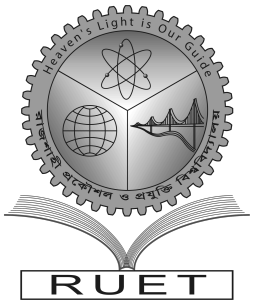
Heaven's Light is Our Guide



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

Face Detection for Driving Assistance using YOLO V-8

Author

Muzahidul Hassan

Roll No. 1703038

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

Supervised by

Professor Dr. Md. Shahid Uz Zaman

Professor

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

ACKNOWLEDGEMENT

First of all, i would like to thank the Almighty Allah for giving me the opportunity , strength and courage along the way for completion of this thesis work. I would also like to express my sincere appreciation, gratitude and respect to my supervisor Md.Shahid Uz Zaman, Professor of Department of Computer Science and Engineering, Rajshahi University of Engineering and Technology, Rajshahi. Throughout the entire semesters, he has not only given us technical guidelines, advise and necessary documents, but also has given us continuous encouragements, helps, sympathetic co-operation and guidance whenever he thought necessary. His continuous support, help and guideline were the most important factors that helped me to achieve this result. Without his constant supervision, this work would have not been possible to carry all the way out.

I am also grateful to all my respective teachers of Computer Science and Engineering department for all the good and valuable suggestions and inspirations during this whole time period.

Finally, i would like to convey my heartiest thanks to my parents, friends and well wishers for their constant inspirations and support throughout this work.

August 28, 2023 Muzahidul Hassan

RUET, Rajshahi

Heaven's Light is Our Guide

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

CERTIFICATE

This is to certify that this thesis report entitled ***"Face Detection for Driving Assistance using YOLO V-8 "*** *submitted by* ***Muzahidul Hassan, Roll:1703038*** *in partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Department of Computer Science & Engineering of Rajshahi University of Engineering & Technology, Bangladesh is a record of the candidates' own work carried out by them under my supervision. This thesis has not been submitted for the award of any other degree.*

Supervisor

_____

Dr. Md. Shahid Uz Zaman
Professor
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204

External Examiner

_____

Dr. Md. Rabiul Islam
Professor
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204

ABSTRACT

Computer Vision has provided significant mechanisms to bring the best out of this automation technology. And in the field of automation, image detection and recognition is one of the most basic building blocks. Therefore, in this thesis paper, we have discussed and analyzed the field of image detection systems that would be used in transports in order to give assistance to the driver to avoid chaotic events that most often occurs on the roads and highways in our country. In case of giving assistance with image detection and recognition systems, we will implement and compare two most popular and well known methods: Viola-Jones and YOLO model. At first we will try to detect the motion of a driver's face to detect drowsiness of a driver while driving on the roads. That will help to alert the driver to take necessary precautions and safety. In recent years Many algorithms for detection have appeared, which depend on extracting the features of the human face, and works continue to develop them to this day. This paper aims to make a comparison between two of the most commonly face detection methods, Viola Jones (Viola-Jones) and YOLO v8. This comparison is made to determine which of the two algorithms is being most useful when used to detect faces in digital video. These algorithms are used in many applications, including image classification, medical analysis of image, and objects detection in real time (especially in surveillance cameras). The experimental results of a sample consists of 20 video frames show that Viola-Jones algorithm consumes less time in comparison with YOLO v8 algorithm, but its results are less accurate, unlike the YOLO v8 algorithm, which is slower in detect face with high accurate rate.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

In this modern world, we can not live a single moment without the help of science. As science has developed gradually at it's pace, it has also made our expectations sky-high. In this thesis, we have tried out and carried out a new way of using the power of science in our object detection process.

In the field of object detection process, whether it's from an image or video, there are a lot of tools and ways to gain the actual purpose. YOLO models are one of those specific models that can be used in this kind of work field.[1]

1.1 Problem Statement

The problem that we need to solve in this thesis paper is to have a better algorithm to detect objects that will help and assist a driver to have a safe journey and without facing any difficulties.

Deep learning is common word in recent days of modern science. In this era of machine learning, a huge data set is required to gather info. Many organizations are using deep learning as CNN, which stands for Convolutional Neural Network and use it to detect object in a video sequence. Face detection is both popular and difficult in case of pattern recognition[2]. Various face recognition patterns such as face verification or clusters of faces are some of those cases.

Effective training needs to be carried out to fulfil our objectives which is object detection and recognition. The accuracy in face detection algorithm is not yielding to a good and desired output. That is why this paper mainly works with the latest model of YOLO which will help us to gain the desired accuracy and output.

1.2 Motivation

The method of object detection is a case of pattern recognition and verification. The main motive of this pattern recognition is to detect the actual object within an image or a real time video.

For this case, YOLO is a popular deep learning library which will be implemented for recognizing and verifying patterns such as faces or cars or vehicles. This proposed methods uses CNN for detecting objects in a real time video [2].

In early days, research in this field was carried out by hand crafted extraction methods which was time consuming and less effective. But now, with the help of algorithms and models such as YOLO-V8, we can detect any object by limited time and dataset.

The model of YOLO-V8 is being compared to other models such as YOLO-V3 and Viola-Jones. This comparison will be based on accuracy, performance and other attributes that can help to detect the best model for that specific purpose.

1.2.1 Example Figure

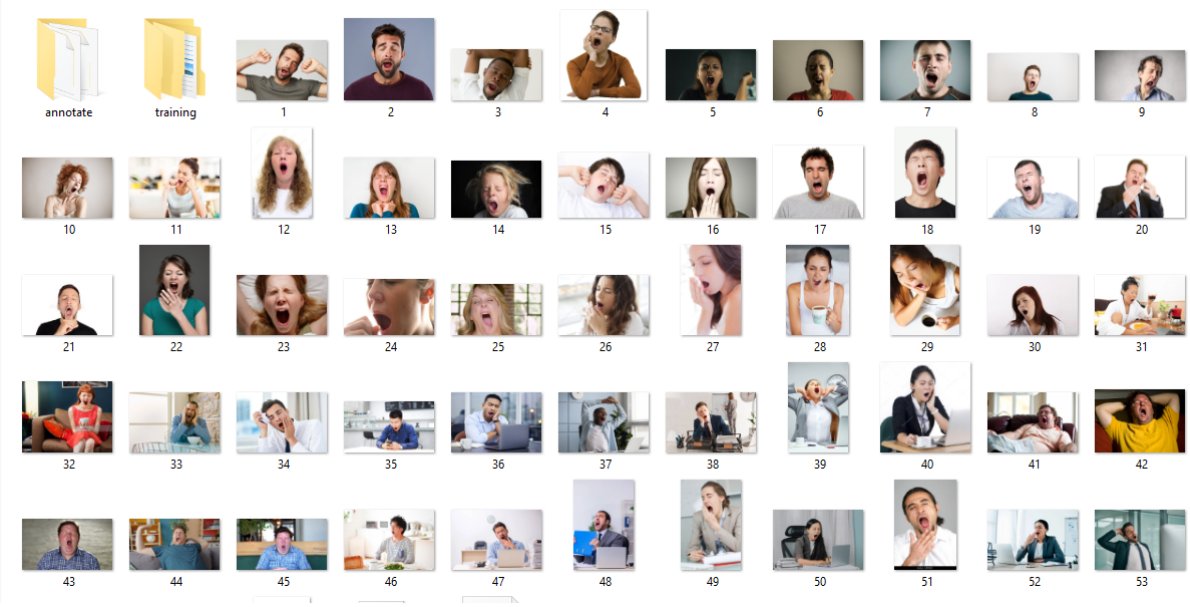


Figure 1.1: Collection of Images

1.2.2 Example Referencing

The examples that has been used is taken from the internet and an open source website that provides different categories of images with annotations. As we are building from scratch, we gathered most of our images from the internet and annotated them manually one by one. We used different tools for this annotation process that will be mentioned soon.

1.3 Thesis Contribution

Driver fatigue is a major issue when it comes to safe driving on the roads. But then again, a person might not be aware of his physical situation and which may lead to unwanted events. By recognizing objects that beholds in front of a vehicle, one can prevent such an event.

Computer vision approaches are very must interested and keen in this area. It has tons of possibility to reduce this kind of event[3].In case of detecting drowsiness of a driver, a method of computer science can be of much help and beneficial too. This benefit will be depend on what model would give the better output. The model that will give the better output, will be selected as the main model for this purpose.

For that particular reason, this thesis will work on detecting drowsiness of a driver while driving a vehicle on the road. This kind of system will alert the driver about his physical situation and will send warnings to the driver.

Driving is a process that involves awareness of the situation, accurate decision making and proper guidance. It needs extra care and evaluation in terms of making decisions. So we need to take appropriate measures to ensure our result is moving towards our desired goal.

1.4 Thesis Organization

In this thesis, we will work on detecting facial expressions, mouth and eyes positioning, pedestrian detection (person/people) and vehicle detection to avoid collusion. We will mainly work on using YOLO-V8 and other algorithms or models such as Viola Jones image detection and others previous versions of YOLO models.

A nonintrusive method will be approached to detect an object in an image or a real time video. Given the quantized fatigue level, the method will alert the driver and the driver will be aware of his physical situation.

Transfer learning is a method that can be used in this project, but above all, deep learning is the fundamental basis of this thesis. We need to have strong command over CNN and deep neural networks to persue the highest accuracy among other existing models and algorithms.[4]

1.5 Chapter Summary

In this chapter, we mainly discussed about the introductory theme and basic things like YOLO-V8 models and Viola-Jones model to detect object in an image such as facial expressions. We use these models to compare and analysis the results and judging accuracy with respect to time and other factors.

Chapter 2

Background Study and Literature Review

2.1 Introduction

Now a days, facial expression technology has gain more attention than any other aspects of this area. A lot of experiments have also been conduct over the past decade. There are many existing methods are used for feature extraction and image classification. This chapter gives a short preview of those existing models.

2.2 Theoretical Background

Vision based fatigue is a common case in this field of experiments. A driver can feel sleepy during the driving of a vehicle. The proposed system specifies the tools that are necessary to involve with different methods to get real time video extraction.[5]In order to detect the person who is driving is yawning or not, we need to implement those methods to detect the actual positions.

The main theme is to gain proper knowledge and gather proper and suitable extraction.We know how the things work in this case with actual help of feature extraction and multiple records. The distance classifier needs to select accurate distance between the mouth opening and eye pupil distance.

2.2.1 Theories and mechanisms

Different theories have been introduced to detect and classify objects from images and real time videos. In this paper, we mainly focus on the face detection using two different algorithms.

Figure 2.1: Fatigue and Tiredness of a Driver

A driver feels drowsy when he is very much tired by driving a transport all day long. The level of tiredness depends upon the time period that he/she has been driving for. So, we can say that the drowsy feeling is almost proportional to that time period.

2.2.2 Previously used version and method

Many models of YOLO were used in previous papers. There are different kinds of models that YOLO architecture has bestowed upon our actual needs. In this paper, we will be only discussing about the specific version of a YOLO model which is: YOLO-V3.

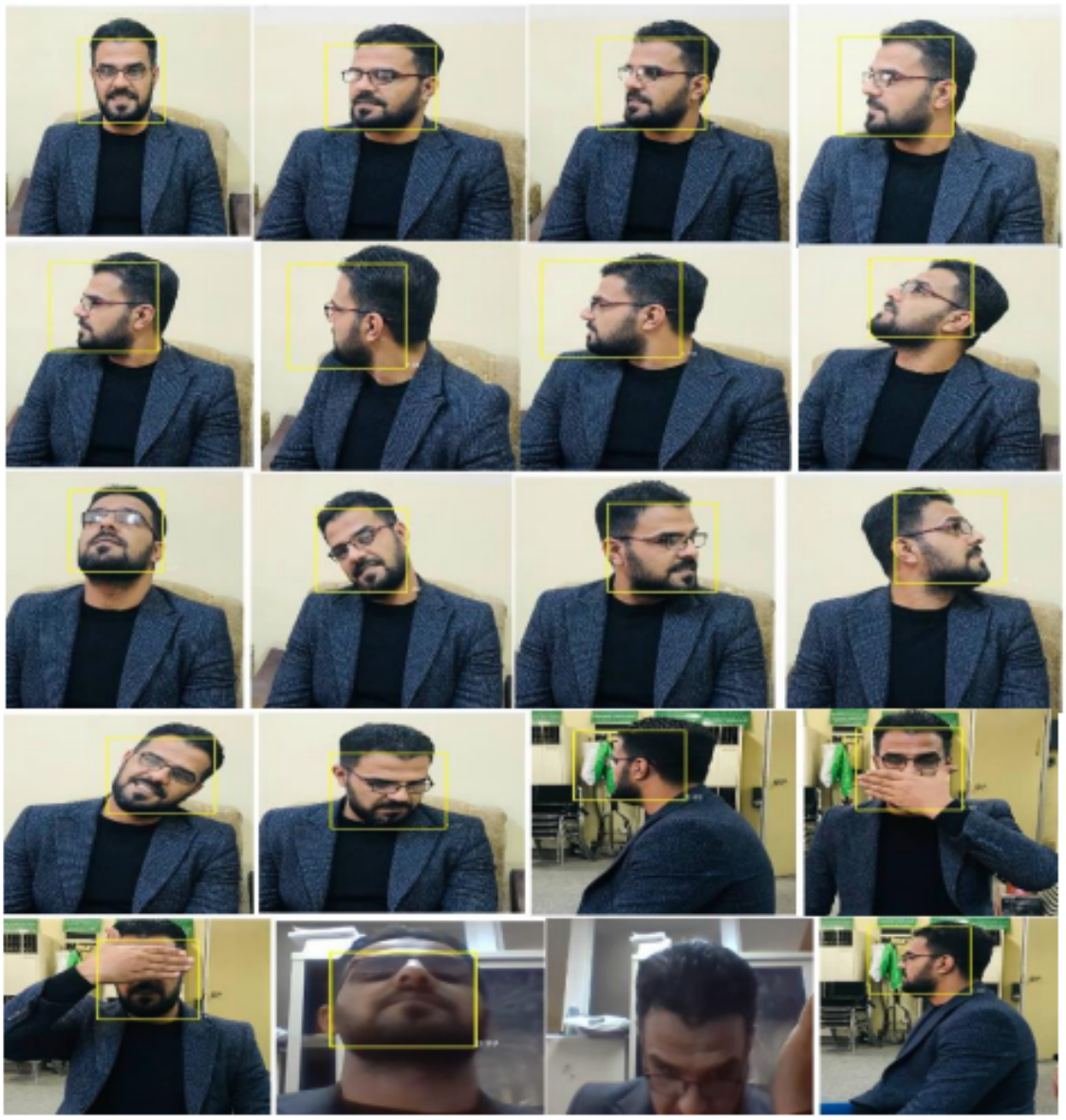


Figure 2.2: Using the model YOLO-V3

In YOLO-V3 model, face detection is being processed using image grids that is used to help and detect from various angles. If an image contains a face in its background, it was able to detect and extract that face from that specific background from any angle. No matter what the angle is the face from the camera, it can detect the actual object smoothly and without any delay.

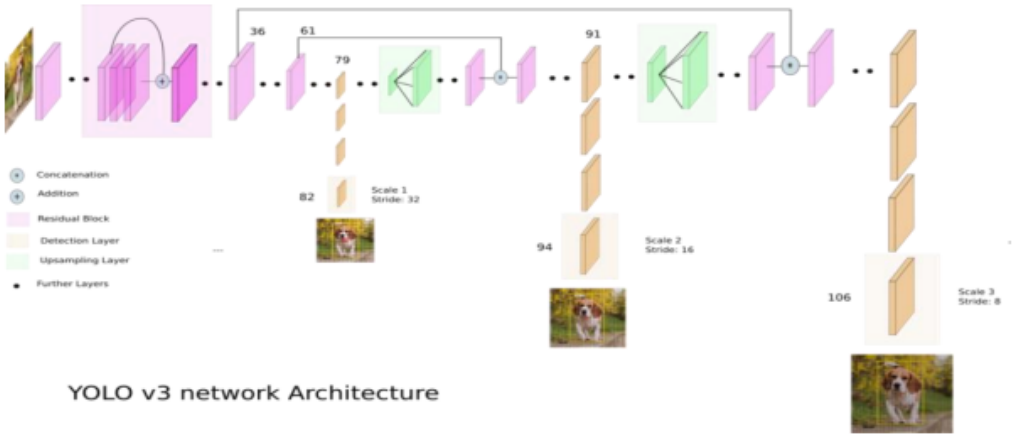


Figure 2.3: YOLO-V3 architecture

2.3 Related Works

There are many researches that proposed many linear and non linear approaches. Those are sub-space based facial recognition techniques for pattern detection. The principal method analysis (PCA) is one of those methods. The Kernel PCA is a nonlinear statistical procedure that uses a dataset to run and test and analyze the data. This performance analysis is highly dependent on the training images that would be used to detect facial expression, or people in general or vehicles.

PCA stands for Principal Component Analysis, and it's a dimensionality reduction technique used in various fields, including statistics, machine learning, and signal processing. PCA aims to transform high-dimensional data into a lower-dimensional representation while preserving as much of the original data's variability as possible. It achieves this by identifying the principal components, which are orthogonal axes along which the data varies the most.

The experimental results have been carried out by those popular methods naming KPCA and other feature extracting algorithms. This is compared to the linear PCA technique which is statistical also. They used euclidean distance classifier to extract the feature and create feature vectors.

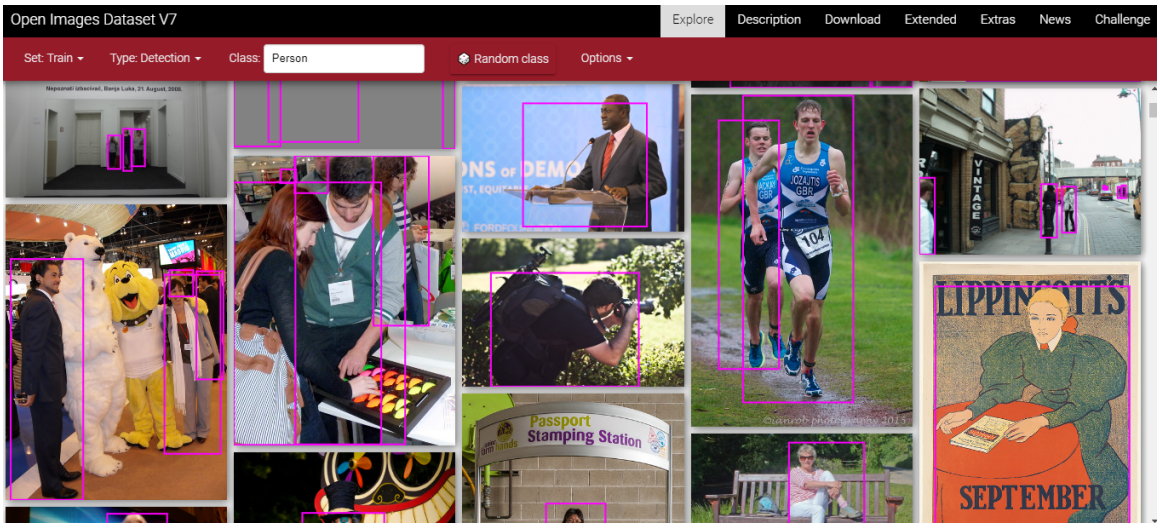


Figure 2.4: Detection and Annotation: Part-1

The above experiment was faulty however. That research produced outliers and biased results. For that reason, we had to deviate the result from the previous experiments and notes them down to gain more accurate and unbiased and reliable result.[6]

While in case for driving a vehicle, a driver must pay proper attention while driving on roads and highways. To detect yawning, there are some particular steps to follow and recognize such as mouth opening or eye positioning. The region around the mouth and the eyes can represent the state of drowsiness and fatigue.

