individual pixels during picture normalization, such as scalar multiplication and addition/subtraction, gamma correction performs a non-linear operation on the pixels of the original image and may result in saturation of the transformed image. If the gamma value is too big or too little, it may also result in poor contrast [24].Figure 3.5 shows how the images after applying Gamma Correction transformation look like.

Consider any random pixel on a display device for a more analytical mathematical study. The intensity value we wish to see on the screen (in our code) is specified as x, but the intensity value we actually see in gray-scale is defined as y. Lastly, gamma is used to define the γ value. The equation below may be used to explain the relationship between the input and output intensity levels.

$$y = x^{\gamma} \tag{3.1}$$

We can account for the gamma correction brought on by the display if we do the opposite of this operation in advance, before showing the pixel on the screen. This is equivalent to

$$y = (x^{\gamma})^{1/\gamma} \tag{3.2}$$

3.2.3 Feature Extraction

A method for shrinking the size of a picture in which a significant number of its pixels are represented while yet effectively retaining essential information is called feature extraction. The primary goal of feature extraction techniques is to retain pertinent data while removing unimportant data from the picture. If we supply the model with the data after feature extraction, which only includes key information, as opposed to the data in its raw form, the

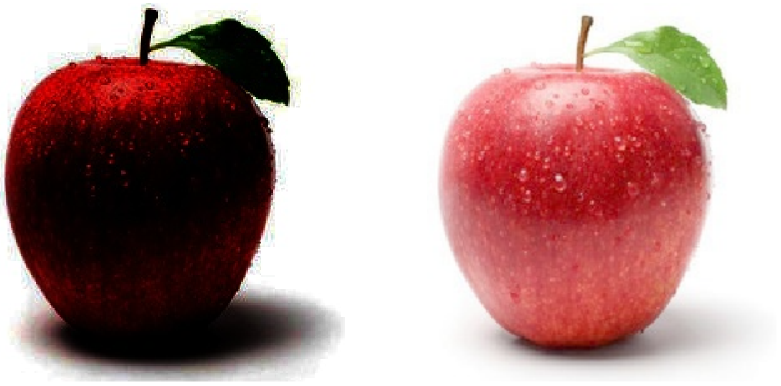


Figure 3.5: Before and After Gamma Correction

model's accuracy improves. As opposed to learning from raw data, the model requires less work to provide results since feature extraction helps reduce redundant data from the picture. As a result, it is simpler to detect objects. Also, by simplifying the data representation and reducing complexity via feature extraction, each variable in the feature space is represented as a linear combination of the initial input variable. Canny edge extraction is a feature extraction technique that we used in this work.

Canny Edge Extraction

The extraction of edges from digital pictures is a common use of the image processing technology known as canny edge detection. It was invented in 1986 by John Canny, and ever since then, it has grown to become one of the algorithms for edge detection that is used the most often owing to its efficiency and reliability. The method of edge detection known as Canny is broken down into multiple stages. The first thing that has to be done to get rid of the noise in the picture is to run it through a Gaussian filter. The picture is made smooth by the use of a Gaussian filter, which works by convolving it with a Gaussian kernel. This helps get rid of high-frequency noise while keeping the edges intact.

After smoothing the picture, the next step is to determine the magnitude and direction of any gradients present in the image. This is accomplished by using the Sobel operation on the picture after it has been smoothed. Convolving the picture with two distinct kernels in the horizontal and vertical directions allows the Sobel operator to compute the gradient of the image. The amplitude of the gradient denotes the degree to which the edges are defined, while the direction of the gradient reveals the arrangement of the edges. The non-maximum suppression step is carried out by the algorithm once the size and direction of the gradient have been identified. This phase includes suppressing non-maximum pixels along the gradient direction in order to flatten down the edges. Only the local maxima in the direction of the gradient are kept, which guarantees that the margins will be thin and well defined [25].

The hysteresis thresholding process is the last one that the Canny edge detection method goes through. In order to do this, you will need to establish two different threshold values, a high threshold and a low threshold. Any pixel that has a gradient magnitude that is higher than the high threshold is regarded as having a strong edge, whereas pixels that have a magnitude that is lower than the low threshold are disregarded since they have weak edges. It is only when they are linked to strong edges that pixels with magnitudes that fall between the two criteria are regarded to have weak edges. Through this technique, bogus and weak edges are removed from the object

The process of canny edge extraction is used extensively in a variety of applications, including computer vision, picture segmentation, and object identification, amongst others. It is very successful at recognising crisp and well-defined edges, which makes it valuable for tasks such as edge-based image analysis, boundary detection, and feature extraction because of these abilities. Figure 3.6 shows the edge extraction by Canny edge extraction technique.

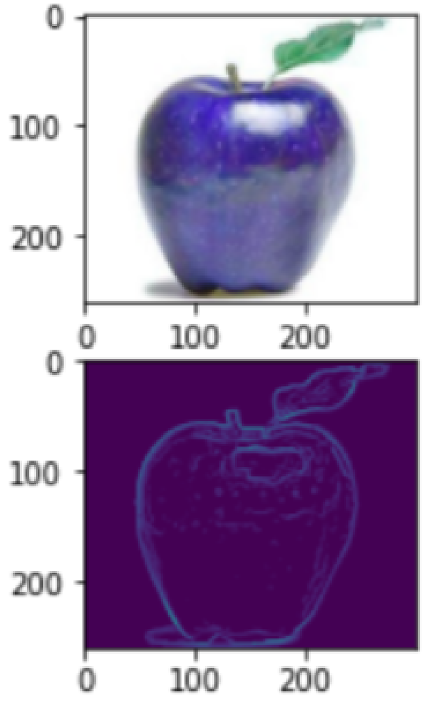


Figure 3.6: Edge extraction by canny edge detector

Proposed Feature Extraction Technique

We have proposed a new technique for feature extraction of images for improving accuracy. It is a hybrid of AutoEncoder, Canny edge extraction and SIFT (Scale-invariant feature transform) method.

1. **Autoencoder:** When it comes to denoising pictures, an autoencoder is a useful neural network design. An autoencoder's central principle is to encode an input picture into a lower-dimensional latent space and then use this encoded representation to

recreate the original image.Denoising pictures using an autoencoder requires training the network on pairs of clean and noisy images. To create a compressed representation of the picture in the latent space, the encoder takes raw images as input. From this compressed form, the decoder reconstructs the original picture.

In order to train an autoencoder, the difference between the reconstructed and original images is minimised as much as possible. This motivates the encoder to learn a compressed, noise-tolerant representation of the picture, while simultaneously training the decoder to recover the original, unnoised image. Figure 3.8 illustrates the detailed layers of autoencoder used here.

2. **Canny Edge Extraction:** This extraction method enables detection of edge. Edge detection is a technique that takes grey discontinuous points into account while identifying and segmenting pictures. In order to extract features for this suggested technique, we applied clever edge detection.

3. **SIFT (Scale Invariant Feature Transform) Method:** It in general consists of four steps.

   - **Scale-space extrema detection:** Finding the local extrema of the difference-of-Gaussian function in the picture across scale space constitutes this phase. To calculate the difference-of-Gaussian function, we convolve the picture using a sequence of scaled-down Gaussian filters.
   - **Keypoint localization:** Here we do keypoint localization, which entails selecting keypoints based on their contrast and edge responses and refining the extrema to sub-pixel precision.
   - **Orientation assignment:** The prevailing gradient orientation in a neighbourhood close to the keypoint is used to determine the canonical orientation given to the keypoint.
   - **Descriptor generation:** To conclude, a descriptor is constructed for each keypoint by generating a histogram of gradient orientations in a neighbourhood around the keypoint and weighting the histogram using a Gaussian function. Figure 3.7 illustrates our proposed feature extraction output distinctly.

3.3 Model

In this study we have proposed a modified YOLOv5 model having changes in architectural design as well as fine tuning of hyperparameters to increase the performance and accuracy while detecting small and low light images. Also we proposed SSD MobileNet v2 model for