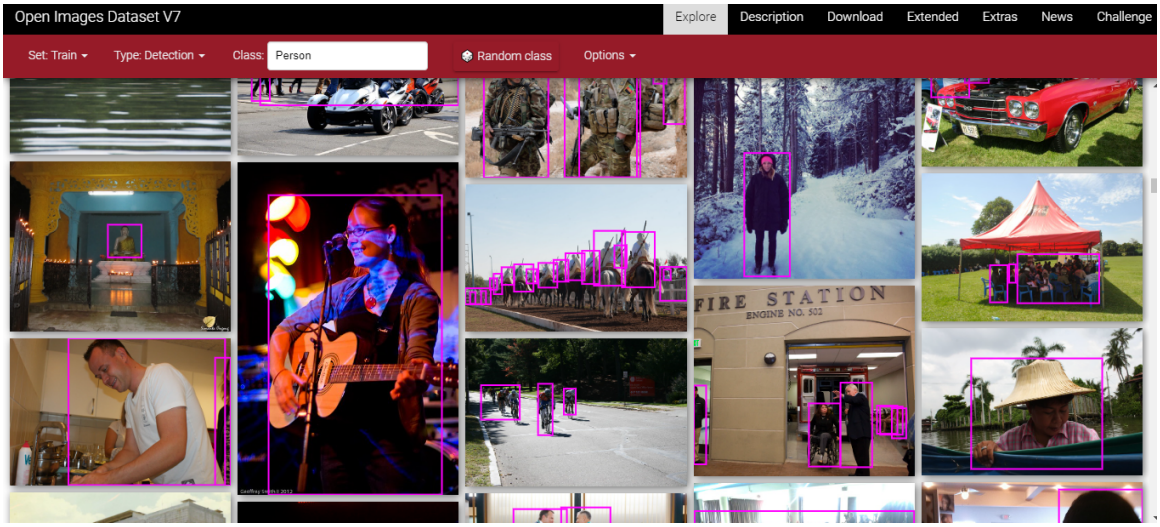


Figure 2.5: Detection and Annotation: Part-2

Now the real challenge is to extract that feature from a real time video[3]. That is very much challenging because eye closure data can only be extract using the distance of the pupil of the eyes. And the state of the eye is truly requires high computational time and the procedure to initialization.

2.4 Chapter Summary

The chapter mainly focuses on the experimental procedures that are used in image extraction and feature detection. Research is being carried out using all these discussed algorithms. We need to find out an efficient combination for classification and feature extraction to carry out this research.

Chapter 3

Methodology

3.1 Introduction

The main goal is to have a system that can properly detect objects without being time consuming. The detection system can be of two general features. One is about the physical appearance of the driver and another is the vehicle[7]. Both the feature extraction techniques can be used more the detection of objects. After the detection of drowsiness, it will alert the driver to take appropriate measures to prevent an unwanted event.

The objective of this detection is based on the yawning of the driver. There are three methods that will help to detect the drowsiness of a person. Firstly, we can use color segmentation of the drivers face. In this method, we have to clear the background area of the face to get the actual desired image.[8]

Then the eye location would be used as a reference point to detect the location of the mouth. As in drowsiness, yawning is the key factor here. And to detect the yawning of a person, mouth and face positioning is a must be factor.

Then the eye location would be used as a reference point to detect the location of the mouth. As in drowsiness, yawning is the key factor here. And to detect the yawning of a person, mouth and face positioning is a must be factor.[9]

Only useful features are need to be selected for generating a desired output. In case of object detection in the face, the background area must have to be removed and we can not take the whole attribute of face in this case. Only mouth and eye positions are mandatory in this detection method.

Without detecting face, the further experiment is not possible. So we to need carefully extract the features from the face. Generally, the mouth component relies on the white area of the face and located in the lower half of the face and also in the center position from the two eyes. These information can be effective in detecting the mouth portion of the face.[10]

3.2 Viola-Jones method

Viola Jones is a real time face detecting algorithm that is used to extract and analyse feature from a given input. The yawning detection can be done with the help of active contour model. By applying skin color segmentation procedure, we have to extract the face from the background of the real time video.

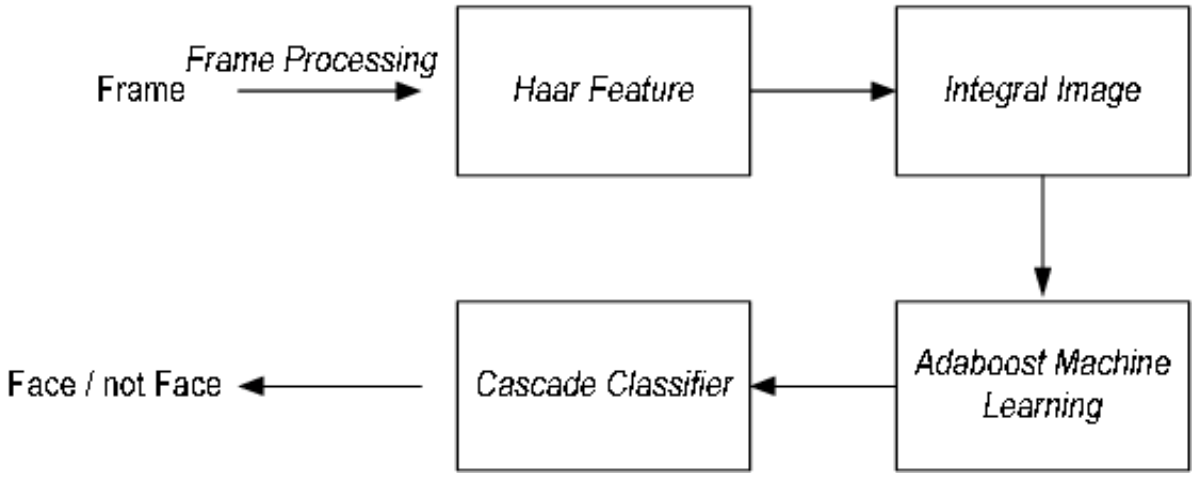


Figure 3.1: Schematic of Viola-Jones method

The correct candidate of the face is detected after many more images that are taken as input. In case of yawning the mouth becomes wider and wider[11].The more the yawning, the wider the mouth. The viola jones method is very robust in comparison to other methods.

3.2.1 Haar feature selection

Haar feature selection uses convolution kernels to extract and detect features from a given input image. We use a black region that defined +1 and the white region can be defied as -1. When this method is applied, the pixel values of the image is divided or subtracted by the above features.

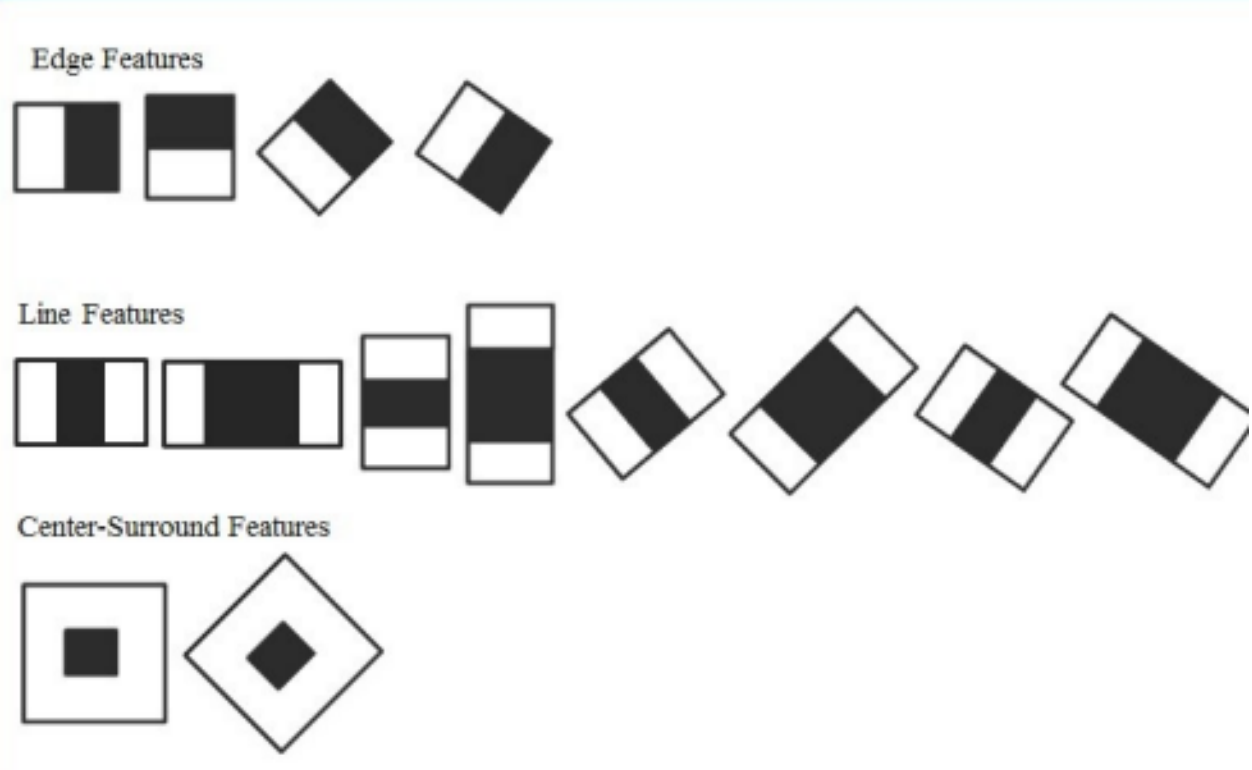


Figure 3.2: Haar Feature

3.2.2 Adaboost training algorithm

In Adaboost training, the human face is detected by some specific method such as it can detect the best feature among 16000+ more features. In mathematical terms, it weights the feature vectors by beta and applies weights whenever the value is optimized.[12]

It can extract only the relatable features and only that is relevant according to our usage. It performs better than randomly guessing a feature.If the output is 0, then the classifier is weak, and if the output is 1, then the classifier is strong enough to give us accurate result.

3.2.3 Cascaded classifier

A cascaded classifier is a combination of multiple stages or steps that helps to determine and create a strong classifier[13].The features are grouped into several stages and they are used by pipeline method from one step to another and by following each step, it generates a feature that is stronger than the previous output.

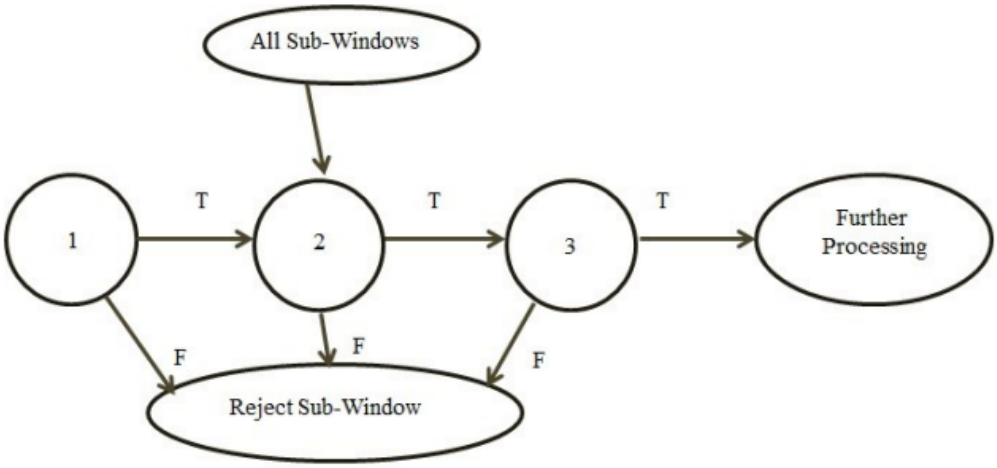


Figure 3.3: Cascading Classifier

Adaboost takes linear combination of a large amount of features and it checks the threshold everytime. If the threshold is over the limit, it stops generating features and if its not, then it continues to generate features.

3.3 Dimensionality Reduction

Dimensionality is the number of feature vectors in a feature space. And when it comes to dimensionality reduction, it means reducing the features of a component in a feature space. When the feature is very large , the extracting procedure becomes more and more complex and it will cause overfitting which is not good for image detection. High-dimensional data can lead to overfitting and poor generalization. Dimensionality reduction helps mitigate these issues by reducing noise and focusing on the most informative features.

If a machine learning algorithm uses a lot of features and it will eventually cause poor performance in a dataset, the testing data will also be overfitted[13]. So by doing dimentionality reduction, we can avoid the problem of overfitting , that is why it is one of the most popular algorithm in recent days.

3.3.1 Advantage of dimensionality reduction

There are some specific advantages of dimensionality reduction than other models out there. It can remove misleading data from a model and thus improving accuracy of that model. It can speed up the task and helps to unload the memory quickly when the task is done. As it uses less dimensions, it prevents the event of overfitting that is caused by a large dimensional dataset. It also helps to remove redundant dataset and noise pollution.

It can be divided into two categories, feature selection and feature extraction. The main difference of both of those categories are, feature extraction helps to create brand new feature and one the other hand , feature selection only selects a less number o feature among them. So, basically feature selection has only a subset of features than feature creation.[14]

In order to find the value of the first feature, the following equation will be applied based on using the integral image:

$$decision - value = sum_0 * weight_0 + sum_1 * weight_1 \tag{3.1}$$

High-dimensional datasets can lead to various challenges, including increased computational complexity, overfitting, and difficulty in visualization. Dimensionality reduction techniques address these challenges by transforming the data into a lower-dimensional space that captures the most important patterns and relationships within the original data.

3.3.2 Principal Component Analysis

PCA is also a dimension reduction method that is used to reduce the dimension or feature of a a large dataset and reduce it into a smaller subset of those dataset that contains most of the important and useful data from the large dataset[15]. It is a technique that helps to find only the most useful and correct feature among the whole lot of dataset of high dimensions. In this method, we compress the dimension of the data features by reducing the feature vectors in numbers without even loss of information.

The main feature vectors that are extracted from the original dataset is called principal components. The large dataset that was given in the first place had a large amount of feature vectors, but after projecting those features in a much smaller space, all we have is the main features that will help to extract our main object in a real time video.

3.3.3 Linear Discriminant Analysis

Considering the label of classes, Linear Discriminant Analysis helps to analyse the dataset by reducing the dimension of a huge amount of data. The main difference of this LDA with PCA is, in case of PCA, it maximizes the variance of the dataset where LDA only works with separating the features among various classes[16].

It maximizes the distance between two existing class labels and on the other hand it minimizes the variance of each class. Thus it reduces the vector space and in another word, the dimensions of a large dataset.

3.4 Chapter Summary

In this chapter, we have discussed about the basic features and approaches of different models. We can select a specific model or a combination of models to get high accurate output. We can also find out the best model among them by testing the dataset within those models.

Chapter 4

Implementation

4.1 Introduction

In this chapter, we focus on the implementation of the YOLO-V8 algorithm with compare to Viola-Jones algorithm.

We all know that, to implement an algorithm or a specific model, we have to carry out some steps to implement those models. Those steps should be executed sequentially and consciously. Some of the steps to be driven in this thesis are given below:

1. Convolutional Neural Network(CNN)
2. Open CV
3. Yolo-V3
4. Face Detection

The methods mentioned above are the main theme of this project. I this thesis paper, we will mainly discuss about these methods and compare them with each other and find the best output.

4.1.1 Convolutional Neural Network(CNN)

Convolutional neural network has been contributing a long way in case of object classification. It is one kind of Artificial Neural Network that is mostly used in visual data analyzing and classification such as real time video or images. It consists of convolution layers, activation functions, fully connected layers and output layer.

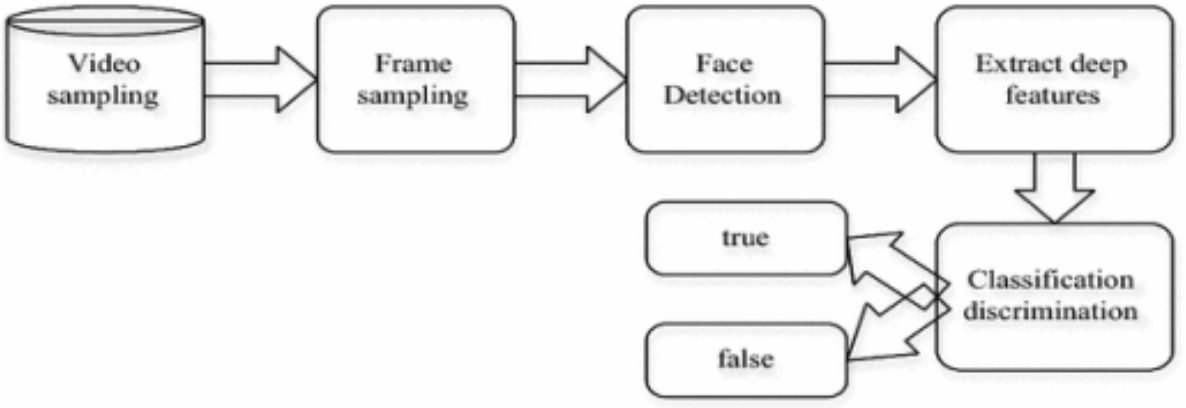


Figure 4.1: CNN Based Classification

4.1.2 Open CV

OpenCV stands for ”Open Source Computer Vision Library,” is an open-source computer vision and machine learning software library. It provides a wide range of tools and functions for image processing, analyzing, and understanding visual data, such as images and videos.

OpenCV can detect and analyze contours (boundaries of objects) in images, allowing for shape analysis and object recognition. In this paper, we will be detecting mouth contour by the systematic approaches that are available to us.

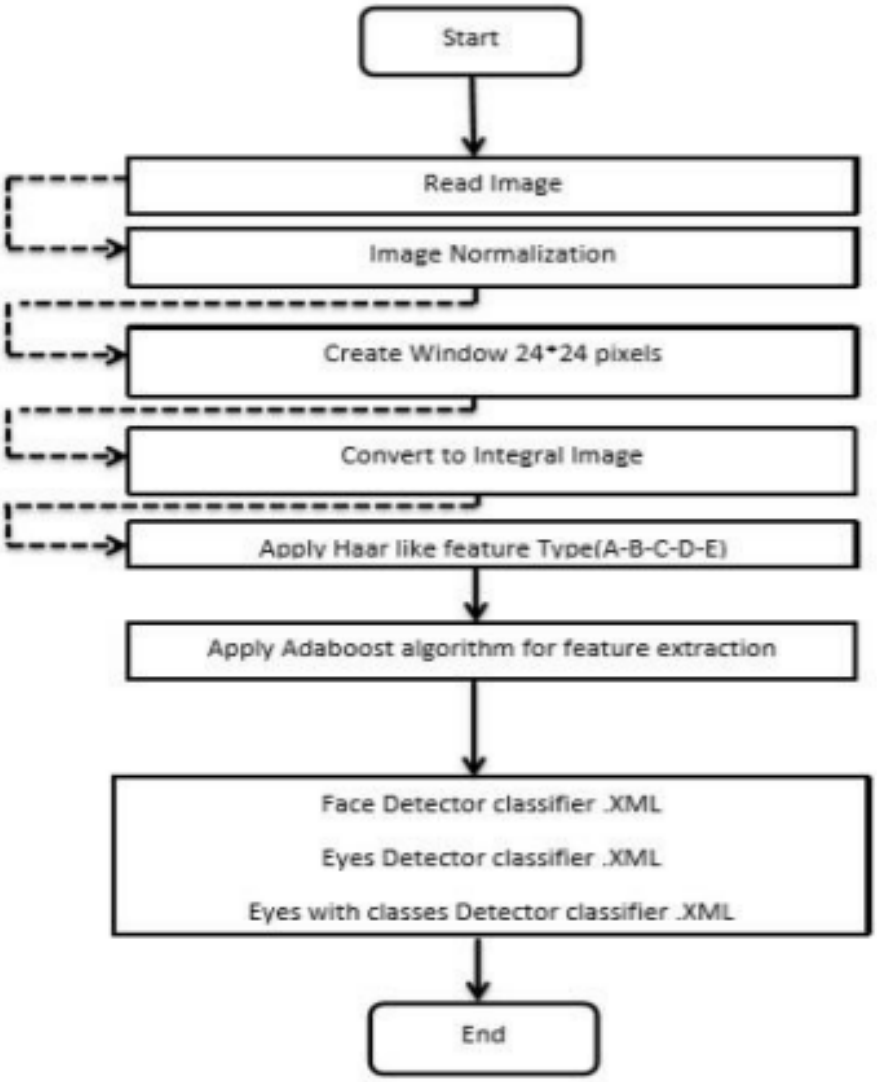


Figure 4.2: OpenCV Detection

The key features that it includes are: Image Filtering and Transformation, Object Detection and Tracking, Camera Calibration, Graphical User Interface (GUI) Support, Geometric Transformations, Contour and Shape Analysis and so on.