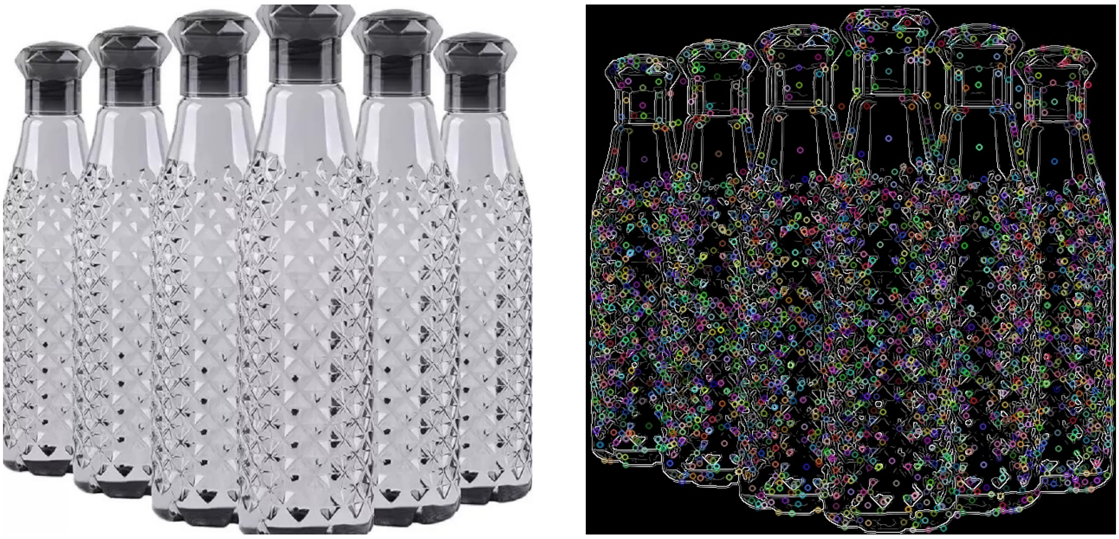


Figure 3.7: Before and After Proposed Feature Extraction Technique

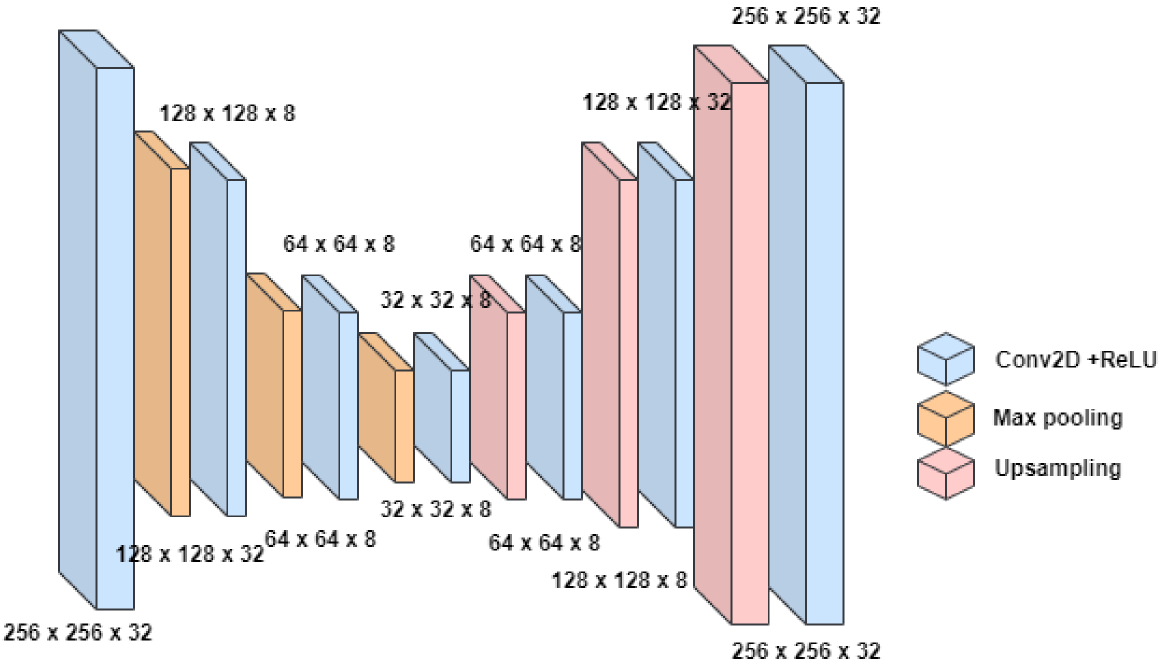


Figure 3.8: Model Architecture of Autoencoder

object detection task with fine tuning of hyperparameters. We compared the performances of both models in terms of accuracy and inference speed.

3.3.1 Conventional YOLO Model

The YOLO method, which has several versions, is a well-known object detecting technique. The whole picture may be trained instantly, and it is simple to apply. YOLO has grown steadily as a result. Very soon after the introduction of YOLOv4, YOLOv5 was made available in 2020. This solution offers comparable speed to YOLOv4 and uses the same concept. The key point of interest is that it is entirely developed in the PyTorch framework rather than using any aspect of the Darknet framework and that it has an emphasis on accessibility and usage in a broader variety of development environments. The models in YOLOv5 also show to be substantially smaller, quicker to train, and more adaptable for usage in a practical application. Speed and accuracy have both improved over the quick R-CNN. YOLO outperforms Fast R-CNN in terms of processing speed since it uses the same network for all candidate areas, rather than using a separate network for extracting them. Rapid R-usage CNN's of hand-crafted and deep convolutional features has difficulties in terms of identifying objects or people. In Figure 3.9, the backbone network portion, the neck part (PANet), and the head (output) part are shown as the three main components of the previous YOLOv5 structure.

3.3.2 Proposed modified-YOLOv5

Backbone

The component responsible for using the input picture and deriving feature maps from it forms the basis of a model.This phase is essential for any object detector since it contains the primary framework for both abstracting and extracting contextual data from the input picture.The object detection performances depend on image size, width and depth of the model. Additionally, to concentrate on recognizing certain feature maps, the neck and head's layer connections may be manually changed. We have added an extra C3 module in the backbone which includes CBS (Conv2D, BN, SILU (Sigmoid + ReLU), 1 BottleNeck with Concat. Unlike previous default architecture, here Leaky ReLU is replaced with SILU(Sigmoid + ReLU) which learns faster and better than ReLU due to its non linearity in nature. Adding an extra layer with suitable dimensions and connecting to Neck layer in exclusive manner by replacing with lower resolution featue maps help to improve accuracy without compromising the inference speed. SPP (Spatial Pyramidal Pooling) is replaced with SPPF (Spatial Pyramidal Pooling Fast) which reduces the number of FLOPS (Floating Point Operations per seconds) and improves speed of the model significantly.

Neck

We refer to the structure between the head and the backbone as the ”neck,” and it serves to collect as much of the data gathered by the backbone as possible before it is given to the head. By preventing small-object information from being lost to higher levels of abstraction, this structure plays a crucial role in the transmission of that information.

In this study, we propose to replace the existing Pan-Net with a biFPN and simplified it to be an FPN. The neck functions similarly in both situations, but the intricacy of the implementation necessitates different numbers of layers and connections due to the different

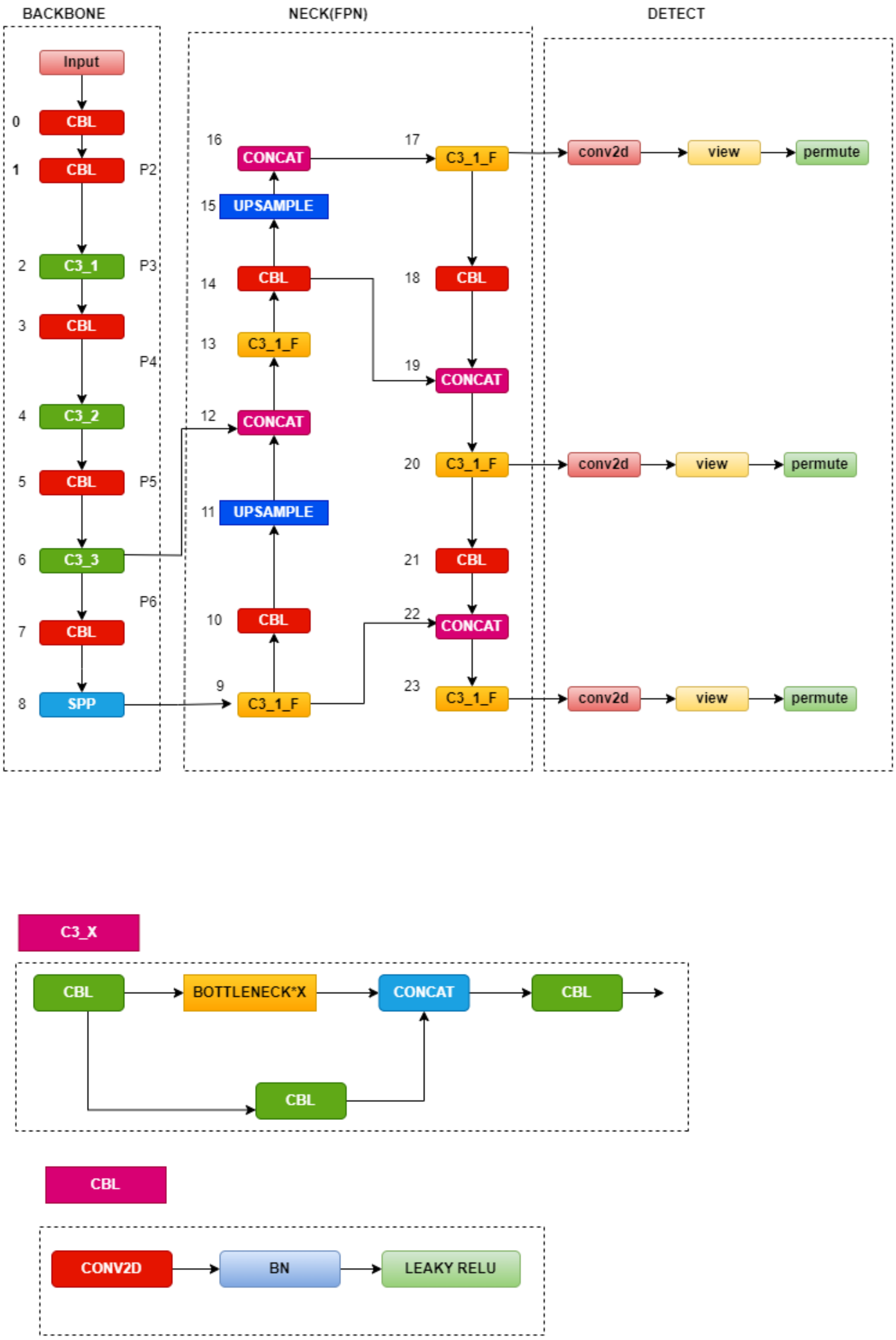


Figure 3.9: Network Architecture of default YOLOv5

complexity.In figure 3.10 biFPN architecture is clearly shown.