Here, we can also see that, the image dataset we will be using will go through transformations like image normalization and then the images will create a new window of 24x24 pixels. Then it will be converted to integral image. After applying Haar like feature and adaboost algorithm, we will get classification of eyes, mouth and so on.

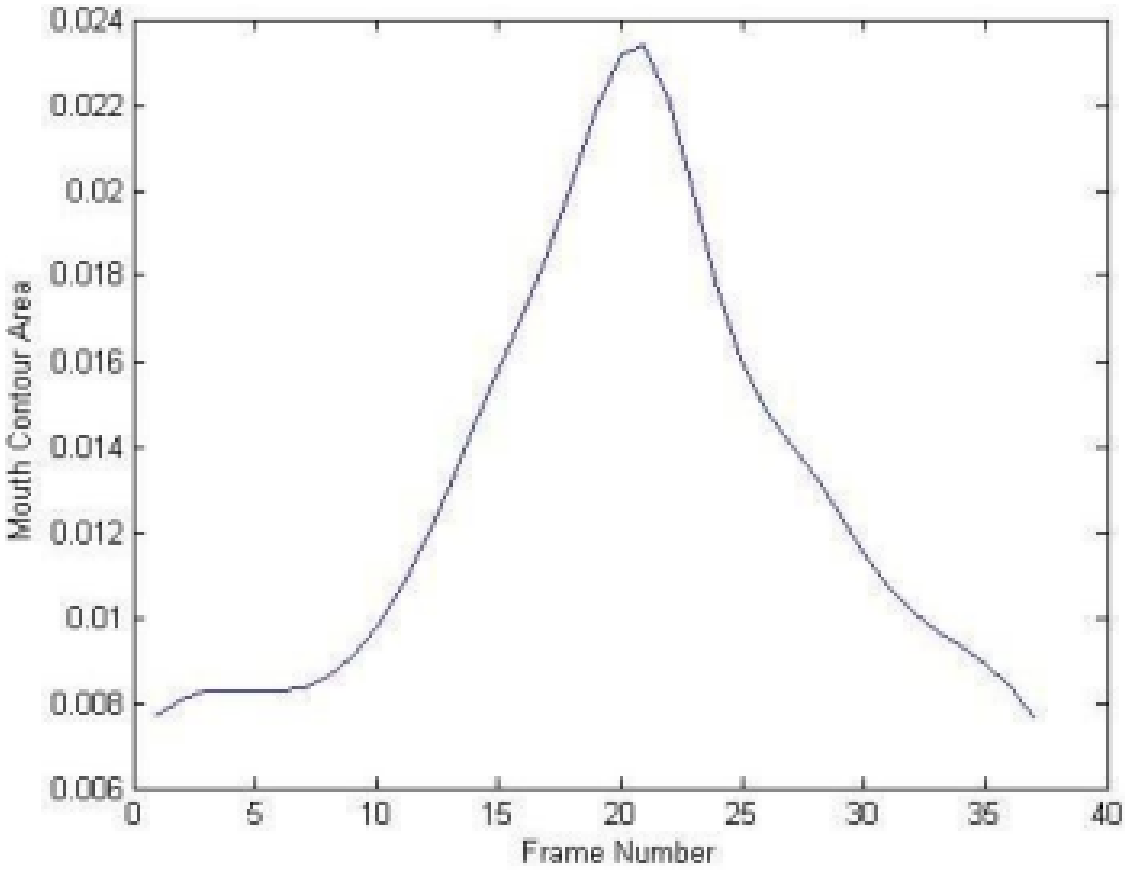


Figure 4.3: FPS in Mouth Contouring Detection

4.1.3 Yolo-V8

Yolo-V8 is the latest model and version of YOLO architecture. The full meaning of it is ”You-Only-Look-Once”.It’s designed to detect and locate objects within images or video frames while classifying the detected objects into predefined categories.

YOLO v8 is an improvement over its predecessor, YOLO v7, and offers significant enhancements in terms of accuracy and speed. The architecture of YOLO-V8 is given on the next page:

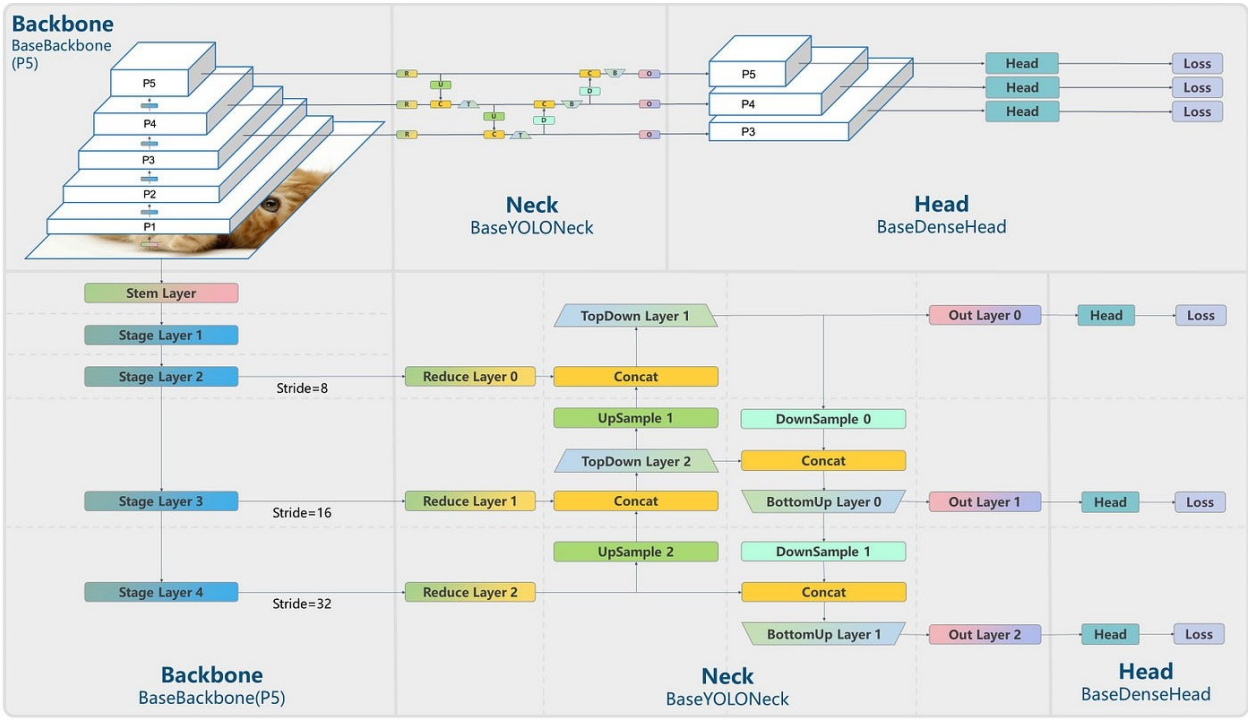


Figure 4.4: YOLO-V8 Architecture

4.1.4 Face Detection

For detecting objects, we can use plenty of methods and algorithms. But in this paper, we will discuss the best algorithm that will be used for detecting faces from an image and its backgrounds. We will consider the accuracy, efficiency, robustness of those algorithms and compare them with each other to find out the best face detecting algorithm among them.

Face detection algorithms are techniques used in computer vision to automatically identify and locate human faces within images or video frames. These algorithms play a crucial role in a wide range of applications, including facial recognition, emotion analysis, identity verification, security systems, and more.

These algorithms use various techniques, including machine learning, feature engineering, and neural networks, to detect faces in different ways. The choice of algorithm depends on factors like accuracy requirements, computational resources, and the specific application you're working on. Additionally, advancements in AI continue to contribute to the development of more accurate and efficient face detection techniques.

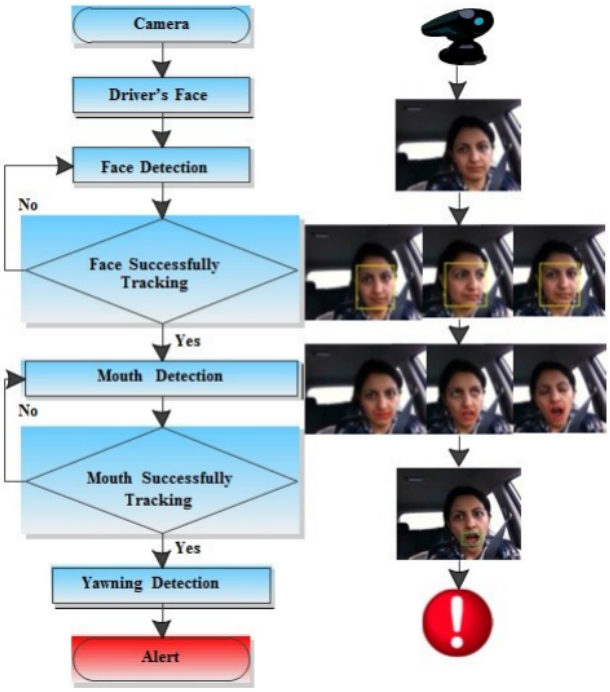


Figure 4.5: Face Detection Algorithm

Data collection means collecting data from various sources. And after that collection of data, there are many more steps that should be fulfilled in order to train and test these datasets. Some of those steps are mentioned below:

1. Downloading Sample Images
2. Preparing the Environment:
3. Annotate the Dataset
4. Train to get Results:

4.2 Downloading Sample Images

In case of downloading sample images, we have used an open source to get image. As we are working on detecting people from images, we have collected a huge amount of dataset containing images of people.

We have used ”Open Images Dataset V7” for this specific purpose. It contains a whole lot of data that we need to carry out this operation. An open-source license provides certain permissions to users regarding the usage, distribution, modification, and sharing of the image. It’s a part of the broader concept of open-source software and content, which promotes collaboration, sharing, and transparency.

The main theme of this thesis was recognizing people so that a vehicle can detect what lies or stand in front of it and to avoid collision, we have downloaded only those specific images to train that will help the machine to detect person or pedestrians. Before using an image labeled as open source, it’s essential to review the specific terms of the open-source license associated with that image to understand how you can use, modify, and share it.

4.3 Preparing the Environment

To prepare the environment, we have to first create the conda environment in anaconda promt.

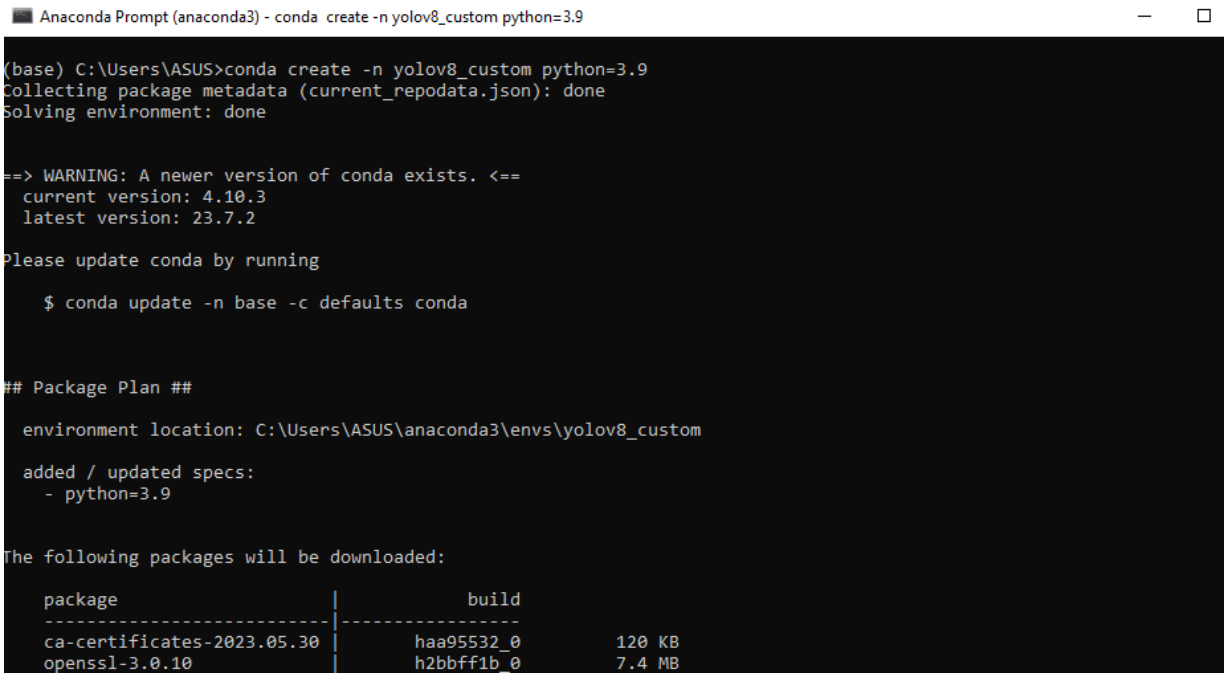


Figure 4.6: Creating Conda Enviornment (Part-1)

We have to download package so that it can render all YOLO-V8 libraries and carry out the whole detecting process.

Most YOLO implementations require specific libraries and frameworks, such as Darknet (the original YOLO framework), TensorFlow, or PyTorch. Follow the installation instructions provided in the repository's README file to set up the required environment.

YOLO models need pre-trained weights to perform object detection. These weights are typically available for download in the repository. You might need to place these weights in the correct directory according to the repository's structure.

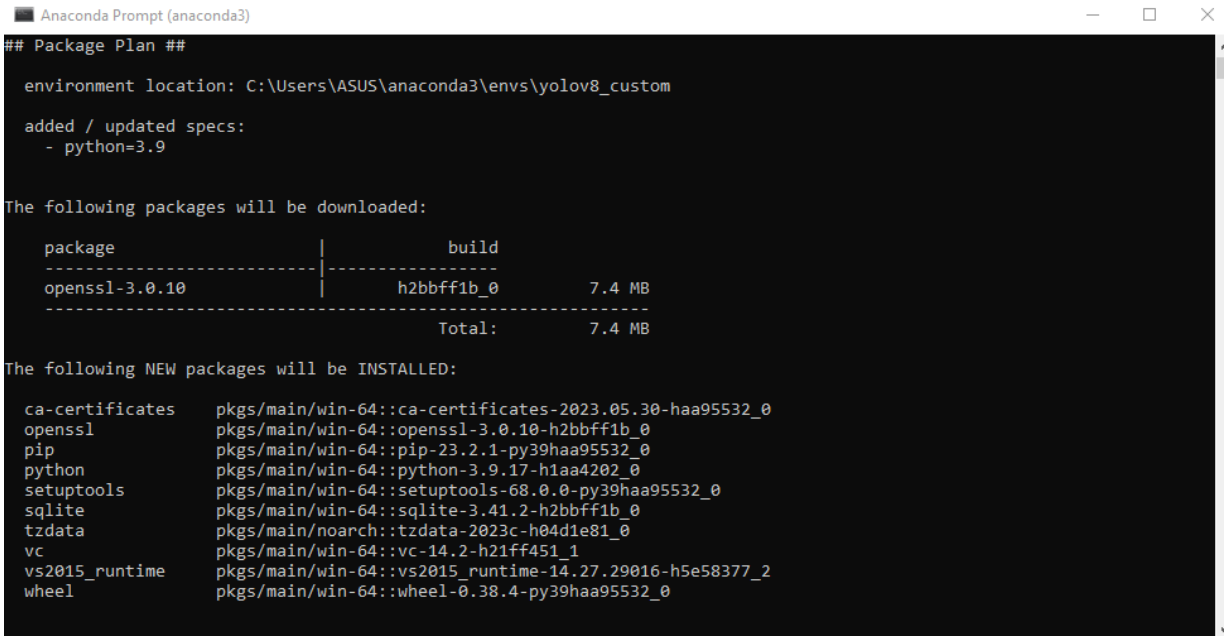


Figure 4.7: Creating Conda Enviornment (Part-2)

After the library installation is finished, we have to call the directory where all our sample images are saved for detecting process.

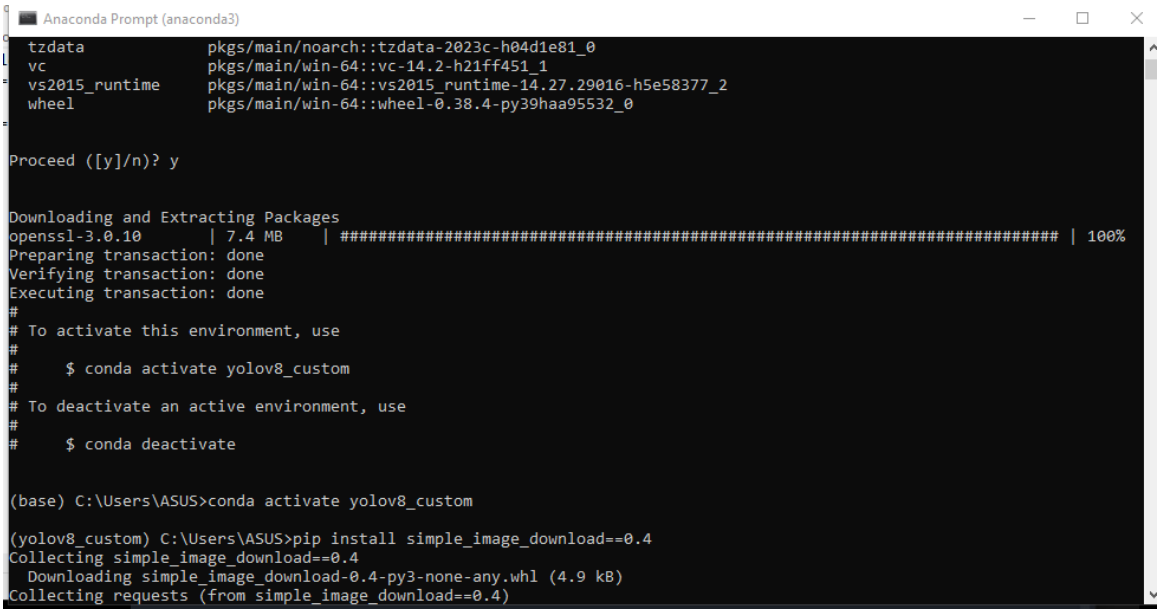


Figure 4.8: Creating Conda Enviornment (Part-3)

We have to label them with bounding boxes using YOLO-V8 model and run that algorithm to detect pedestrians on the road.

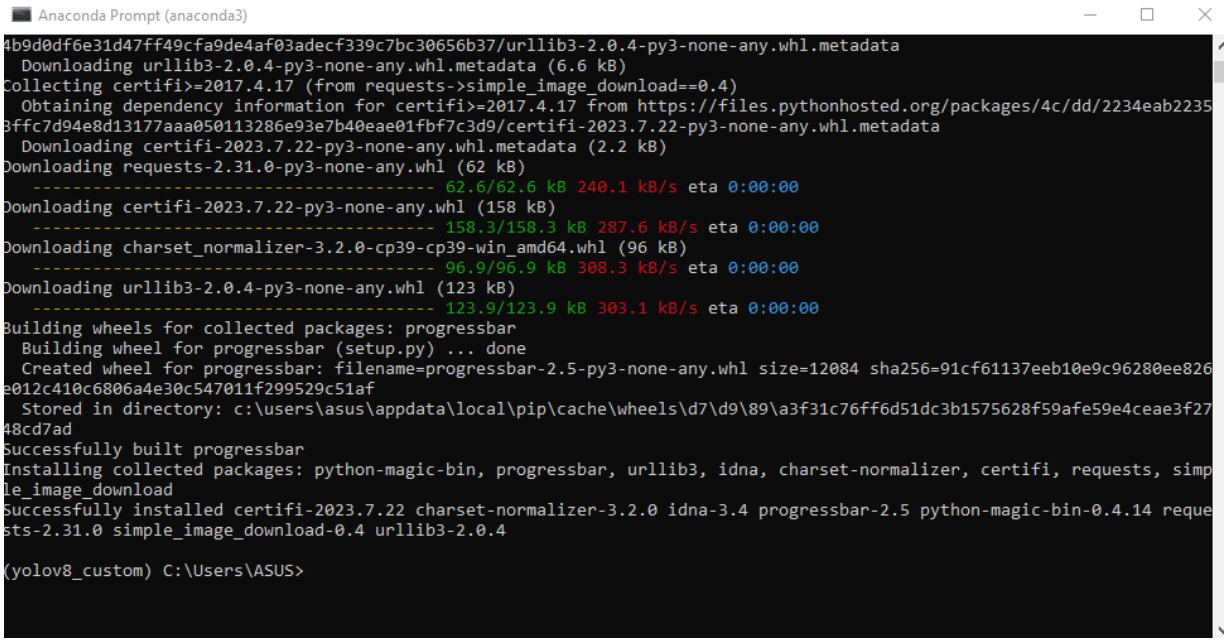


Figure 4.9: Creating Conda Enviornment (Part-4)

4.4 Annotate the Dataset

Annotations mean labelling the dataset with appropriate figuaratives. We can create bounding boxes that will keep only the pictures of people inside those boxes so that when we train this model, the machine or vehicle can predict the people in front of it.

4.5 Train to get Results

Finally we need to train the annotated dataset to get our desired results. We can run those annotated images into the YOLO-V8 model and it can detect all the portions of the images that contains pedestrians or people in general. Then , to find out the overall result or performance in this object detection model, we can create confusion matrix, F-1 score, recall or precision and even accuracy one by one. We can do everything that we need to do in order to rate its performance by running it itno the YOLO-V8 model.

4.6 Classification

By performing the above dimensionality reduction, we remove the dimentionality of our dataset by projecting them on another space. In that reduced space, classification has been performed.

4.6.1 Distance Classifier

Euclidean Distance from test image that is projected from the train images that has been calculated.[17]

The necessity of building a model for itself is no need in this case. The minimum distance that has been acquired means the most similar trained image. Those images belong to the same class. It reduces the extra tuning of several parameters. But the main disadvantage is the computational cost is too high.

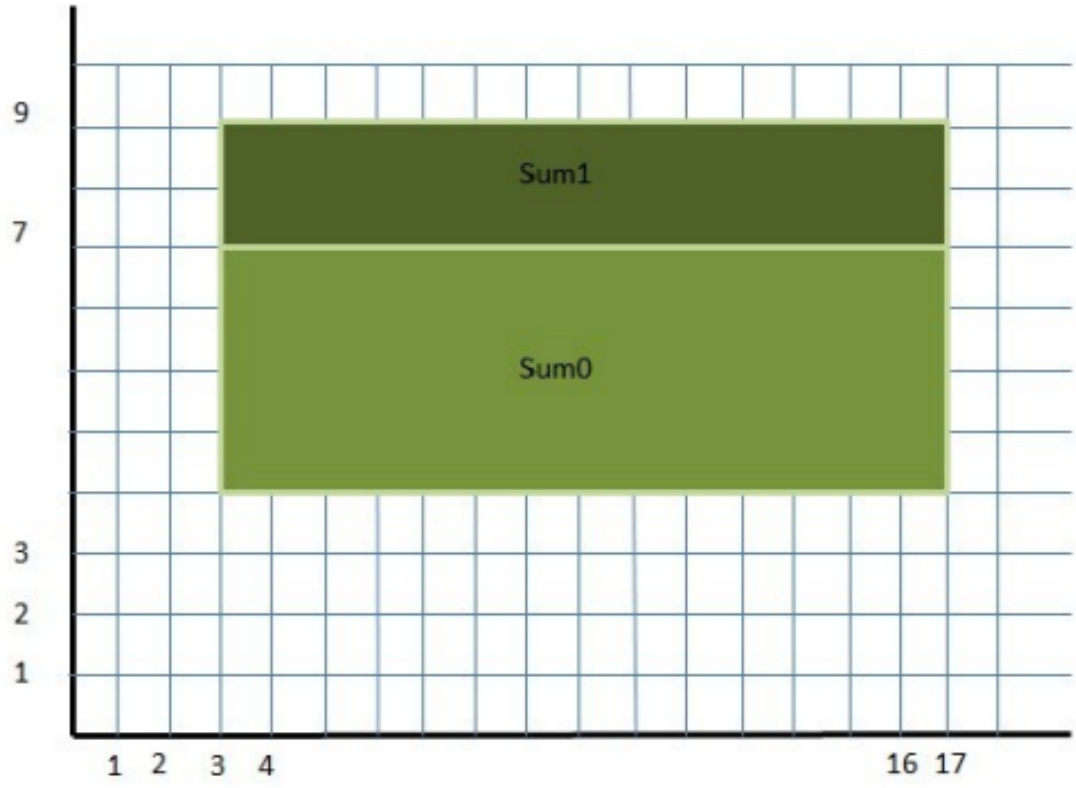


Figure 4.10: Comparison of two sample blocks/features

4.6.2 Support Vector Machine

It is a powerful tool that uses labeled training sample to classify new observation by finding decision boundary that is optimal that maximizes the training data. The recognition accuracy of the model depends on how the data in being trained inside of it.[18]

4.7 Chapter Summary

In this chapter, we have discussed the models that are available in case of image detection and classification. Various approaches can be made to distinct the desired object from an image set. We will train those models and find out which one gives us the better performance among them.

Chapter 5

Data collection, training and testing

In data collection process, first and foremost task is to collect the data from somewhere, we can collect images by clicking random pictures that contains the desired objects that need to be detected or recognized which is done manually, or we could just gather the images that we want to use in this whole detection program by some online open sources.

For this specific purpose, we are using ”Open Images Dataset V7” for our specific purpose. We can download as many as we need to train our datasets and labelling them or annotate them with desired causes.

After collecting all the data, we have to annotate the data according to our use. For the work of annotation, we use bounding boxes to cover the area of a image that carries the specific or desired object, which in this case, people in general.

When a bounding box is created, it automatically generates a text document that contains the information of a bounding region. A bounding box, often referred to simply as a ”bbox,” is a rectangular frame that surrounds an object or region of interest in an image or a two-dimensional space. Bounding boxes are commonly used in computer vision and image processing tasks, particularly in object detection, tracking, and annotation.

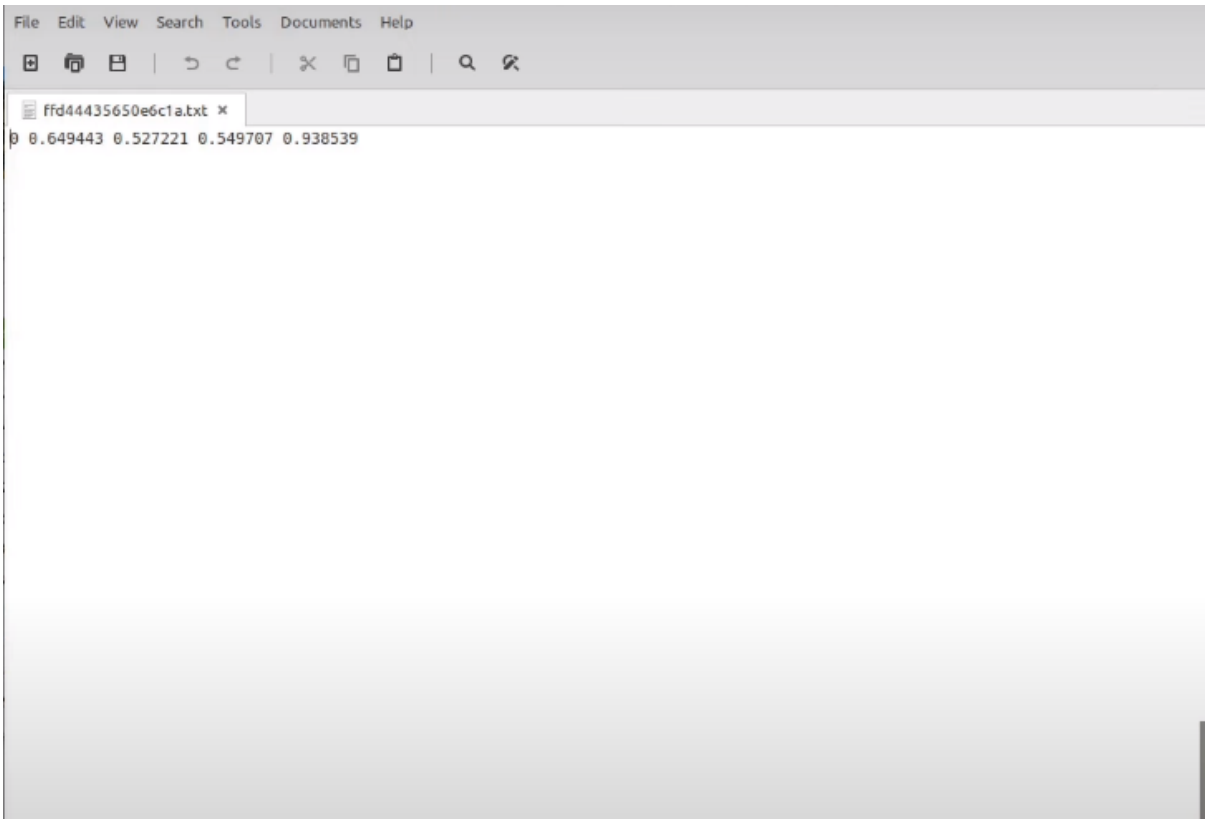


Figure 5.1: Parameters containing bounding regions

For the above example, we can see that the numbers that is generated after annotating an image. There is only a row of these sequence of numbers. It number of rows depends on the number of objects found in that specific image or the number of specific object that is detected.

Bounding boxes can be used to extract the region of interest from the image, apply further analysis to the object contained within the box, and make predictions about the object's class and other attributes.

For instance, in object detection models like YOLO (You Only Look Once) and Faster R-CNN, bounding boxes are predicted to localize and identify objects within an image. In tracking algorithms, bounding boxes can be used to follow objects across frames in videos.

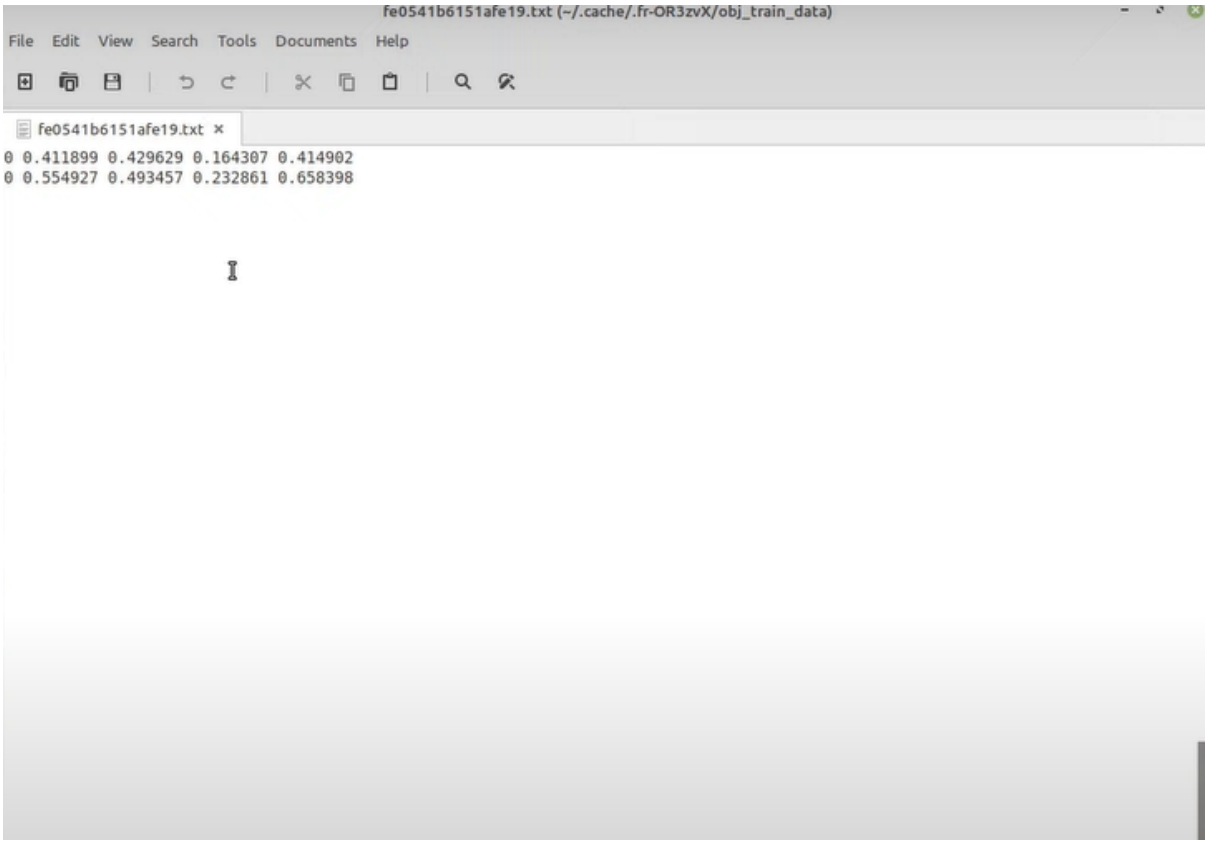


Figure 5.2: Two rows containing parameters of bounding regions

In the above example, we can see that there are two rows of numbers that contains parameters of the bounding regions. It basically means that, there are two bounding regions that contains or detects people or pedestrians in that specific image sample.

In object detection tasks, a bounding box is drawn around each object of interest in an image to indicate its location and extent. The bounding box is defined by four parameters: the coordinates of the top-left corner (x, y) and the coordinates of the bottom-right corner (x, y). Bounding boxes

have numerous uses in various fields, particularly in computer vision and image processing. They provide a structured way to define and locate regions of interest within images or other forms of data.

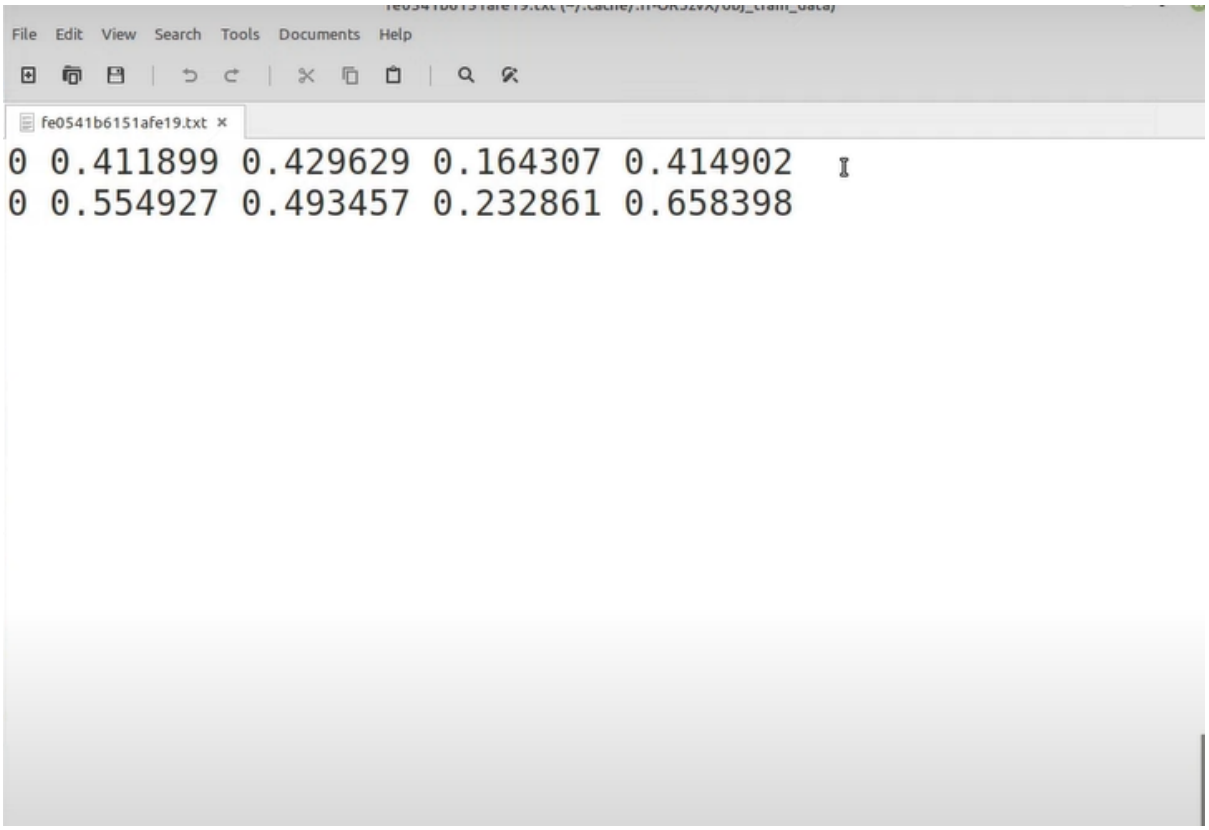


Figure 5.3: Four attributes in one bounding box

In this figure, we can see that, the rows containing numbers as parameters are actually for four attributes of a bounding box. They are class labels, position of the center, width and height of a bounding box.

Bounding boxes are extensively used in object detection tasks. They mark the positions and extents of objects within images, enabling algorithms to recognize and classify objects present in a scene. Object detection has applications in autonomous driving, surveillance, and more.

In video analysis and tracking, bounding boxes help track objects across frames. Algorithms use the position and size information from bounding boxes to maintain the identity of objects as they move and change appearance.