Other modifications

The amount of parameters will need to be changed to accommodate the new structure since it may impair the network's capacity for learning. Specifically, the dimensions of the anchor

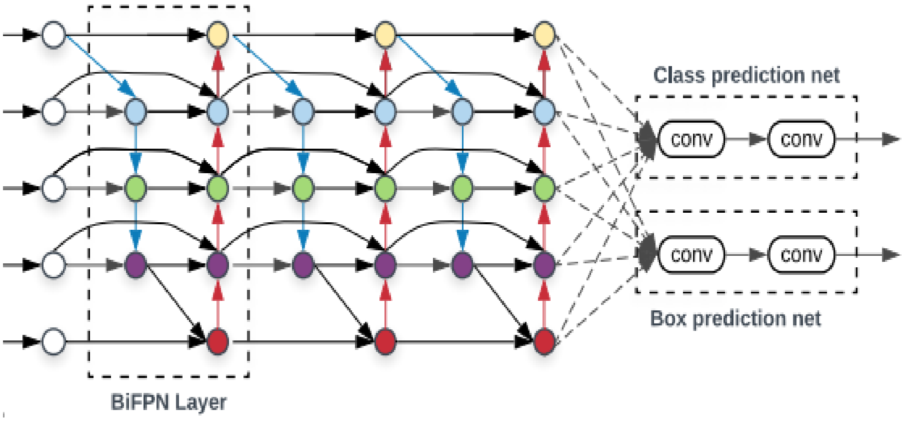


Figure 3.10: Network Architecture of BiFPN [1]

boxes used in the head, which must be adjusted to the quality of the feature maps being utilized. Other modifications like freezing some layers of the model will allow us to train in less GPU memory and may improve mAP score of the model as well.

3.3.3 Anchor Box Selection using Genetic Algorithm

A group of predetermined boundary boxes with a certain height and width are called anchor boxes. These boxes are often selected based on the item sizes in the training datasets and are created to capture the scale and aspect ratio of various object classes you wish to identify. It is crucial to use solid anchors since YOLO predicts bounding boxes indirectly, as displacements from anchor boxes. Naturally, neural networks predicts tiny displacements more effectively (more precisely) than they do massive displacements. Therefore, a neural network will have to perform less ”work” and create models that are more accurate the better the anchor boxes are chosen. Anchor box selection is done through following methods.

1. **Obtain bounding box dimensions from the train data :** The bounding boxes (labels) in all train photos’ height and breadth are important specifications. Pay close attention to the fact that the height and width of pictures that have previously been scaled (to fit the input size of the model) should be computed in pixels. Here by default input image size is taken as 640 X 640 which means all the images are resized to 640 ensuring preserving of aspect ratio and padding of shorter length side.

2. **Define anchor fitness using a metric :** The loss function, specifically the box loss, should be coupled to the metric in the ideal case. The better the metric, the smaller the loss. And if anchor boxes are chosen based on this parameter, the model immediately begins training with lesser loss. Since the evolutionary algorithm will utilise this metric, one may choose whatever metric he/she likes without worrying about the limitations imposed by other optimisation techniques. Following are the detailed steps for the algorithm used in defining anchor fitness.

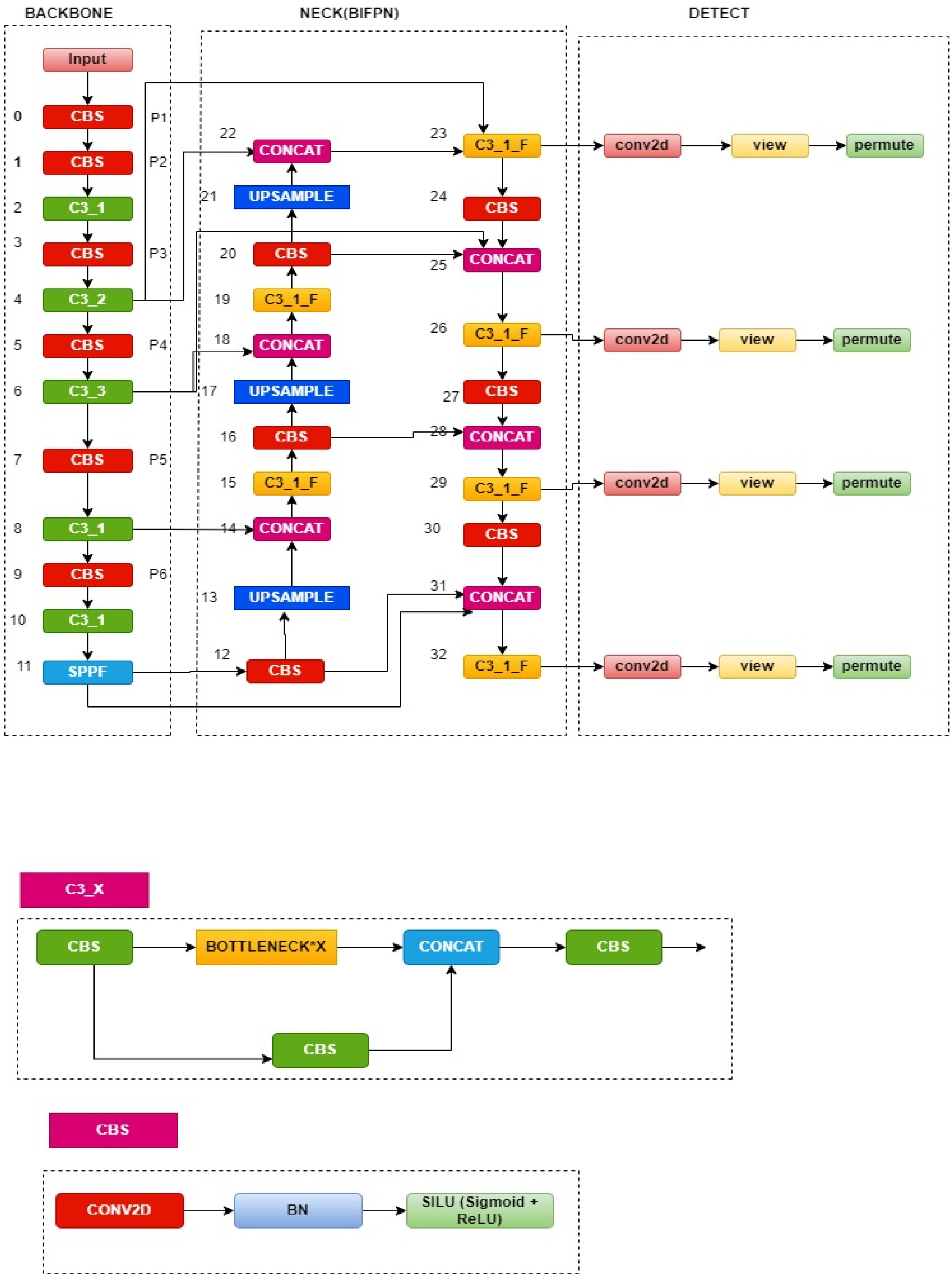


Figure 3.11: Modified Network Architecture of YOLOv5

- The anchor-t hyperparameter defines a threshold, which is typically 4 but may instead be interpreted as 1/anchor-t, or 0.25. If the difference between the anchor box and the bounding box label is less than four, we consider it to be an acceptable anchor box.

- As much as feasible, we'd want each labelled bounding box to be situated next to an anchor box. Also, it should not be more than a factor of four greater or lower than the threshold value.
- On average, excellent fitness is achieved, which suggests that certain bounding boxes (likely those that are considered to be anomalies) may still be located at a considerable distance from anchors even after they have achieved high fitness.
- However, we calculate the fit of each bounding box by beginning with the side that doesn't fit quite as well known as worse fitting. This allows us to choose the ideal anchor for each box. Figure 3.12 shows detailed fitness metric calculation steps for anchor box sets.