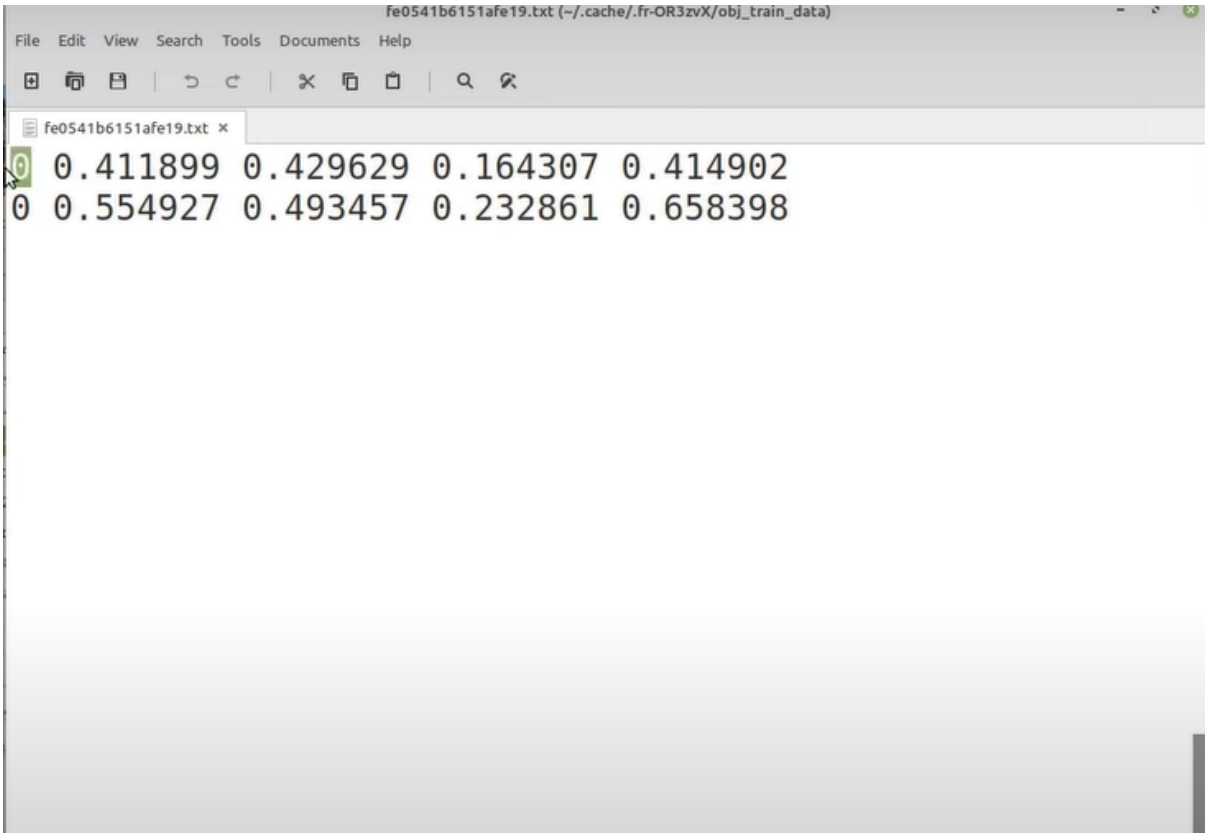


Figure 5.4: First Attribute: Class

In this figure, the first number is zero, that means the there is only one class label in that sample image. As we only need to annotate the area where a person can be seen and detected or recognized, we only mentioned one class label.

When creating labeled datasets for training machine learning models, bounding boxes are often used to annotate objects of interest within images. This is common in tasks like training object detection or localization models.

In semantic segmentation, where pixels are assigned to specific classes, bounding boxes can be used to initialize or guide segmentation algorithms. This provides a rough estimate of where certain objects are located.

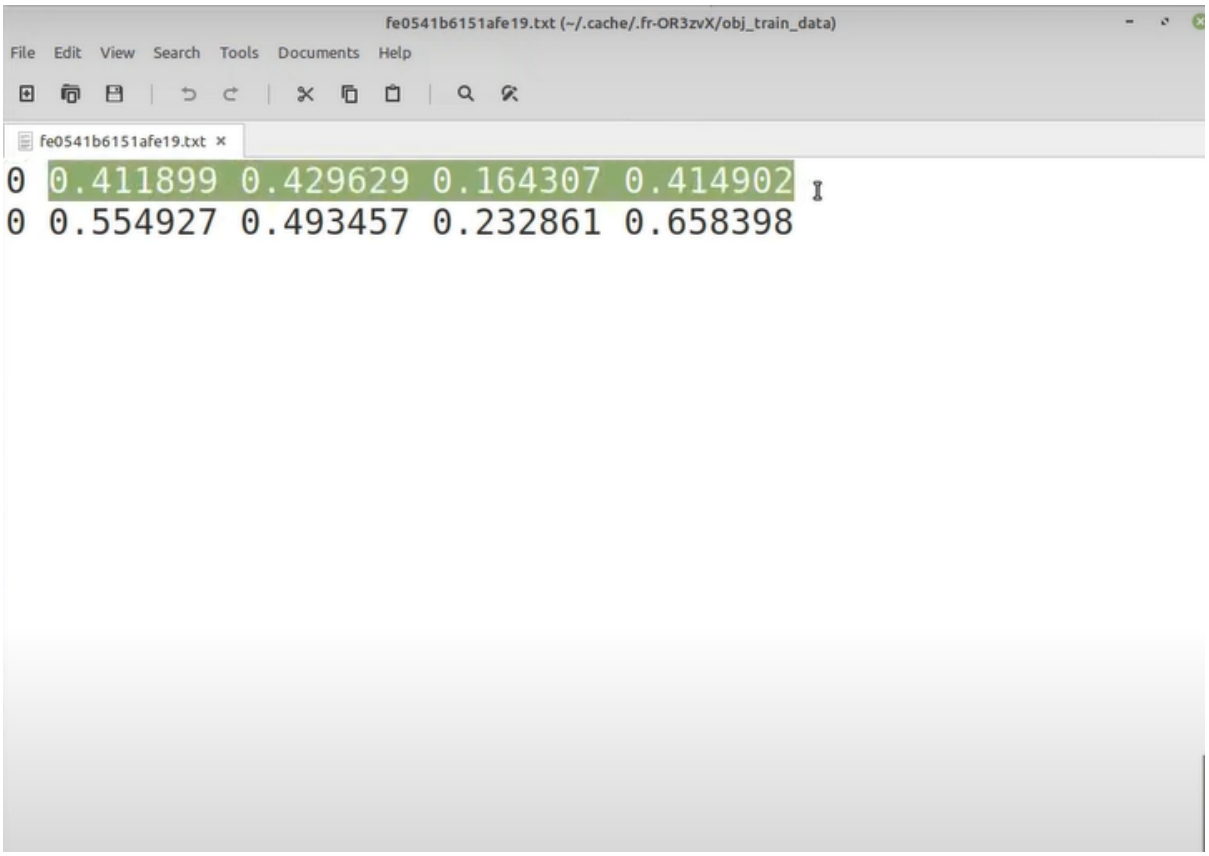


Figure 5.5: Rest of the Attributes: Bounding Box

In this picture, we can see that the last or the rest of the attributes depicts the parameters of a bounding box. After the class label, the rest of the attributes that remains are:- center position, width and height.

Similar to semantic segmentation, bounding boxes can also help in the initialization of instance segmentation, which aims to differentiate between individual instances of objects within an image.

Bounding boxes are used to extract specific regions from an image for further analysis. This can involve cropping the image to focus only on the area of interest.

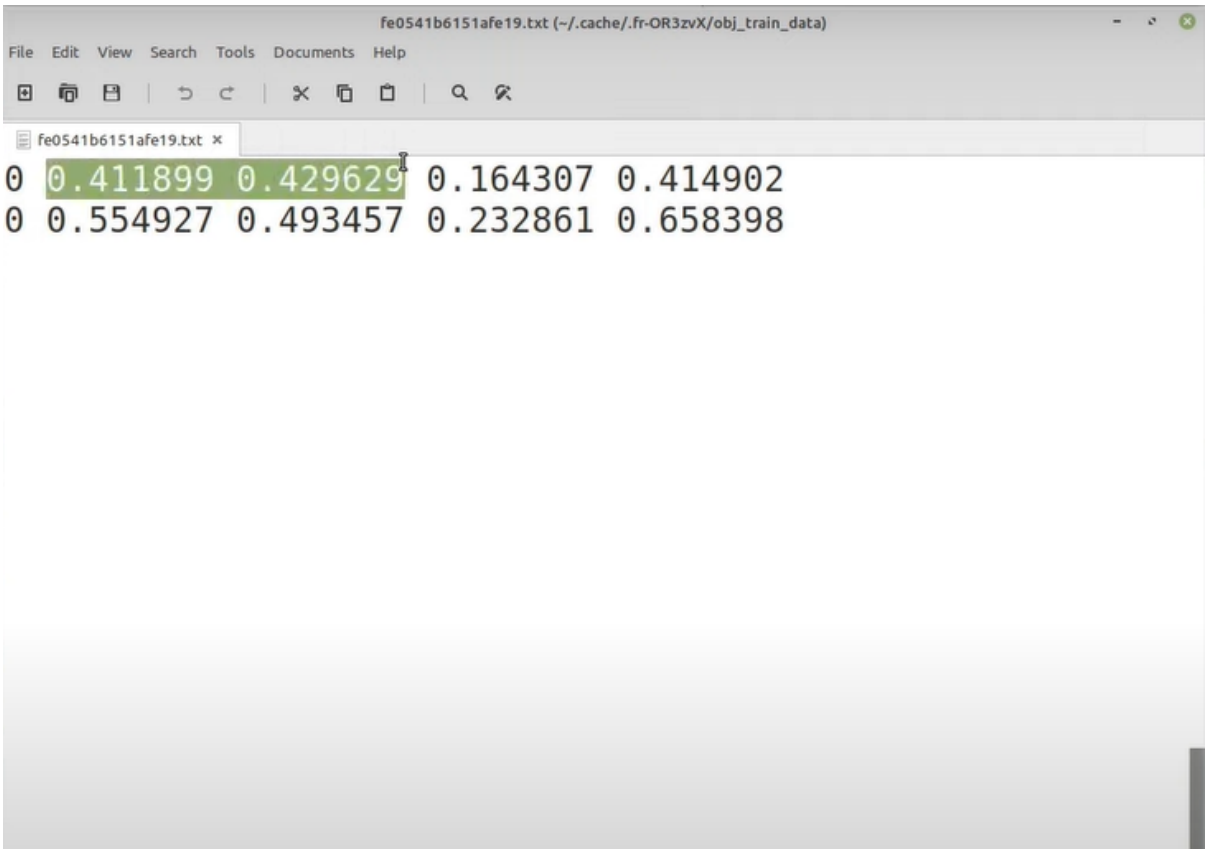


Figure 5.6: Center positioning of a bounding box

The center of a bounding box is declared as an attribute after the declaration of class label in the first place. The position of the center is determined by 2-D or 2 dimensional axis system (X and Y). The first attribute after the class label determines the position of X axis and the second parameter determines the position of Y axis.

Bounding boxes help identify the position and scale of objects within an image, which is essential in applications where the exact location of objects matters, such as medical imaging and satellite imagery analysis.

Bounding boxes are fundamental in image annotation tools that allow human annotators to draw boxes around objects to train or evaluate machine learning models.

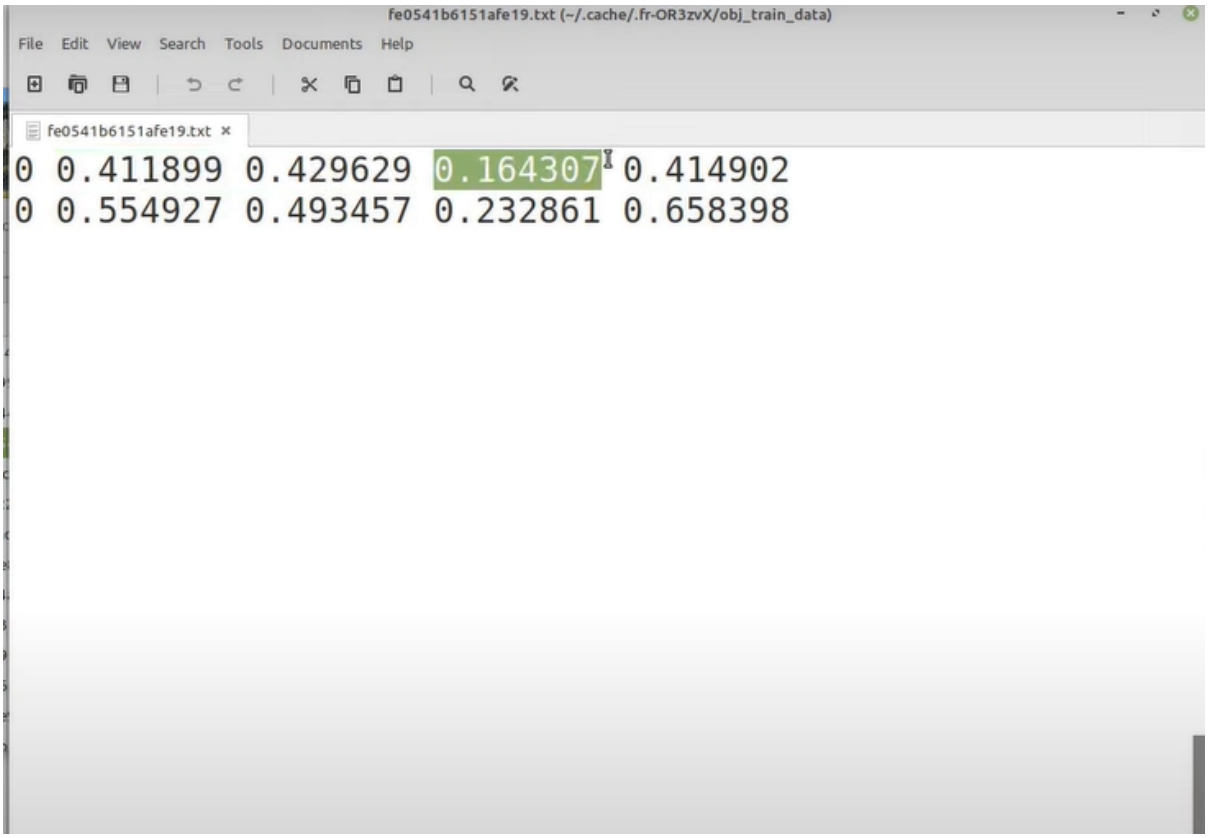


Figure 5.7: Width of a bounding box

Here, we can see that, the next parameter that comes after center positioning is the width of a bounding box. The width of a bounding box is the third parameter of the overall bounding box.

Bounding boxes can be used to track the interactions between objects in a scene, such as tracking hands in gesture recognition or tracking interactions between objects in sports analysis.

Bounding boxes are often used to visualize the results of object detection and tracking algorithms, showing where the algorithm has identified and localized objects. Objects are hardly seen in an image or a video are easily detected and created a box or boundary around them to specify the object.

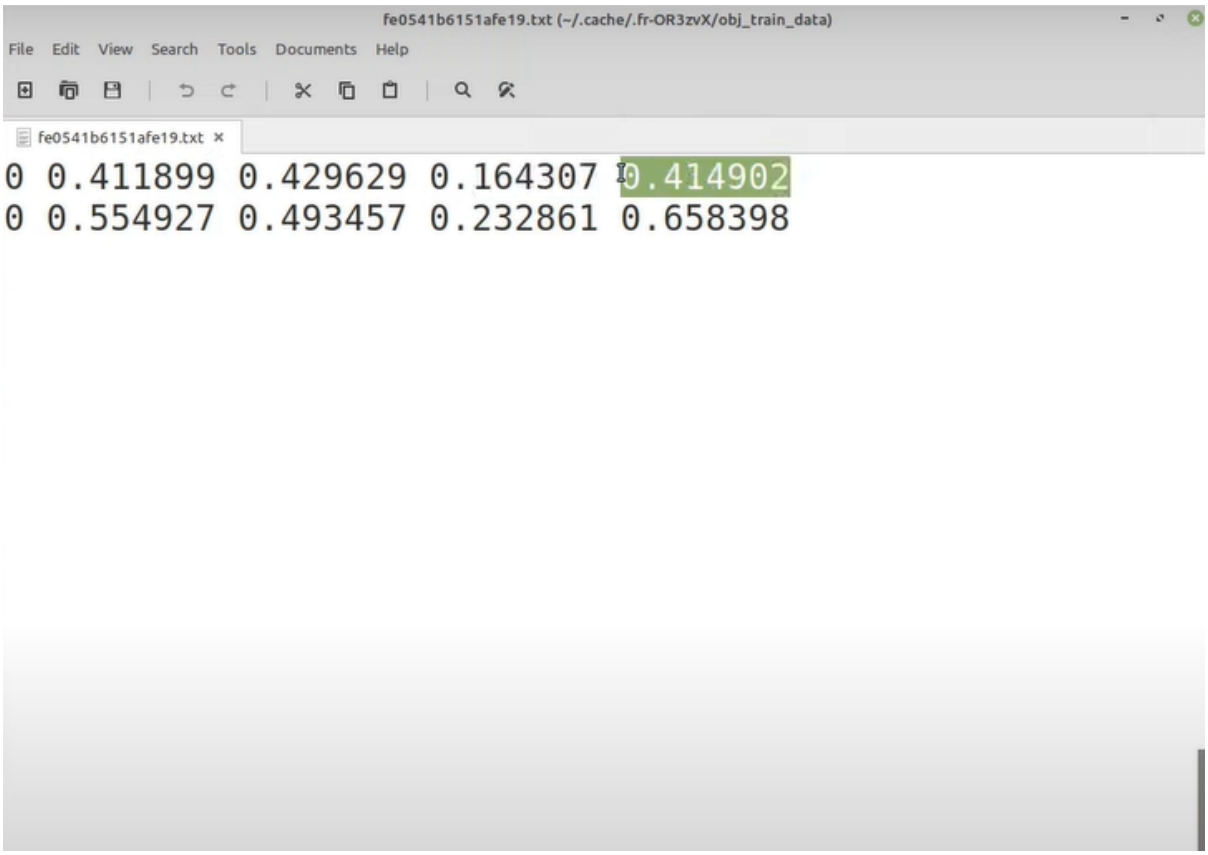


Figure 5.8: Height of a bounding box

And finally, the last parameter of the bounding box is denoted as the height of the bounding box. It is the fourth and last parameter of a bounding box that contains the information of the height of a bounding box.

Bounding boxes are used to define the area occupied by an object to be removed or replaced in image editing tasks. Bounding boxes assist in handling situations where objects are partially occluded by other objects or elements in the scene.

Overall, bounding boxes serve as a structured way to define and communicate the positions and extents of objects within images and videos. They are a foundational concept in various computer vision tasks, enabling algorithms to understand and manipulate visual data effectively.