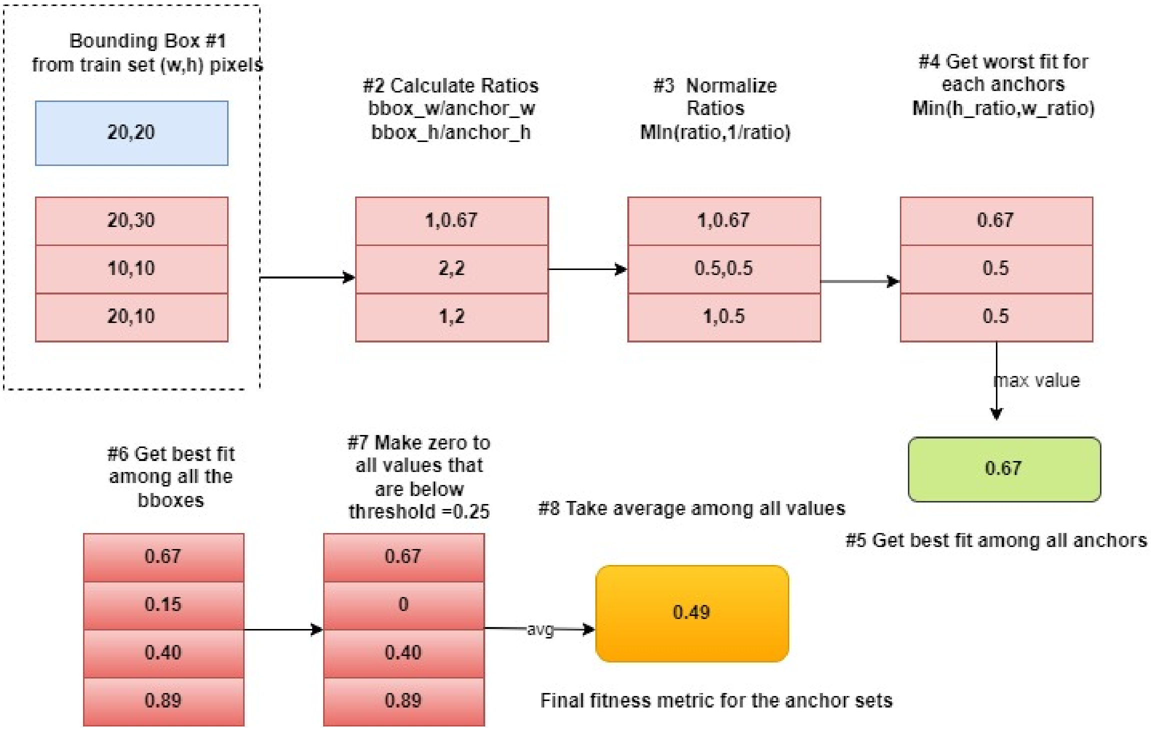


Figure 3.12: Fitness Metric Calculation for Anchor Boxes

3. **Making initial guess for anchors using K-means clustering :** All of the train set's bounding box identifiers are processed using a K-means clustering algorithm. The width and height of individual pixels are among the criteria used for clustering. The anchors denote the ultimate cluster centers.By default we assumed 5 clusters for our dataset.
4. **Enhance anchor fitness performance via evolving anchors:** Evolutionary algorithms take their cues from natural processes, and their inherent simplicity makes them rather sophisticated. Here evolution algorithm runs for 500 iterations to determine a suitable anchor . The k-means anchor set is used as a starting point,

and then the fitness measure is calculated after making modest, random changes to the size of chosen anchor boxes. If the next round of mutation using a new set of mutated anchors produces good results, the previous anchors are saved for use in future mutations. Figure 3.13 shows the flow diagram of evolutionary concept for improving anchor fitness.

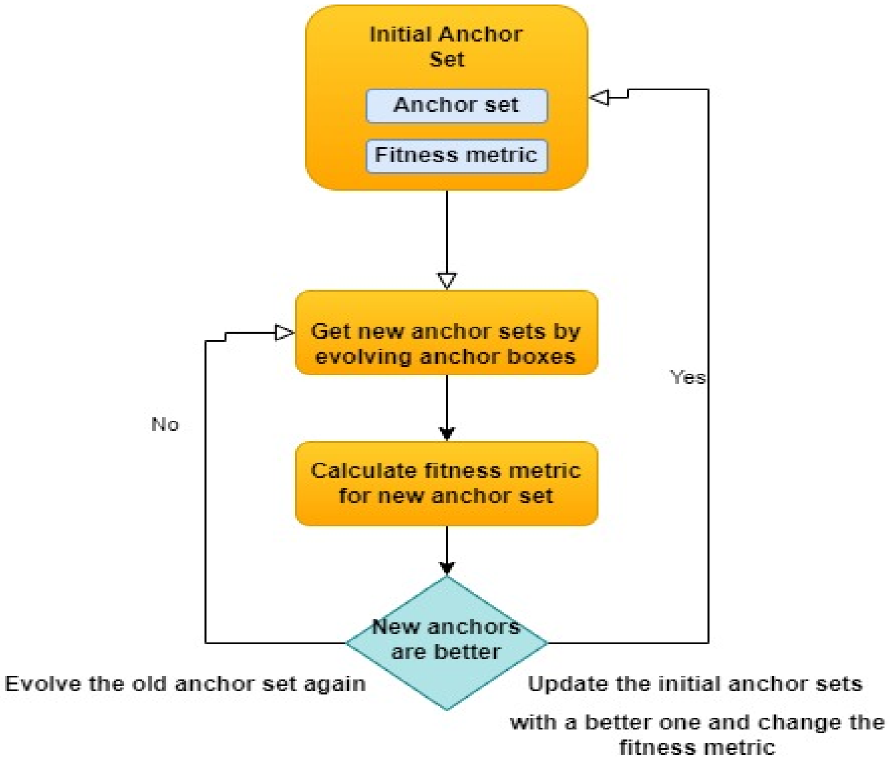
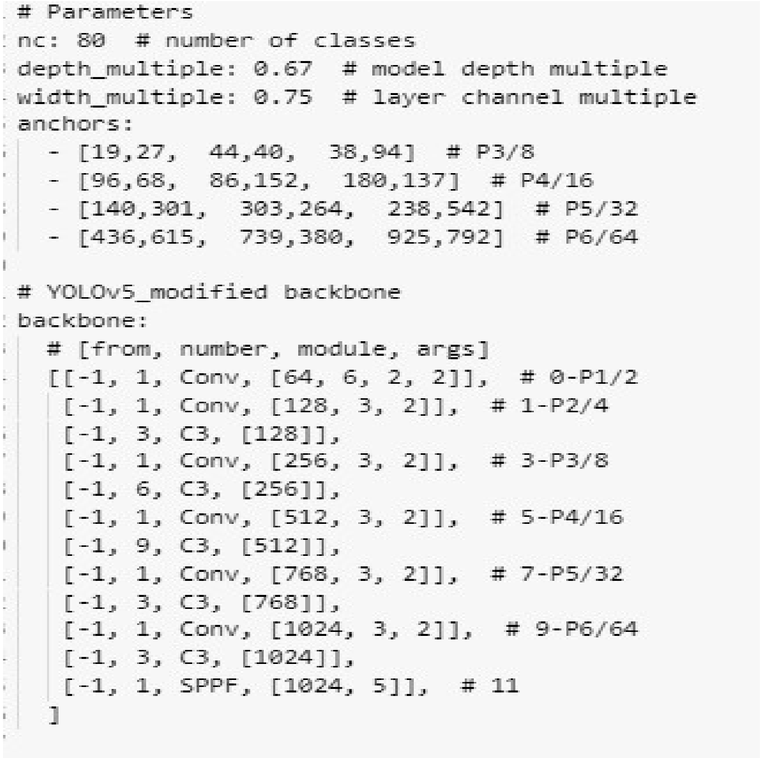

Initial Anchor Set
Anchor set
Fitness metric
Get new anchor sets by evolving anchor boxes
Calculate fitness metric for new anchor set
New anchors are better
No
Yes
Evolve the old anchor set again
Update the initial anchor sets with a better one and change the fitness metric

Figure 3.13: Evolutionary Algorithm to Improve Anchor Fitness

3.3.4 SSD MobileNetv2

Google substituted the Mobilenet network model for the VGG16-SSD detector to reduce the overall computing burden, which improved the SSD detector's real-time outputs. Six feature maps with a variety of shapes are provided by the VGG16-SSD detector, which also includes the front Mobilenet-v2 network, for the back-end detection network to use while performing multi-scale object detection. Because the core network model was modified from the VGG-16 network to the Mobilenet-v2 network, the Mobilenet-SSD detector is now capable of achieving real-time results and is speedier than other existing object detection networks.

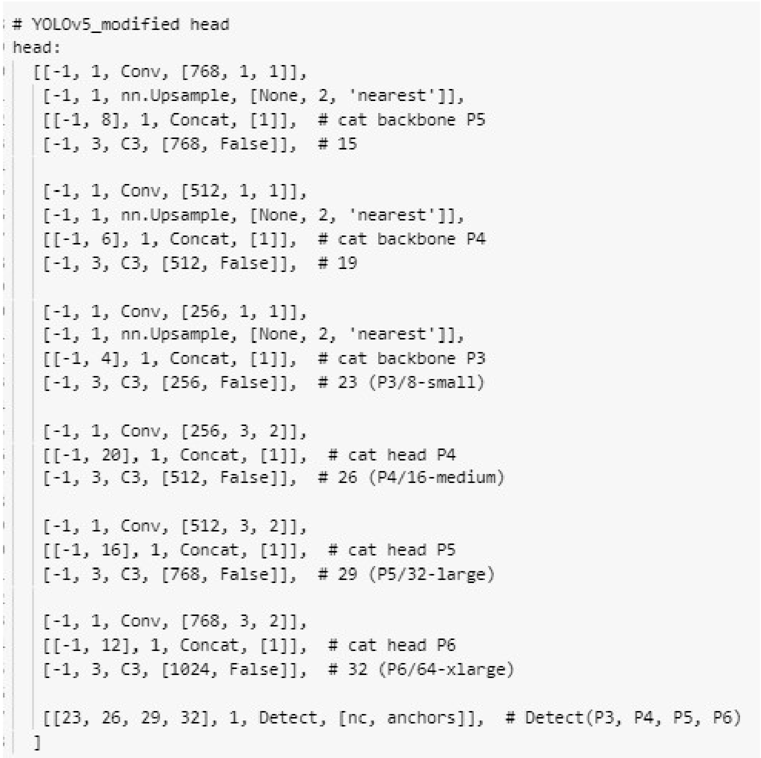
There are two different types of blocks in MobileNetV2. One of them is a one-stride residual block. Another choice for shrinking is a two-stride block. The two different types of blocks each have three levels. This time, a 1x1 convolutional network's first layer uses the ReLU6 activation function. Convolution in depth makes up the second layer. The third



```
# Parameters
nc: 80  # number of classes
depth_multiple: 0.67  # model depth multiple
width_multiple: 0.75  # layer channel multiple
anchors:
  - [19,27,  44,40,  38,94]  # P3/8
  - [96,68,  86,152,  180,137]  # P4/16
  - [140,301,  303,264,  238,542]  # P5/32
  - [436,615,  739,380,  925,792]  # P6/64

# YOLOv5_modified backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [64, 6, 2, 2]],  # 0-P1/2
   [-1, 1, Conv, [128, 3, 2]],  # 1-P2/4
   [-1, 3, C3, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
   [-1, 6, C3, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 5-P4/16
   [-1, 9, C3, [512]],
   [-1, 1, Conv, [768, 3, 2]],  # 7-P5/32
   [-1, 3, C3, [768]],
   [-1, 1, Conv, [1024, 3, 2]],  # 9-P6/64
   [-1, 3, C3, [1024]],
   [-1, 1, SPPF, [1024, 5]],  # 11
  ]
```

Figure 3.14: YAML file configuration of Proposed Modified YOLOv5 Model (Anchors and Backbone)



```
# YOLOv5_modified head
head:
  [[-1, 1, Conv, [768, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 8], 1, Concat, [1]],  # cat backbone P5
   [-1, 3, C3, [768, False]],  # 15

   [-1, 1, Conv, [512, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 6], 1, Concat, [1]],  # cat backbone P4
   [-1, 3, C3, [512, False]],  # 19

   [-1, 1, Conv, [256, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 4], 1, Concat, [1]],  # cat backbone P3
   [-1, 3, C3, [256, False]],  # 23 (P3/8-small)

   [-1, 1, Conv, [256, 3, 2]],
   [[-1, 20], 1, Concat, [1]],  # cat head P4
   [-1, 3, C3, [512, False]],  # 26 (P4/16-medium)

   [-1, 1, Conv, [512, 3, 2]],
   [[-1, 16], 1, Concat, [1]],  # cat head P5
   [-1, 3, C3, [768, False]],  # 29 (P5/32-large)

   [-1, 1, Conv, [768, 3, 2]],
   [[-1, 12], 1, Concat, [1]],  # cat head P6
   [-1, 3, C3, [1024, False]],  # 32 (P6/64-xlarge)

   [[23, 26, 29, 32], 1, Detect, [nc, anchors]],  # Detect(P3, P4, P5, P6)
  ]
```

Figure 3.15: YAML file configuration of Proposed Modified YOLOv5 Model (Head)

layer employs a 1x1 convolutional network once again, but this time non-linearity is not incorporated. The claim states that deep networks will only be able to conduct a linear classification method on the fraction of the result space with non-zero volume if ReLU is repeated. The SSD MobileNetv2 network architecture is shown in Figure 3.16.

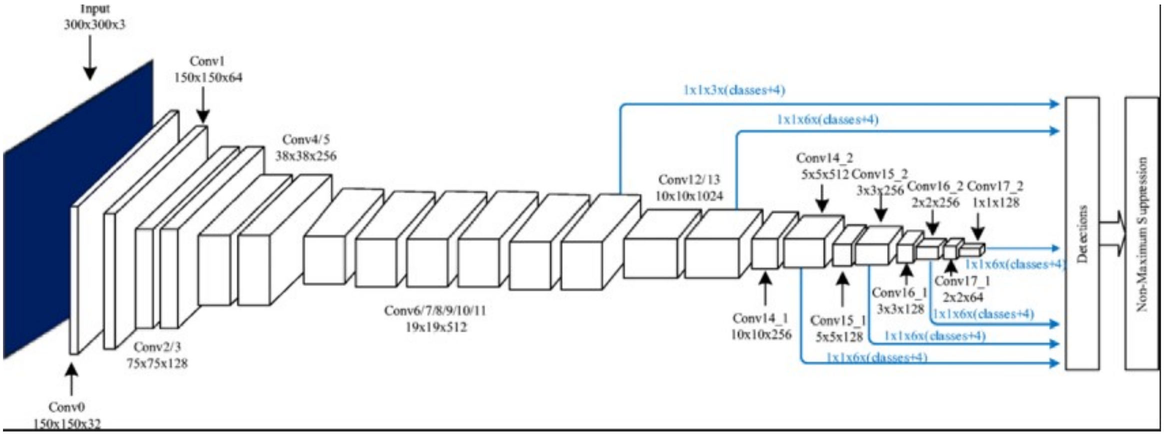


Figure 3.16: Network Architecture of SSD MobileNetv2 [2]

SSD is used for localization, whilst MobileNet is employed for feature extraction and categorization. Each class, the SSD finds 8732 predictions. It creates numerous bounding boxes per item for each prediction. Therefore, in order to choose the bounding boxes whose IOU value is higher than the threshold value, the Non-Max Suppression (NMS) approach must be used. Ultimately, the bounding with the greatest IOU value and confidence score will be selected for the object prediction. The IOU threshold in this model is set at 0.5.

Chapter 4

Results and Discussion

We performed several experiments to our proposed algorithm of modified YOLOV5 model. With a batch size of 16, the model has successfully trained across 25 epochs (20 steps each epoch).Stochastic gradient descent (SGD) optimiser with learning rate of 0.05 and 4 number of anchors are used to train this proposed model. We have recorded the performance metrics in the form of mean average precision(mAP), precision,recall and F1-score.

Our modified YOLOv5 model gives a mAP score of 95.8% at the IOU threshold of 0.5 which is quite satisfying .Figure 4.1 shows experimental results plot for mAP-0.5 scores of different yolov5 models which includes YOLOv5n,YOLOv5s,YOLOV5m,YOLOv5l and YOLOv5x. These are the initial results of conventional default YOLOv5 model with low light, noisy,small and occluded images. From this experiment we conclude that although large and extra larger versions of YOLOv5 gives better precision but these are not good for realtime object detection as these models use large GPU memory due to extensive amount of parameters. Hence number of frames processed for these models are less which makes it slow. Therefore we focused on improving YOLOv5m model which can give better accuracy as well as fast training.

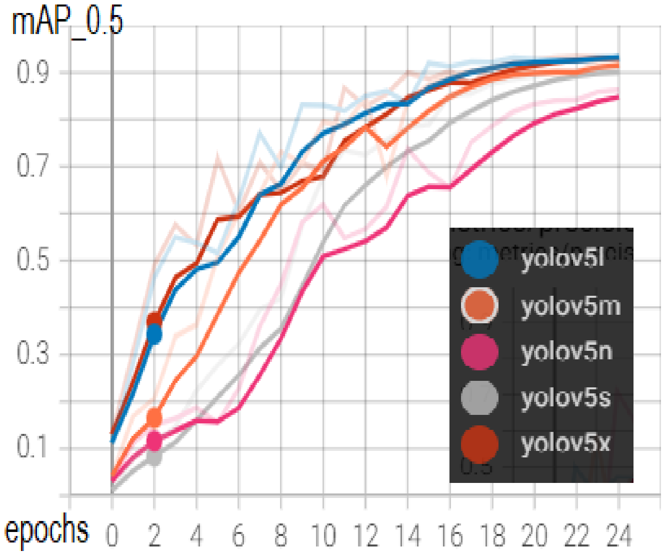


Figure 4.1: Comparison of mAP score for all YOLOv5 models

Figure 4.2 shows comparison of mAP-0.5 scores plot for default YOLOv5 model when raw images are used with no preprocessing techniques, after making gamma correction, histogram equalisation and finally proposed feature extraction technique which is a hybrid of AutoEncoder, Canny edge extraction and SIFT (Scale-invariant feature transform) method . The plots clearly depicts improvement in precision when we use different feature extraction and preprocessing techniques in our model. Finding a suitable feature extraction technique is a very challenging task as it requires through experiments and evaluation of algorithms. Our proposed feature extraction technique gives a very satisfying result which can be concluded from the graphs plotted here with respect to epochs. The curve of our proposed technique lies above all other preprocessing techniques which shows a satisfying performance. It gives a mAP scores of 93.24% when experimented on initial YOLOv5m unchanged model.