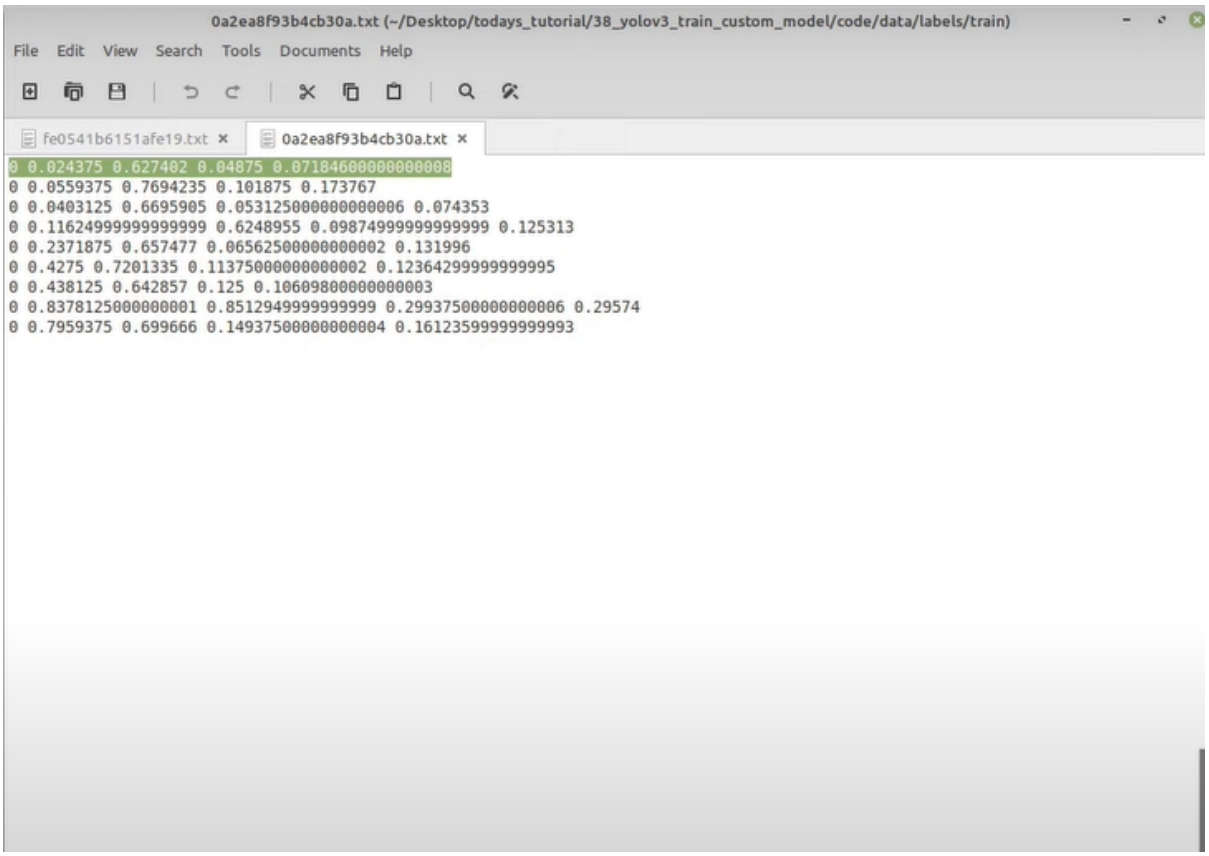


Figure 5.9: Multiple bounding boxes in a sample image

In the above figure, we can understand the fact that there are more than one bounding boxes containing specific bounding parameters. All those bounding boxes are from a single sample image. That means, in that specific image, we have found multiple numbers of our desired objects. In this case, multiple person might be showing in that particular sample image.

A list of the collection of our training dataset is given on the next page:

To gather a huge amount of dataset, we invited some people of different ages that voluntarily participated in our thesis program.[19] As there were a wide variety of dataset which included different ages and differet genders of people, it gave us more valid and reliable result in case of accuracy.

Table 5.1: Parameters of Different Annotations

Image no:	Center Position	Width	Height
1	0.045677 , 0.0237654	0.4757683	0.0876535
2	0.0844754, 0.04857464	0.046638934	0.08477645
3	0.0364578, 0.023746574	0.075663535	0.01276584
4	0.045654857, 0.053647585	0.02357586	0.0768465475
5	0.00045675, 0.000476263	0.0028384765	0.0000506978
6	0.000338487, 0.000067436553	0.09587364	0.00487506060
7	0.045677 , 0.0237654	0.4757683	0.0876535
8	0.0844754, 0.04857464	0.046638934	0.08477645
9	0.0364578, 0.023746574	0.075663535	0.01276584
10	0.045654857, 0.05364758	0.02357586	0.0768465475
11	0.00045675, 0.000476263	0.0028384765	0.0000506978
12	0.000338487, 0.00006743655	0.09587364	0.00487506060
13	0.045677 , 0.0237654	0.4757683	0.0876535
14	0.0844754, 0.04857464	0.046638934	0.08477645
15	0.0364578, 0.023746574	0.075663535	0.01276584
16	0.045654857, 0.053647585	0.02357586	0.0768465475
17	0.00045675, 0.000476263	0.0028384765	0.0000506978
18	0.000338487, 0.00006743655	0.0958736	0.00487506060
19	0.045677 , 0.0237654	0.4757683	0.0876535
20	0.0844754, 0.04857464	0.046638934	0.08477645
21	0.0364578, 0.023746574	0.075663535	0.01276584
22	0.045654857, 0.053647585	0.02357586	0.0768465475
23	0.00045675, 0.000476263	0.0028384765	0.0000506978
24	0.000338487, 0.00006743655	0.0958736	0.00487506060
25	0.045677 , 0.0237654	0.4757683	0.0876535
26	0.0844754, 0.04857464	0.04663893	0.08477645
27	0.0364578, 0.023746574	0.075663535	0.0127658
28	0.045654857, 0.053647585	0.0235758	0.0768465475
29	0.00045675, 0.000476263	0.0028384765	0.0000506978
30	0.000338487, 0.000067436553	0.09587364	0.00487506060
31	0.045677 , 0.0237654	0.4757683	0.0876535
32	0.0844754, 0.04857464	0.046638934	0.08477645

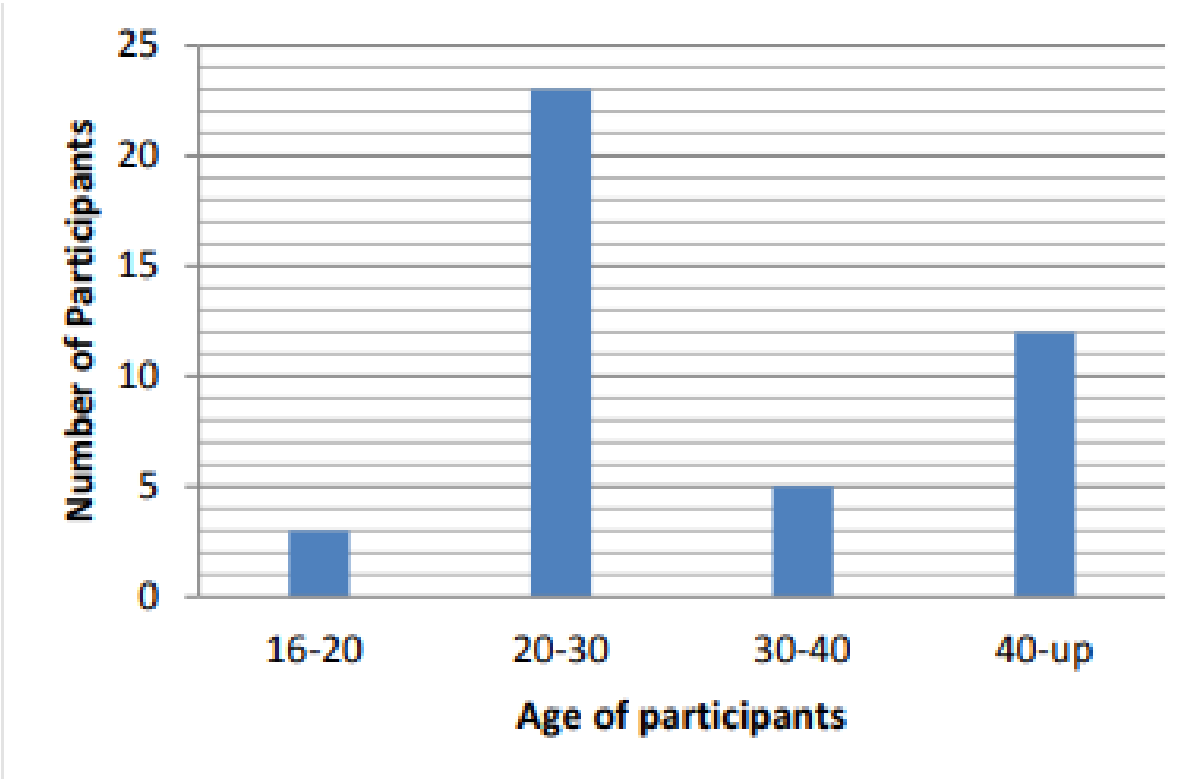


Figure 5.10: Male Participation

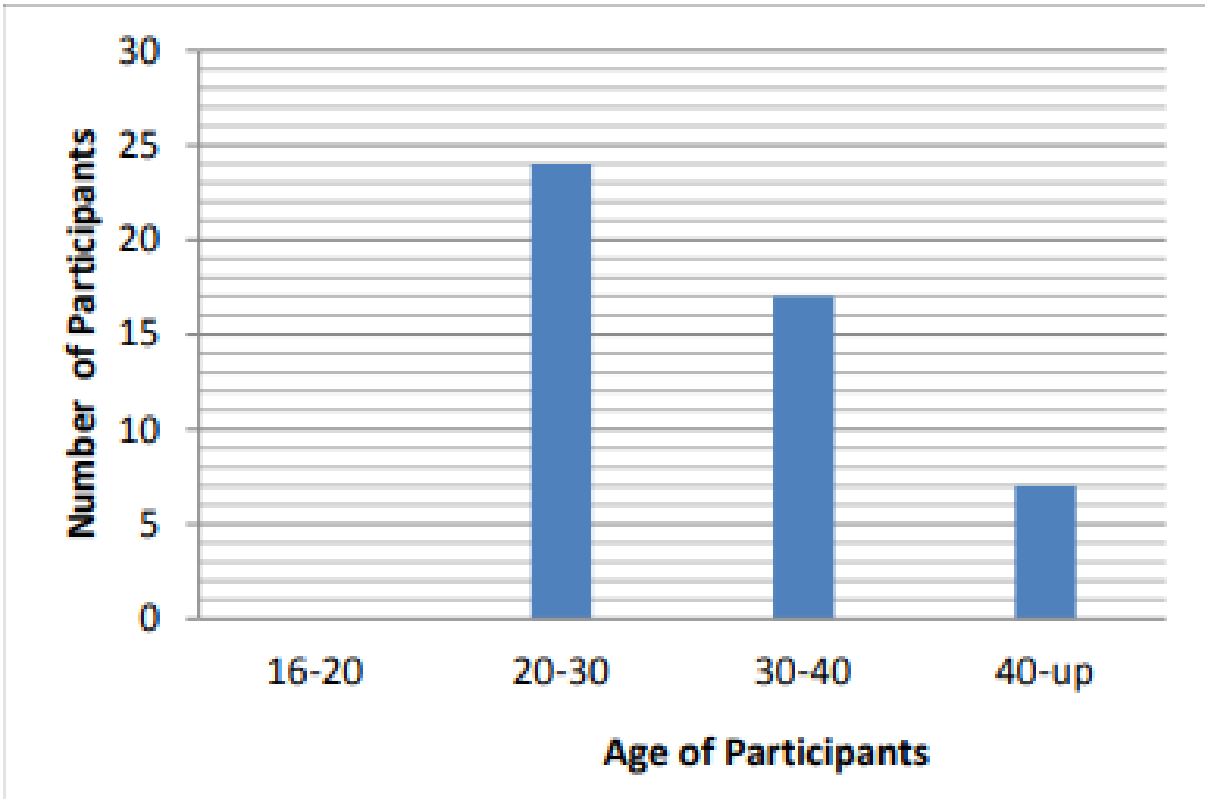


Figure 5.11: Female Participation

The dataset that we found in the given period of time, we managed the dataset by providing different measures such as liability and YOLO-V8 is the recent model that has been used in this case[20]. Yolo-V8 is the latest version of this model that has been released in 2023.

5.1 Training and Testing

The collection of data is used in APEX board. Before transferring the data, we need to stimulate the dataset and reduce the feature according to its use. It was tested with two previous dataset and in OpenCV format which renders to the face and mouth detection.

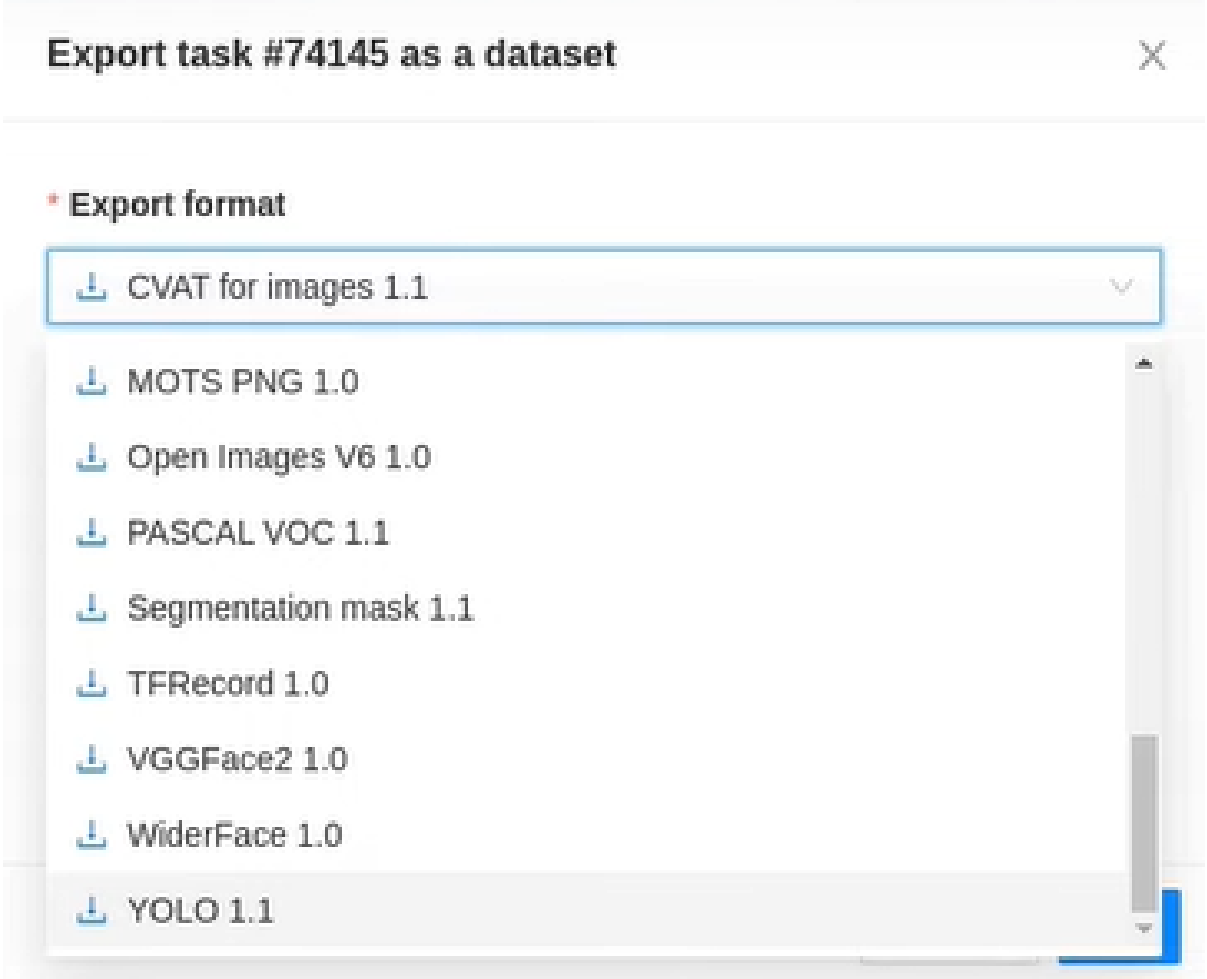


Figure 5.12: Creating Yolo Environment

In this figure, we have created an environment that is suitable to run the dataset in YOLO format.

By doing so, all the sample data will convert into YOLO format and can be executed into YOLO engine to detect objects from them.

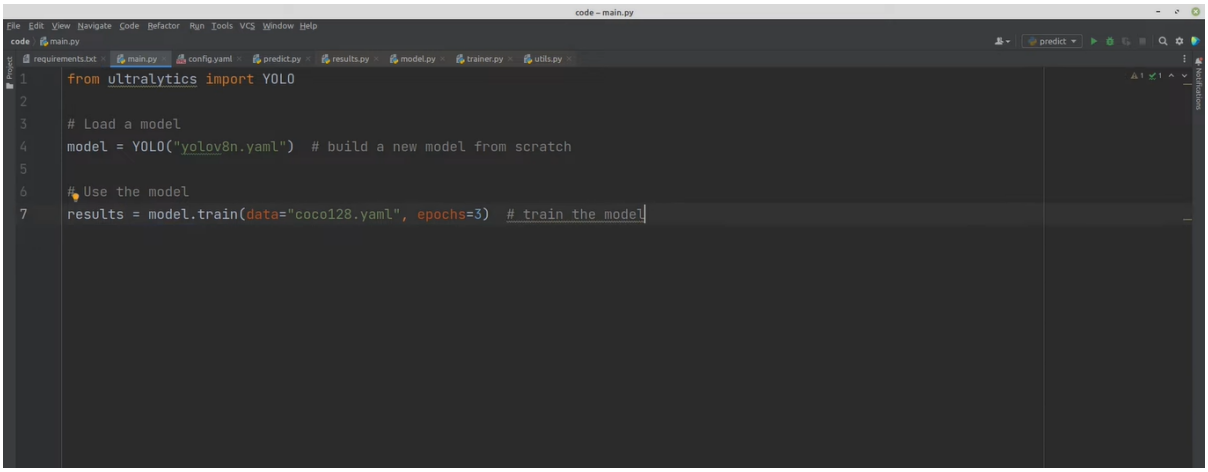
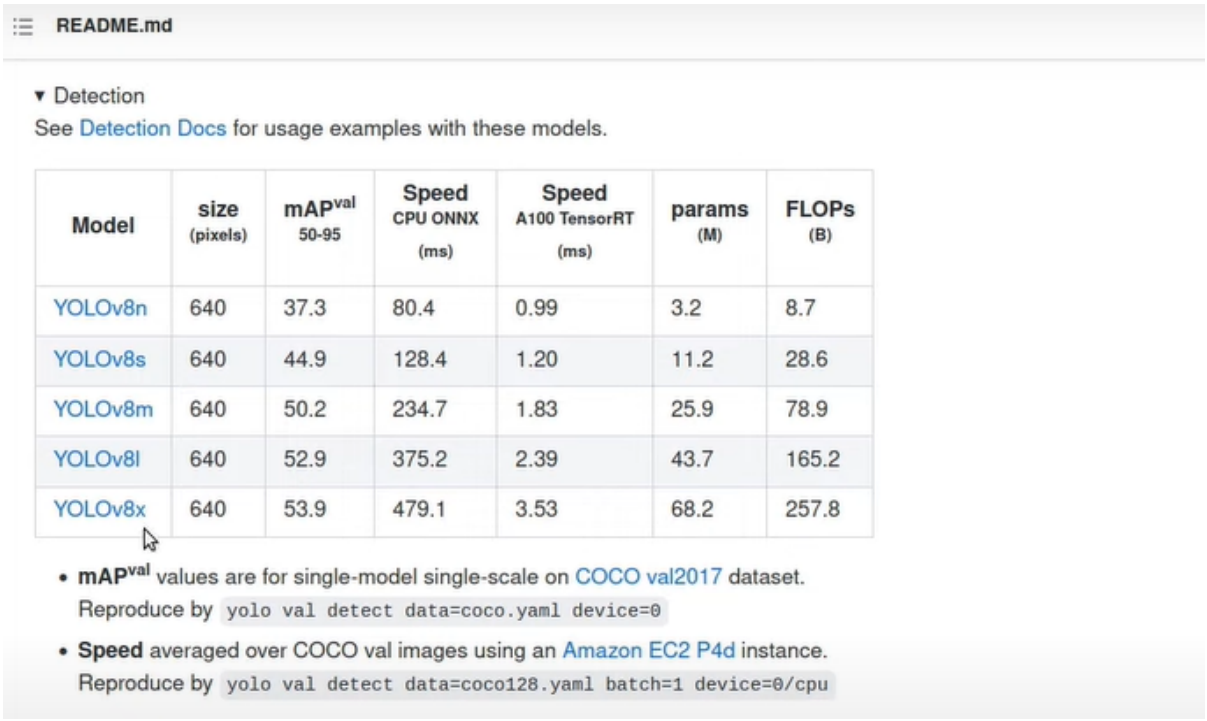


Figure 5.13: Creating Yolo Environment in a local environment (PyCharm/anaconda)

We can also run this environment on some local environments like PyCharm or Anaconda. In online platform, the code snippets are almost the same. We can execute all these datasets on a local environment just like an online platform and detect objects by using YOLO.