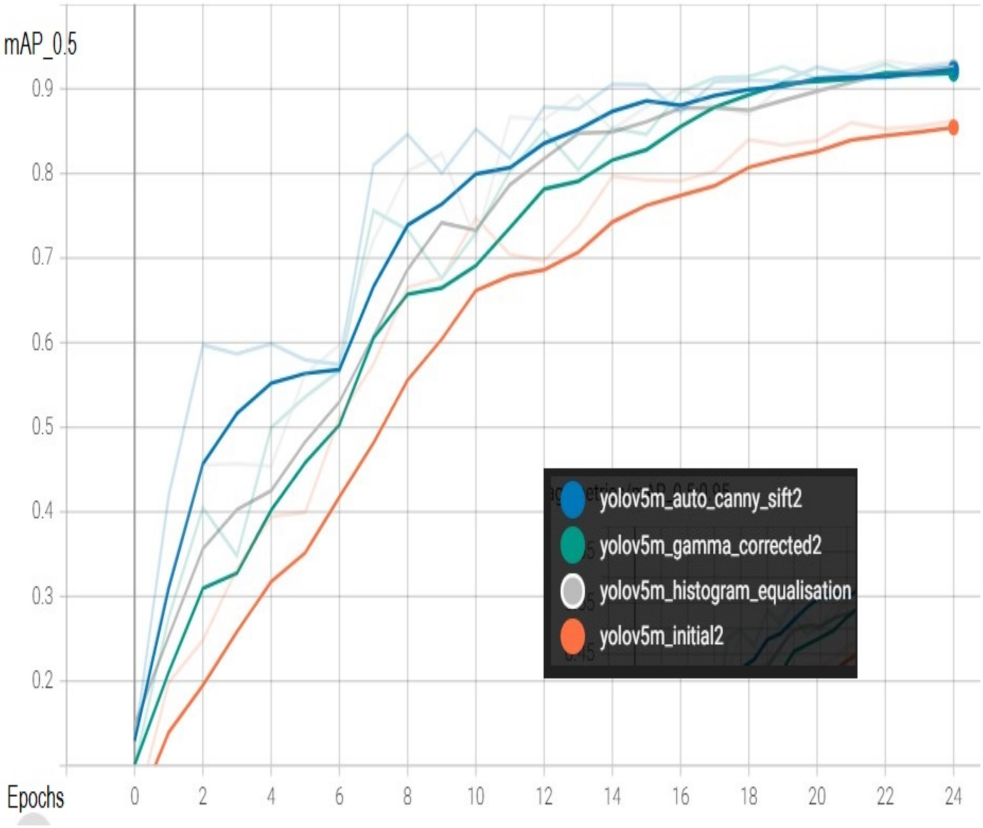


Figure 4.2: Comparison of mAP scores of proposed feature extraction and preprocessing techniques

Figure 4.3 shows comparison of performance measures (mAP-0.5) of our proposed modifed YOLOv5 model with initial unchanged YOLOv5 model. The plot clearly depicts that our proposed algorithm outperform the initial unchanged YOLOv5 model in terms of mAP-0.5 (Mean Average Precision at 0.5 IOU threshold) score. Also the modified algorithm curve lies above all other curves which shows effectiveness and efficiency of the architecture. Overall we get a mAP-0.5 score of 95.8% with proper hyperparameter tuning which is a phenomenal result. Figure 4.4 shows graph for total loss incurred for all the proposed algorithms. Total loss includes classification loss,objectness loss and bounding box loss. The plots clearly shows that our modified YOLOv5 model gives better curve of loss compared to others.It recorded a total loss of 0.015 which is quite impressive. Figure 4.7 shows some predictions for validation set images with their labels and confidence score. It clearly detects multiple images with occluded or partial views.Bounding box loss is very less as it can able to localise and detect the objects properly.

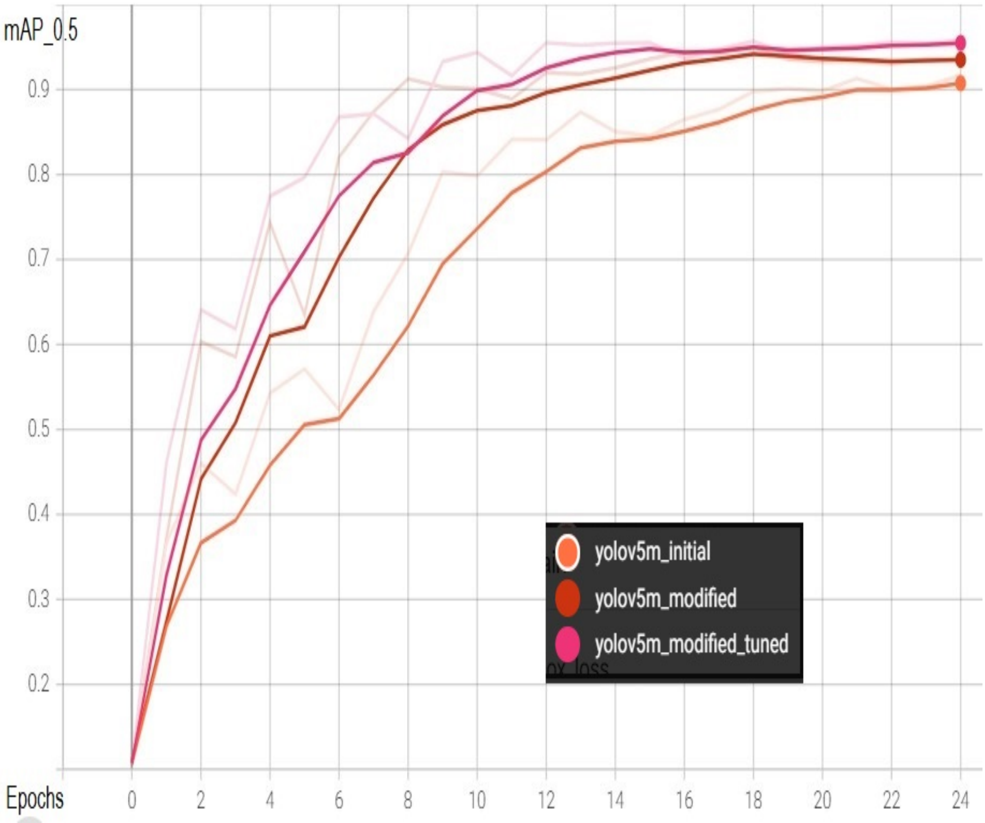


Figure 4.3: Comparison of mAP scores of modified with initial architecture

Table 4.1 depicts mAP score for all the techniques applied in this project. This clearly shows that our modified YOLOv5 model with fine tuning of hyperparameters after applying all the preprocessing techniques and feature extraction methods gives a good mAP score of 95.8% which is very satisfying.

Figure 4.5 shows PR curve of the proposed modified YOLOv5 model. It provides an overview of the trade-offs between the actual positive rate and the positive predictive value for our model by utilising various probability thresholds. Table 4.2 shows the performances of each class in terms of precision, recall and mAP. We conclude that Banana and Wrist-watch gives lower mAP scores of 0.872 and 0.911 respectively as compared to other classes while Spoon, Cricket-Bat, Cup gives excellent mAP score of 99.5% each. Hence we should train more images of Banana and Wrist-watch to improve their mAP scores.

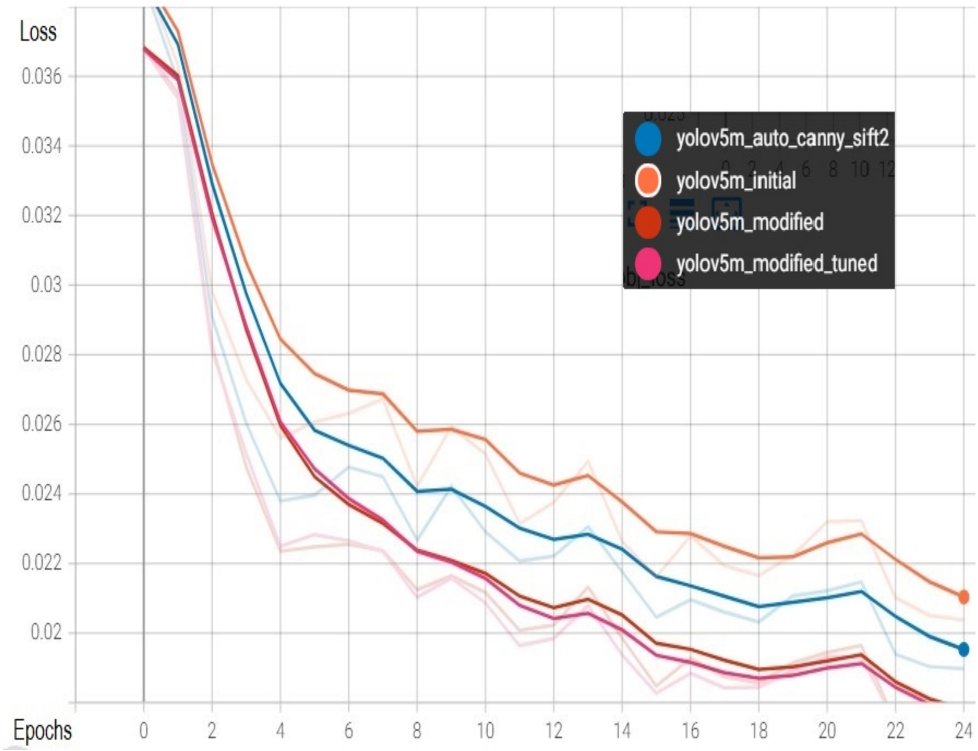


Figure 4.4: Total loss incurred in all the methods

Table 4.1: Techniques Implemented

Method	**mAP**
Without applying any technique	0.89
After applying gamma correction techniques	0.918
After applying histogram equalization	0.928
After applying modified feature extraction techniques	0.932
After modifying architecture	0.946
After modifying architecture with fine tuning of hyperparameters	0.958

Figure 4.6 represents total loss incurred when ssd-mobilenet-v2-fpnlite-320x320-coco17-tpu-8 is trained in the given custom dataset. Total loss is the sum of classification loss, localization loss and regularization loss. Total loss of 0.05 is recorded according to the experimental result.

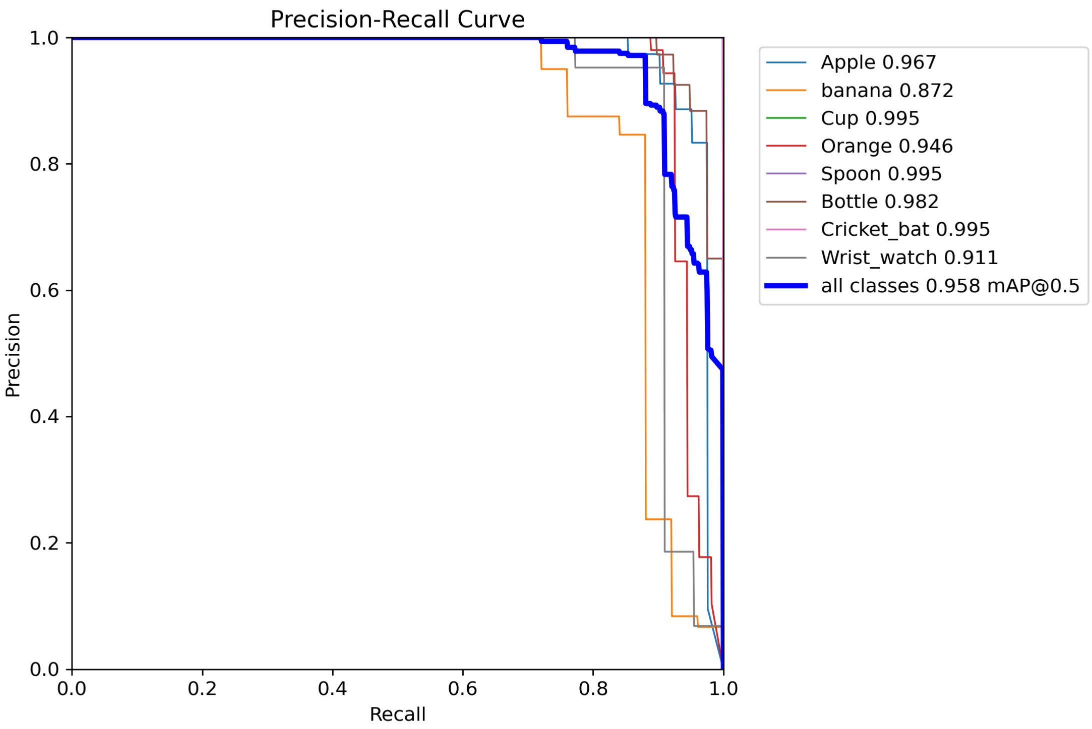


Figure 4.5: Precision Recall Curve for modified YOLOv5

Table 4.2: Performance of each class

Classes	Precision	Recall	mAP
All	0.968	0.92	0.958
Apple	1	0.842	0.967
banana	0.871	0.81	0.872
Cup	0.99	1	0.995
Orange	0.99	0.889	0.946
Spoon	0.989	1	0.995
Bottle	0.963	0.923	0.982
Cricket-Bat	0.993	0.1	0.995
Wrist-watch	0.952	0.899	0.911

Table 4.3 shows comparison in mAP and fps among various object detection models. The result clearly shows that our proposed modified YOLOv5 model gives best performance in terms of mAP as well as fps. This is because we have added an extra C3 module in the backbone which includes CBS (Conv2D, BN, SILU (Sigmoid + ReLU), 1 BottleNeck with Concat. Unlike previous default architecture , here Leaky ReLU is replaced with SILU(Sigmoid + ReLU) which learns faster and better than ReLU due to its non linearity in nature. Adding an extra layer with suitable dimensions and connecting to Neck layer in exclusive manner by replacing with lower resolution featue maps help to improve accuracy without compromising the inference speed.

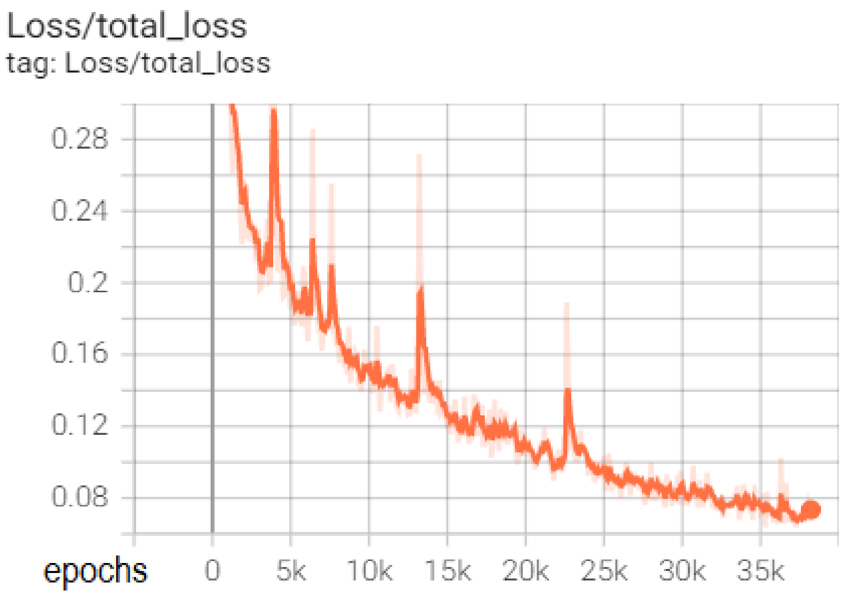


Figure 4.6: Total loss incurred in SSD mobileNetv2 model

SPP (Spatial Pyramidal Pooling) is replaced with SPPF (Spatial Pyramidal Pooling Fast) which reduces the number of FLOPS (Floating Point Operations per seconds) and improves speed of the model significantly. By replacing Neck part with biFPN retains information of small objects which is usually lost to higher layers of abstractions as we have seen in unmodified YOLOv5 model.

Anchor boxes play a crucial role in improving accuracy of a model. If best possible recall rate (bpr) is greater than 0.98 which is a assumed threshold then anchor sets are considered as the best fit else they are evolved with new anchor sets using genetic algorithm proposed in our project. Finally we get the optimized and best fit anchor sets [19,27, 44,40, 38,94] , [96,68, 86,152, 180,137] , [140,301, 303,264, 238,542] , [436,615, 739,380, 925,792] . Hence our fine tuned modified YOLOv5 architecture gives the highest mAP score as well as better fps which enhance real time object detection.

Table 4.3: Performance of different models

Model	**mAP**	**fps**
YOLOv5-intial	0.89	150
SSD Mobilenetv2	0.86	210
YOLOv5-modified-tuned	0.958	180