README.md

▼ Detection

See Detection Docs for usage examples with these models.

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

- **mAP^{val}** values are for single-model single-scale on COCO val2017 dataset.
  Reproduce by `yolo val detect data=coco.yaml device=0`
- **Speed** averaged over COCO val images using an Amazon EC2 P4d instance.
  Reproduce by `yolo val detect data=coco128.yaml batch=1 device=0/cpu`

Figure 5.14: Different Versions of YOLO-V8

Here we can observe that there are different versions available for YOLO-V8. We can use whatever we want according to our need for research purposes.

For example, if we want to train only a little amount of data, we can use YOLOv8n which stands for nano version of YOLO-V8. If we work with samll amount of data, we can use YOLOv8s which is the smaller version of YOLO-V8. We can use YOLOv8m if we have medium amount of data. And for large and extra large amount of data, we can use the versions YOLOv8l and YOLOv8x respectively.

On the APEX board, we only used two times for the demo and other times for data collection.[21]. The success rate for face detection was almost 90 percent where all the other attributes were 82 percent and 78 percent. But the frameset was not fixed because the number of iteration is dependent on the possibility of locating the face and mouth.

$$Eye_map = (1/3) * ((C_b)^2 + (C_r)^2 + (C_b/C_r)) \tag{5.1}$$

The average speed of detection was 2-3 fps in cases of both face and mouth detection.[22] Which is more than enough in case of yawning detection. The speed of detection can be much higher ,almost 20fps but in that case, the result would not be this much reliable.

To detect and analyze, we have to gather and prepare a labeled dataset containing images and corresponding bounding box annotations. Each annotation should include the class label and coordinates of the bounding box (usually in the format of xmin, ymin, xmax, ymax).

Then we have to modify the model configuration file to specify the number of classes we want to detect, anchors for bounding box prediction, network architecture details, and training settings.

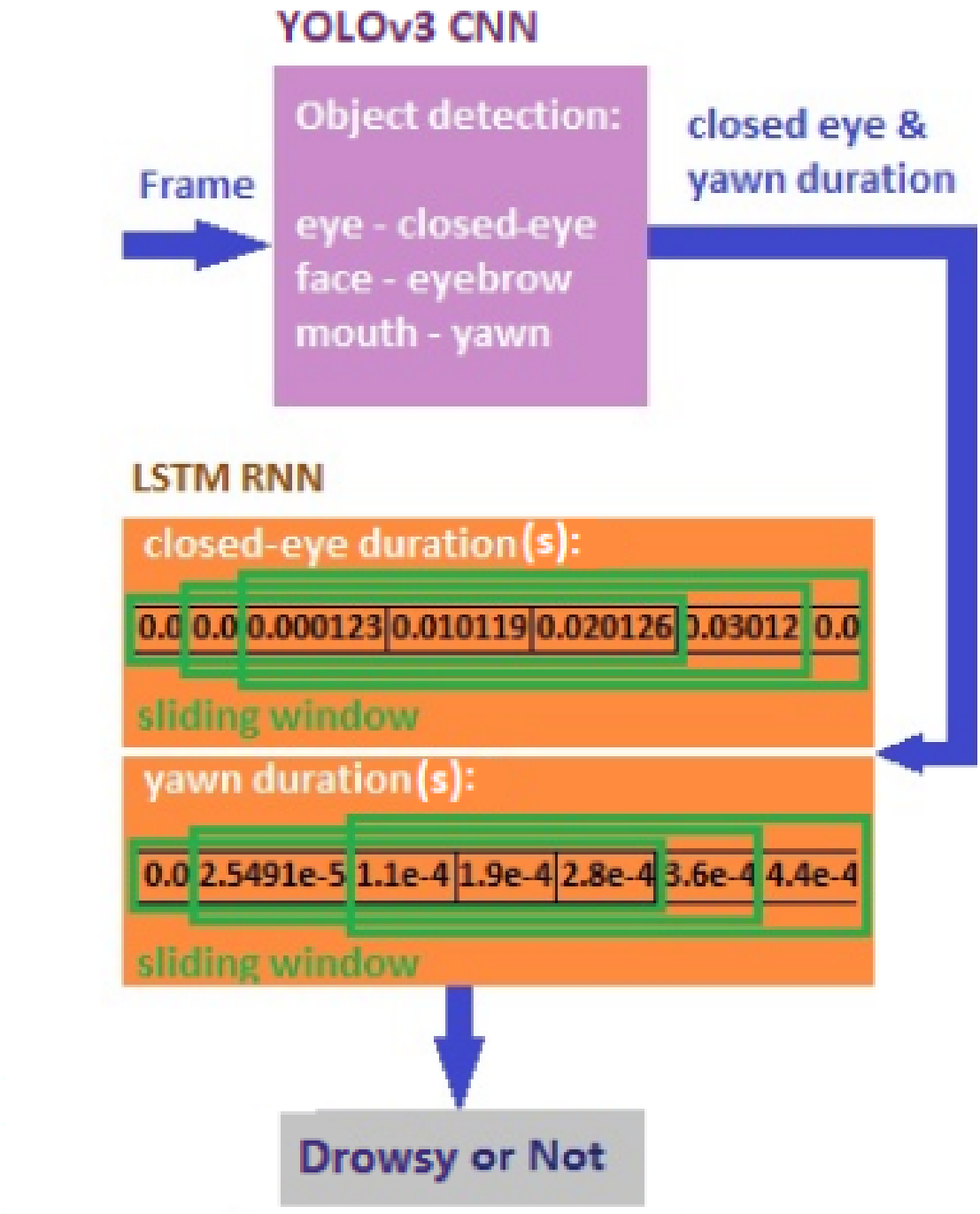


Figure 5.15: LSTM Analysis

We can use YOLO to detect objects in video frames, and then use LSTM to track the detected objects over time. This could involve associating objects detected in consecutive frames to maintain object identity and predict future object locations.

However, the direct combination of YOLO and LSTM in a single architecture is not a common approach, as they serve different purposes and operate on different types of data.

5.2 Training Neural Network

Training the neural network is a must in case of machine learning approaches. During the tracking system, it needed to fall back to back propagation. [23]The easier way to train the model is to add wights to each and every vector parameter and modify the weights in according to the related network.

$$mouth - map = (C_r)^2 * ((C_r)^2 - ((n * C_r)/C_b)) \tag{5.2}$$

Neural network contains the property of weight modification and adaptation techniques. The overall success rate for yawning detection in case of APEX board was 80 percent while training in the Neural Network. [24].

5.3 Testing the trained model

After the model is trained, it is time to test the trained model. In previous section we discussed how we can train the model in APEX board and the given neural network. At first we need to collect images from a source of datasets. Then we need to specify the objects in bounding boxes to name an object. That trained set is now used for machine learning[25] models such as YOLO-V8 or any other models. This kind of trained data are used for object detection such as mouth and eyes and their locations in a face.

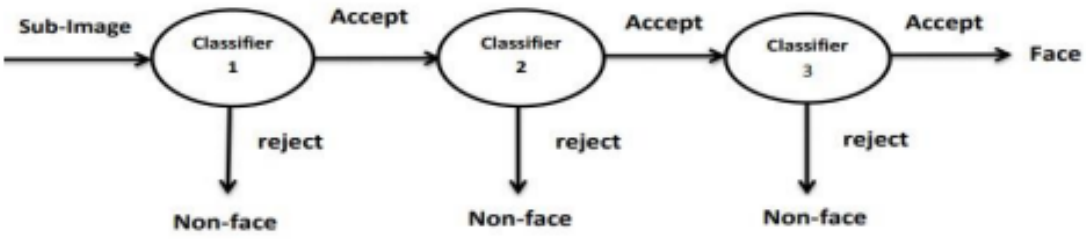


Figure 5.16: Cascade of Classifiers

5.4 Chapter summary

In this chapter, we broadly discussed about the data collecting methods and techniques which will pave the way to our desired result. In the field of object detection, we need to train the machine that we will use to extract feature, which will eventually lead us to object detection.

Chapter 6

Result and Performance Analysis

In this paper, the first and foremost goal is to detect and object and take necessary steps according to that object. In this case, we used Yolo-V8 which is the updated version of its model and in comparison to this version, we used viola jones method and other yolo models to analyze which one performs better or accurately and which does not[25].

There are two types of face recognition techniques, one is called the static technique ,which means the detection of an image is done from the image that is stored in memory. And another way is the real time video technique which implies on the online image or video and stores the data dynamically to detect or recognize objects.[26]. We have work on the following datasets that are given in static techniques. Which means we have static images that is stored in the computer memory and use it to detect features of a face. [27]

Moreover, we use sigmoid function as bounding boxes:

$$b_w = p_w * e^(tw) \tag{6.1}$$

$$b_h = p_h * e^(th) \tag{6.2}$$

6.1 Technique For Face Recognition System

YOLO-v8 or You-Only-Look-Once is a model that can be effectively used in many terms to detect and image. The most widely used model to train the dataset and detect object in a wide area of image collections. In this model, the accuracy is gained using labled data and classes.[28]

$$E_(Image) = W_(Line)E_(Line) + W_(Edge)E_(Edge) + W_(Term)E_(Term) \quad (6.3)$$

6.1.1 Dataset Preparation

At first we need to collect the datasets that are containing or having the desired images that we need to detect. From this wide range of data, we only need to gather those data that has our object. Then we need to annotate the images in specific proportions for example, around the face and near the area of mouth and eyes. We have to create bounding boxes and label that object with unique classes for each individuals.[29]

6.1.2 Pre Trained Weights

Then we have to gather all the pre trained weights for the specific YOLO version that we are working with. The pre trained data or weights are unique for each versions of YOLO models. Thus to gain the maximum accuracy, we have to get the weight update of the specific set of YOLO-V8.

6.1.3 Feature Extraction

[30] Each particular yolo model has specific ways to extract features. To train and modify the features of a large amount of dataset, we need to configure the model that we are having so that it can extract features from the intermediate layers.[31]

6.1.4 Training

For training purpose, we need to add an embedded layer that will compile the extracted features to the latest version of YOLO model and crate an embedded vector space of fixed size and weights. Now we need to train the modified yolo model with the annotated image datasets and

these data are featured into that embedded layer. Then we have to use triplet loss function that optimizes the networks ability in detecting objects. Then compare the embeddings that detected faces with the embeddings that has facial images in our dataset.[32] Use a similarity fucntion like cosine similarity to gather the similar images all together in a cluster. After similarity between the embeddings crosses a threshold value, then classify the face from that obejct of that image.

6.1.5 Face detection and recognition

After training the datasets, we have to find the embedded layers comparison to detect the object from those image dataset. We need to clarify the data that a similar will belong together as the use and measure of triple loss function and cosine similarity.[33]

6.2 Performance Analysis of the Face Recognition System

Performing a performance analysis and comparison of face recognition between YOLO (You Only Look Once) and Viola-Jones algorithms involves evaluating their accuracy, speed, robustness, and efficiency.

Choose a diverse dataset with a variety of face images and backgrounds for a fair evaluation.[34] Annotate the dataset with ground truth bounding boxes.

Then we have to calculate precision, recall, F-1 score in both the cases of YOLO-v8 and Viola Jones. We have to compare the detection and recognition accuracy in terms of both the algoithms.

[35] Analyze false positive and false negative cases for both algorithms to understand their failure modes.Analyze the memory and computational resource requirements for both algorithms.Measure the processing speed of both algorithms on the same hardware platform.

[36] Calculate frames per second (FPS) or processing time per image for each algorithm.

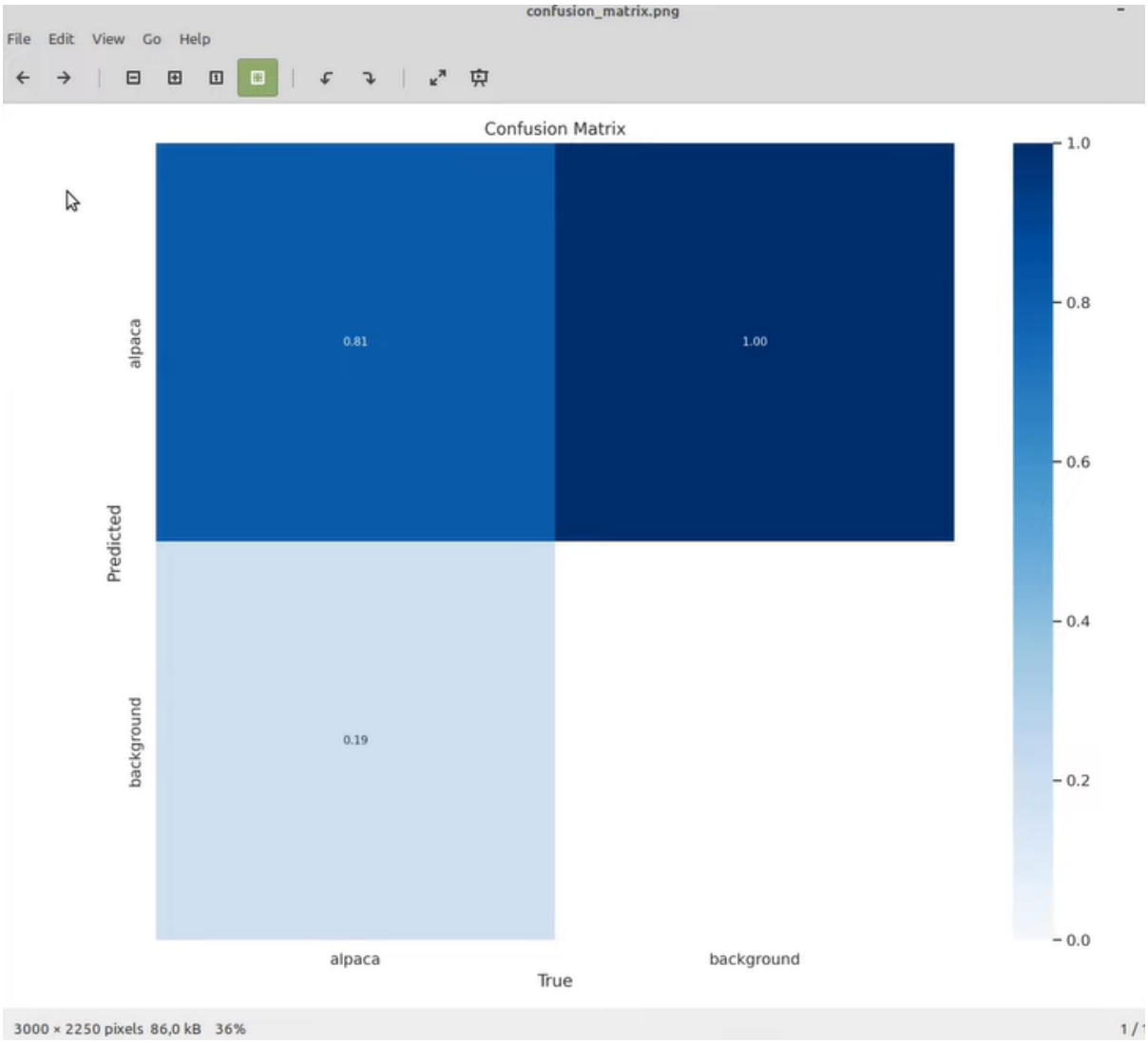


Figure 6.1: Confusion Matrix

The above image shows the confusion matrix of the tested value and the predicted value. We can see that, the value of true positive (TF) is 0.81 whereas the value of true negative (TN) is 1.00. We can also observe that the value of false positive (FP) is 0.19.

A confusion matrix is a tabular representation used in classification tasks to visualize the performance of a machine learning model. It shows the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by the model. The confusion matrix is particularly useful for understanding how well the model is performing in terms of correctly and incorrectly classified instances.

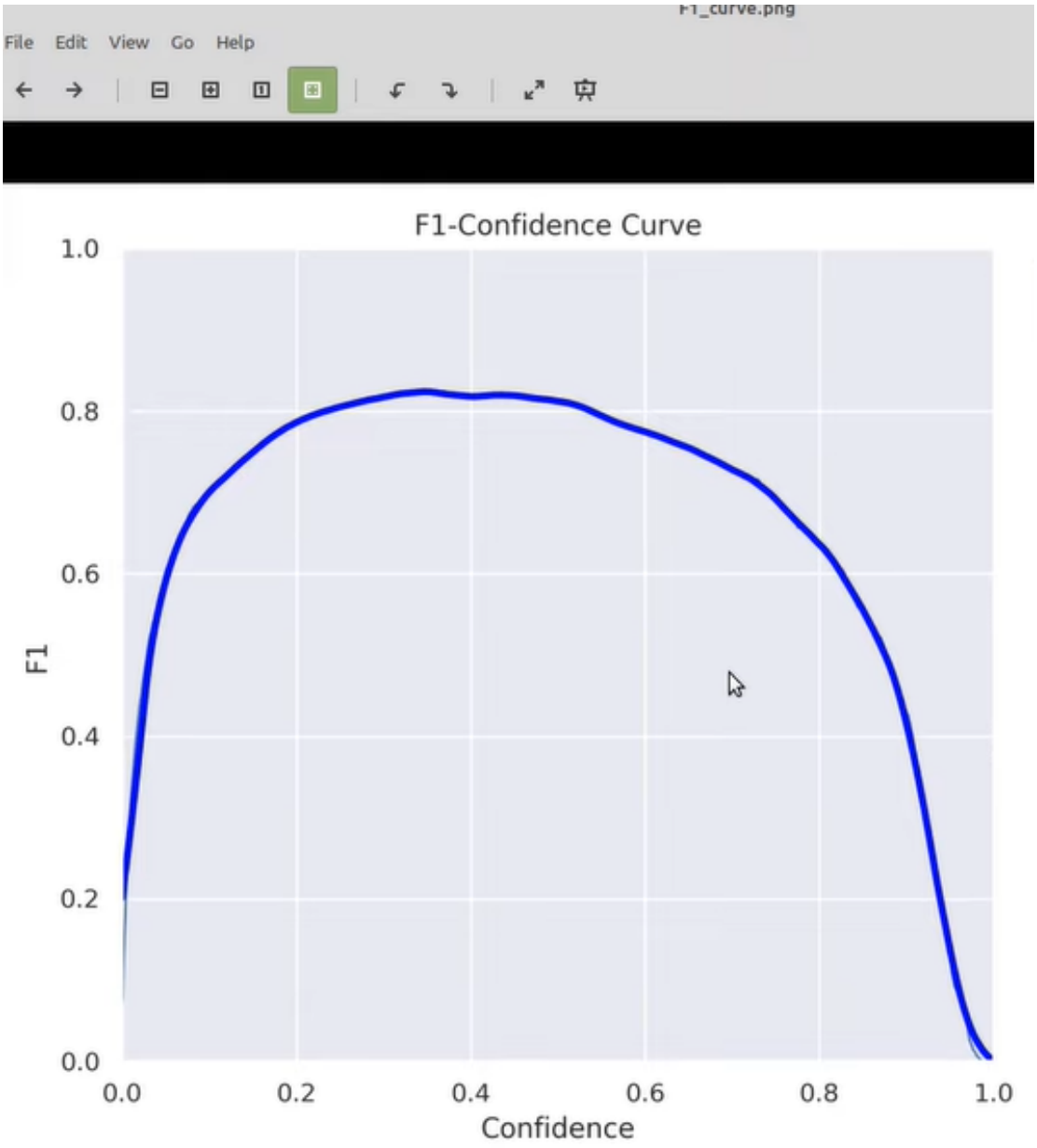


Figure 6.2: F1 Confidence Curve

This image shows us the F-1 score or F-1 confidence curve of the given tested sample and predicted sample.