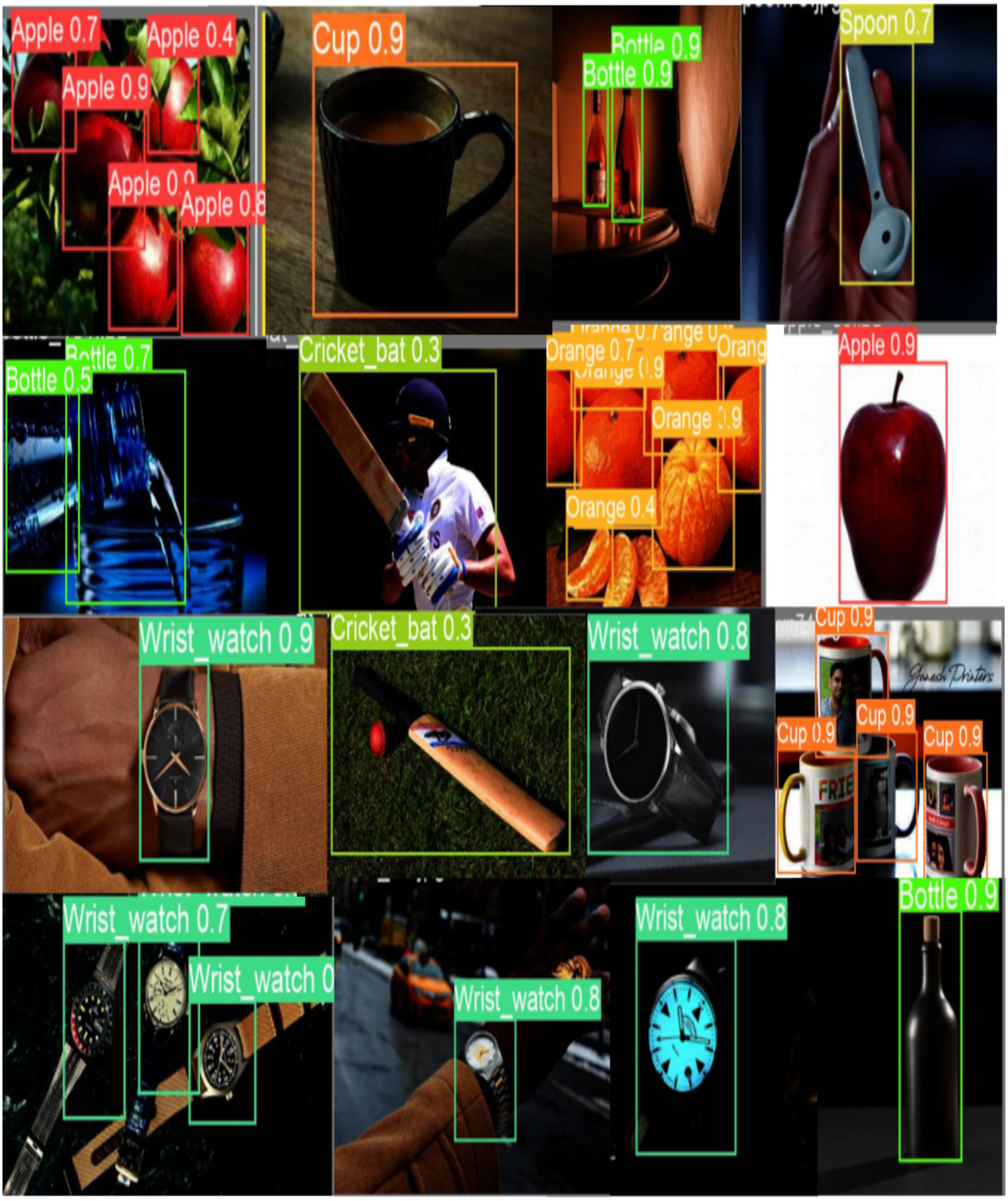


Figure 4.7: Some Experimental results of predictions

Chapter 5

Conclusion and Future Work

In this research, we have used a modified YOLOv5 model on a unique dataset made up of eight unique classes that we have collected from different sources which includes low light, tiny and occluded images. The model's accuracy is poor when data is supplied directly into it. Nevertheless, performance improves when data augmentation and image processing techniques like gamma correction, Gaussian blur filtering and histogram equalization are used. Feature extraction techniques like canny edge extraction doesn't make significant improvement in precision of the object detection but rather Our proposed technique for feature extraction of images which is a hybrid of AutoEncoder, Canny edge extraction and SIFT (Scale-invariant feature transform) methods gives a satisfying improvement in precision of the model.

Our modified YOLOv5 model gives a good result after preprocessing and feature extraction techniques. We achieved mAP of 95.8% with 0.015 total loss which is very satisfying. Selection of good fit anchor box sets are very vital in improving precision which is achieved through genetic algorithm and kmean clustering. In comparison with SSD MobileNetv2 our model gives almost same frames per second with higher precision than it. Hence our proposed model gives better performance metrics compared to other models enabling real time object detection. In future this model can be integrated in a nao bot which can enhance real time object detection and grasping of objects using inverse kinematics algorithm.

References

[1] Yang, Xingshuai and Zhao, Jingyi and Zhao, Li and Zhang, Haiyang and Li, Li and Ji, Zhanlin and Ganchev, Ivan, 2022. Detection of river floating garbage based on improved yolov5. [Online; accessed March 14, 2023].

[2] Wang, Z., Feng, J., and Zhang, Y., 2022. “Pedestrian detection in infrared image based on depth transfer learning”. *Multimedia Tools and Applications,* **81**(27), pp. 39655–39674.

[3] Chatterjee, S., Zunjani, F. H., and Nandi, G. C., 2020. “Real-time object detection and recognition on low-compute humanoid robots using deep learning”. In 2020 6th International Conference on Control, Automation and Robotics (ICCAR), IEEE, pp. 202–208.

[4] Zhao, Z.-Q., Zheng, P., Xu, S.-t., and Wu, X., 2019. “Object detection with deep learning: A review”. *IEEE transactions on neural networks and learning systems,* **30**(11), pp. 3212–3232.

[5] Cao, G., Xie, X., Yang, W., Liao, Q., Shi, G., and Wu, J., 2018. “Feature-fused ssd: Fast detection for small objects”. In Ninth International Conference on Graphic and Image Processing (ICGIP 2017), Vol. 10615, SPIE, pp. 381–388.

[6] Zou, Z., Chen, K., Shi, Z., Guo, Y., and Ye, J., 2023. “Object detection in 20 years: A survey”. *Proceedings of the IEEE.*

[7] Ren, S., He, K., Girshick, R., and Sun, J., 2016. “Faster r-cnn: Towards real-time object detection with region proposal networks”. *Advances in neural information processing systems,* **28**.

[8] Singh, B., Najibi, M., Sharma, A., and Davis, L. S., 2021. “Scale normalized image pyramids with autofocus for object detection”. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **44**(7), pp. 3749–3766.

[9] Singh, B., Najibi, M., and Davis, L. S., 2018. “Sniper: Efficient multi-scale training”. *Advances in neural information processing systems,* **31**.

[10] Yang, G., Feng, W., Jin, J., Lei, Q., Li, X., Gui, G., and Wang, W., 2020. “Face mask recognition system with yolov5 based on image recognition”. In 2020 IEEE 6th International Conference on Computer and Communications (ICCC), IEEE, pp. 1398–1404.

[11] Nguyen, N.-D., Do, T., Ngo, T. D., and Le, D.-D., 2020. “An evaluation of deep learning methods for small object detection”. *Journal of electrical and computer engineering,* **2020**, pp. 1–18.

[12] Zhou, F., Zhao, H., and Nie, Z., 2021. “Safety helmet detection based on yolov5”. In 2021 IEEE International conference on power electronics, computer applications (ICPECA), IEEE, pp. 6–11.

[13] Huang, T., Cheng, M., Yang, Y., Lv, X., and Xu, J., 2022. “Tiny object detection based on yolov5”. In 2022 the 5th International Conference on Image and Graphics Processing (ICIGP), pp. 45–50.

[14] Teng, X., Fei, Y., He, K., and Lu, L., 2022. “The object detection of underwater garbage with an improved yolov5 algorithm”. In Proceedings of the 2022 International Conference on Pattern Recognition and Intelligent Systems, pp. 55–60.

[15] Benjumea, A., Teeti, I., Cuzzolin, F., and Bradley, A., 2021. “Yolo-z: Improving small object detection in yolov5 for autonomous vehicles”. *arXiv preprint arXiv:2112.11798*.

[16] Jung, H.-K., and Choi, G.-S., 2022. “Improved yolov5: Efficient object detection using drone images under various conditions”. *Applied Sciences,* **12**(14), p. 7255.

[17] Kim, J.-H., Kim, N., Park, Y. W., and Won, C. S., 2022. “Object detection and classification based on yolo-v5 with improved maritime dataset”. *Journal of Marine Science and Engineering,* **10**(3), p. 377.

[18] CENGİL, E., and ÇINAR, A., 2021. “Poisonous mushroom detection using yolov5”. *Turkish Journal of Science and Technology,* **16**(1), pp. 119–127.

[19] Wu, T.-H., Wang, T.-W., and Liu, Y.-Q., 2021. “Real-time vehicle and distance detection based on improved yolo v5 network”. In 2021 3rd World Symposium on Artificial Intelligence (WSAI), IEEE, pp. 24–28.

[20] Ting, L., Baijun, Z., Yongsheng, Z., and Shun, Y., 2021. “Ship detection algorithm based on improved yolo v5”. In 2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE), IEEE, pp. 483–487.

[21] Yan, B., Fan, P., Lei, X., Liu, Z., and Yang, F., 2021. “A real-time apple targets detection method for picking robot based on improved yolov5”. *Remote Sensing,* **13**(9), p. 1619.

[22] Shorten, C., and Khoshgoftaar, T. M., 2019. “A survey on image data augmentation for deep learning”. *Journal of big data,* **6**(1), pp. 1–48.

[23] Cheng, H.-D., and Shi, X., 2004. “A simple and effective histogram equalization approach to image enhancement”. *Digital signal processing,* **14**(2), pp. 158–170.

[24] Amiri, S. A., and Hassanpour, H., 2012. “A preprocessing approach for image analysis using gamma correction”. *International Journal of Computer Applications,* **38**(12), pp. 38–46.

[25] Sekehravani, E. A., Babulak, E., and Masoodi, M., 2020. “Implementing canny edge detection algorithm for noisy image”. *Bulletin of Electrical Engineering and Informatics,* **9**(4), pp. 1404–1410.