It is a graphical representation that showcases how the F1 score of a machine learning model changes as you vary the confidence threshold for predictions. The F1 score is a metric that balances both precision and recall, providing a measure of a model's performance that considers both false positives and false negatives.

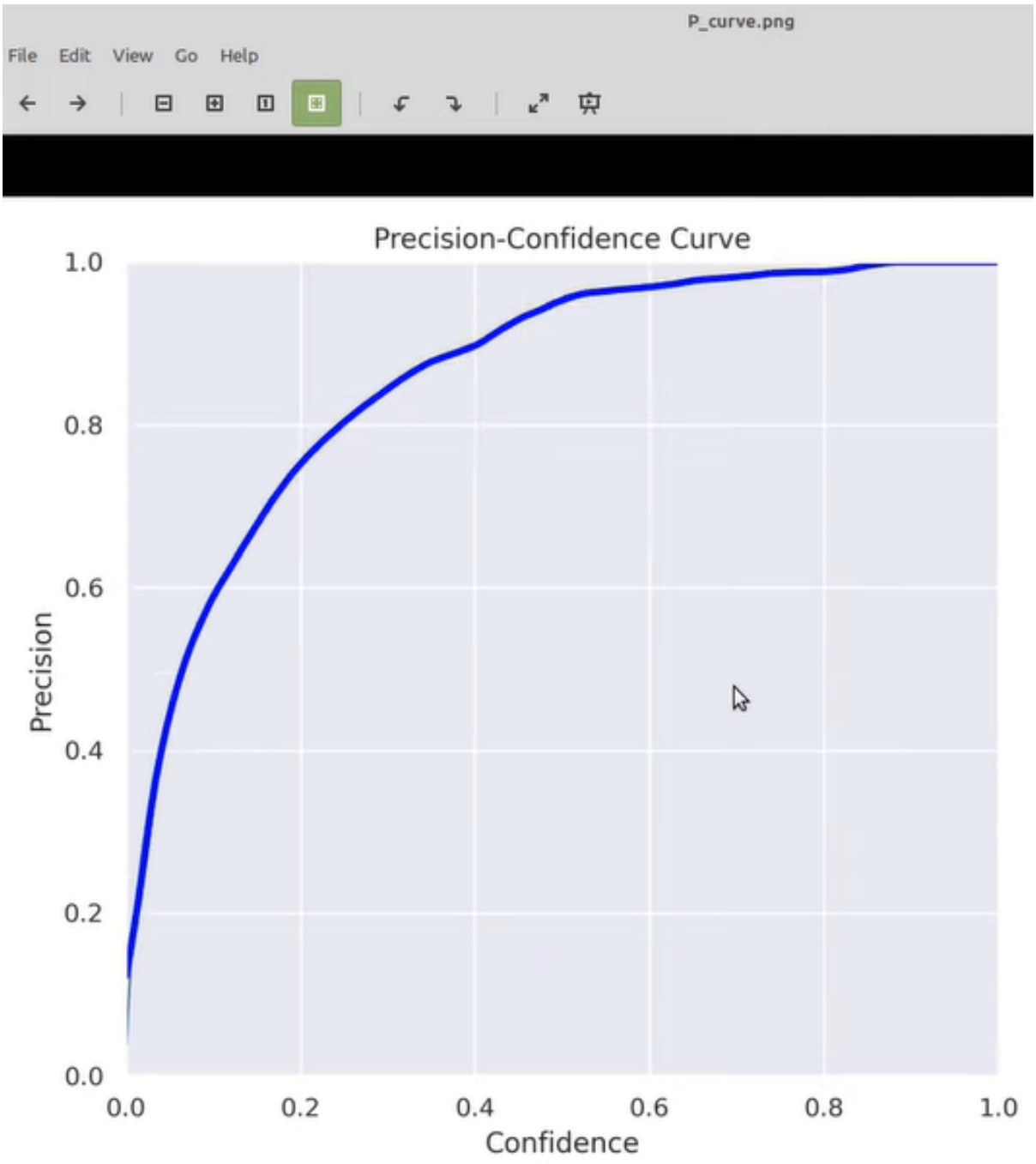


Figure 6.3: Precision-Confidence Curve

The above curve means the Precision-Confidence curve of the given sample.

Precision-Confidence curve is a graphical representation that illustrates how the precision of a machine learning model changes as we vary the confidence threshold for predictions. This curve is particularly useful in binary classification tasks, where we have two classes (positive and negative).

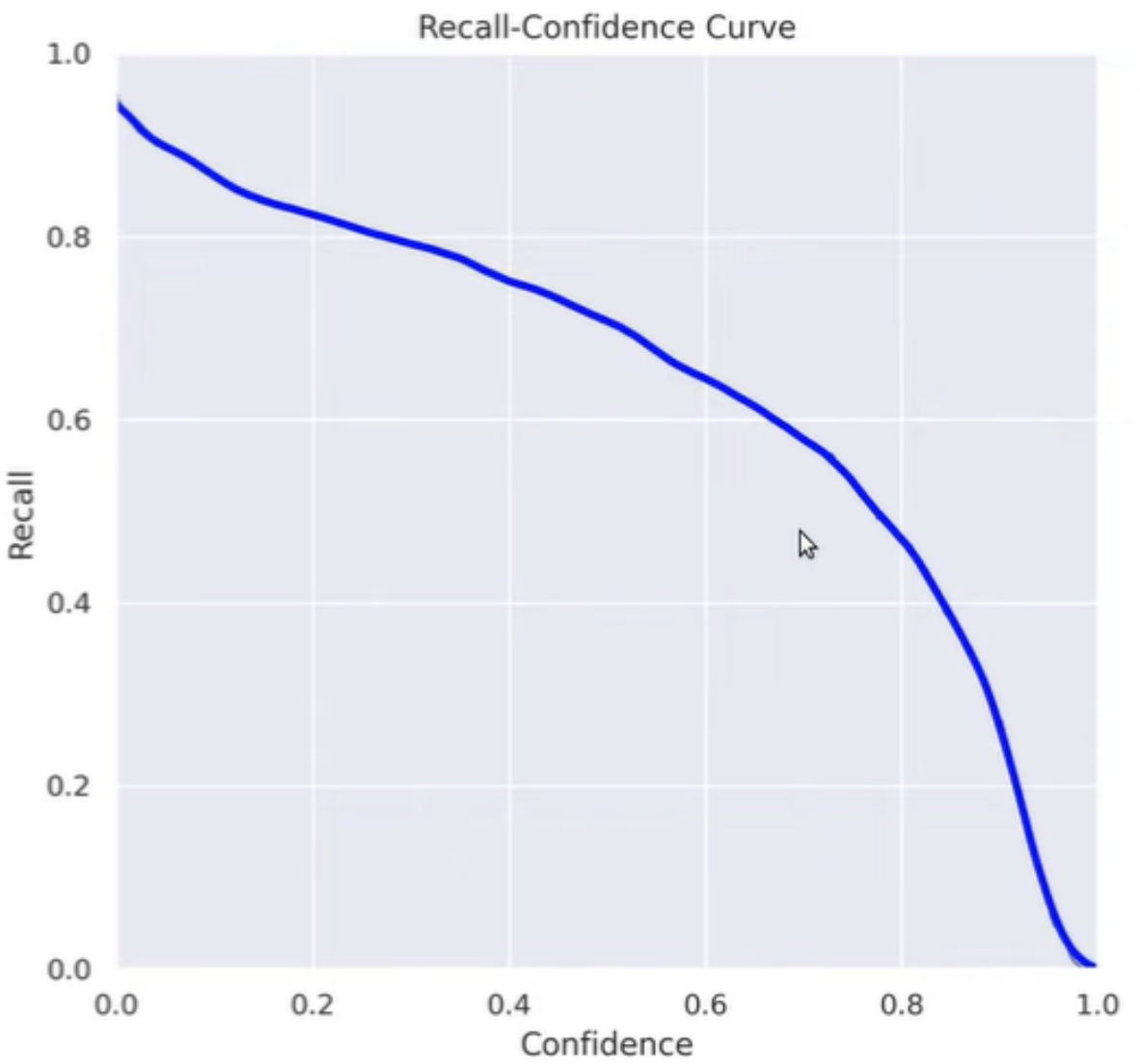


Figure 6.4: Recall-Confidence Curve

The above curve is called the Recall-Confidence curve of the tested sample dataset.

Recall-Confidence curve is a graphical representation that shows how the recall of a machine learning model changes as we vary the confidence threshold for predictions. This curve is particularly useful in binary classification tasks, where you have two classes (positive and negative), and we are interested in assessing how well the model can correctly identify positive instances (recall) at different levels of confidence.

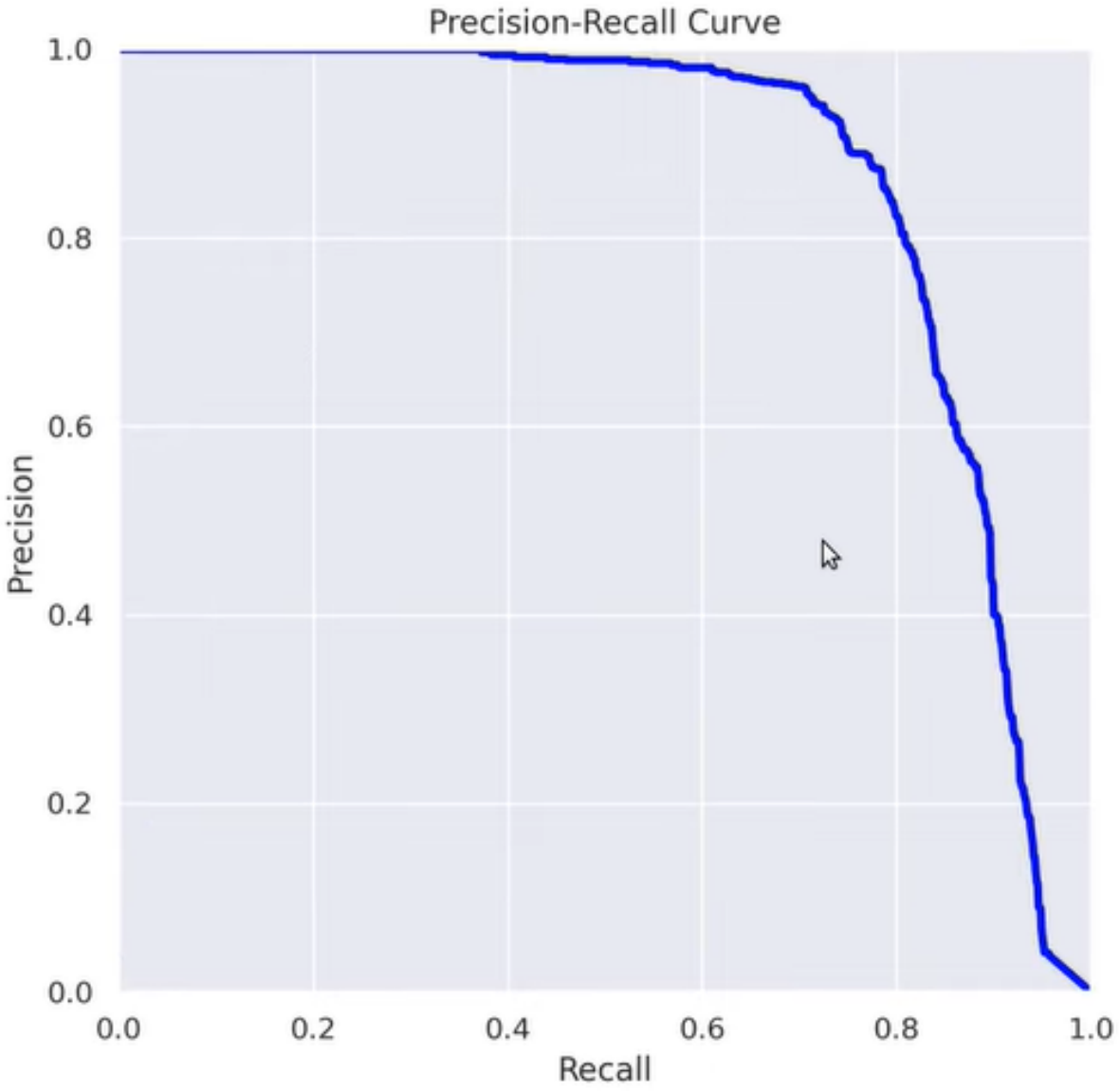


Figure 6.5: Precision-Recall Curve

The above image is the visual representation of the Precision-Recall curve of the tested sample dataset.

For binary classification, precision-recall curves plot the trade-off between precision and recall as you adjust the decision threshold of the model. These curves help you understand how well the model performs across different levels of precision and recall.

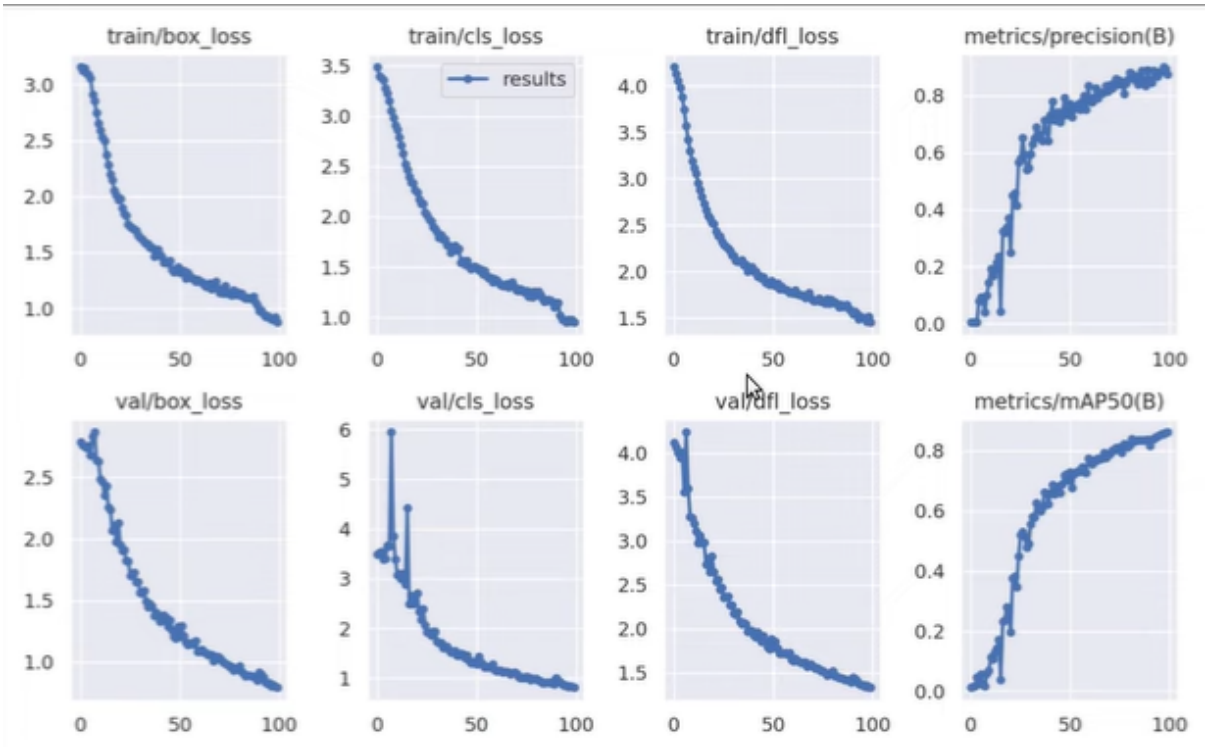


Figure 6.6: Curves with different functionality

In this figure, we can see the representation of various functions that are visualized by different curves.

In the context of machine learning and deep learning, training and validation curves are graphical representations of how certain metrics change over the course of training. These curves are used to monitor the performance of the model as it learns from the training data and to detect potential issues like overfitting or underfitting.

These curves are typically visualized using libraries like Matplotlib in Python. Keep in mind that the specific metrics and curves you use may depend on your problem type (regression, classification, etc.) and the evaluation metrics that are meaningful for your specific task.

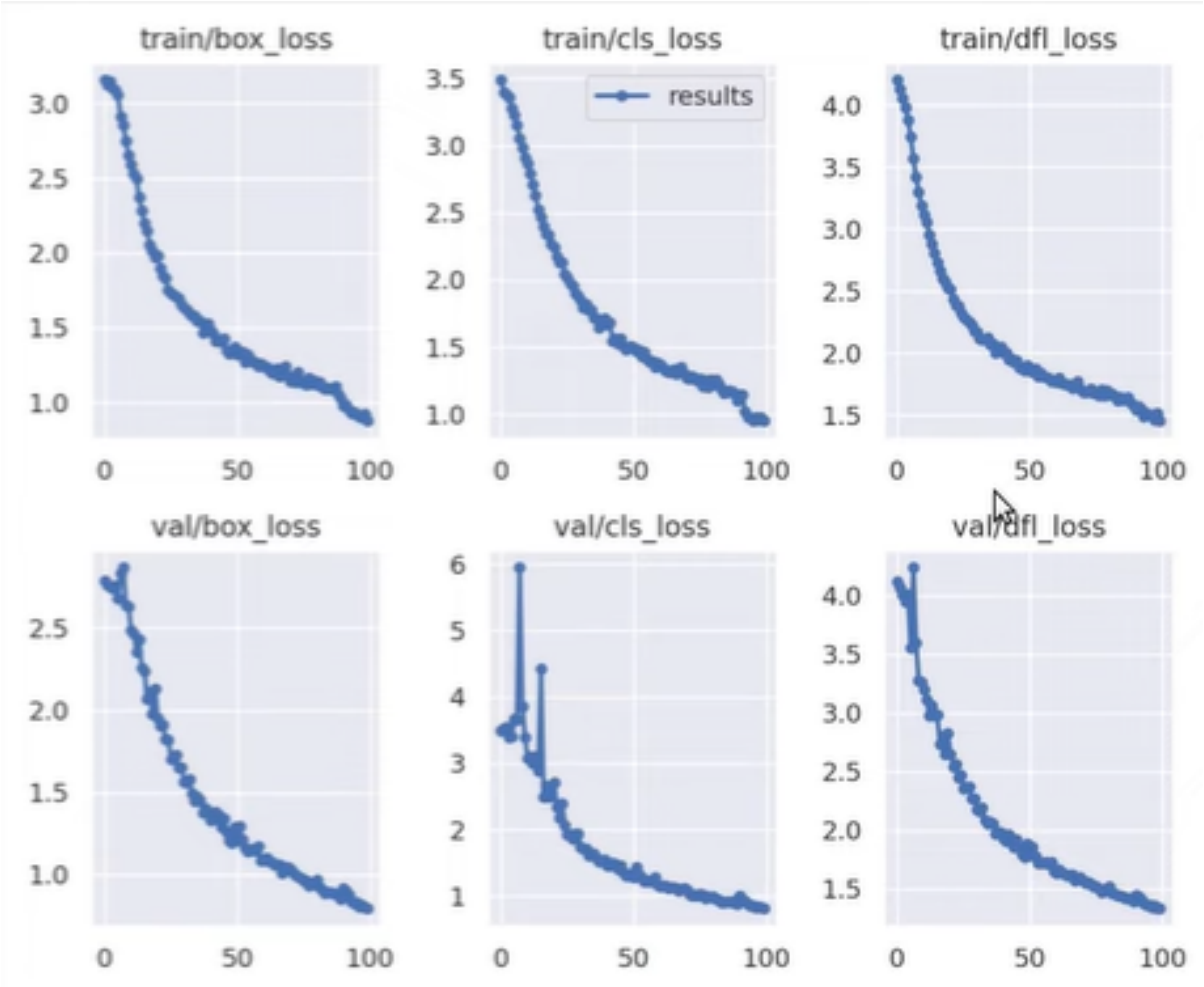


Figure 6.7: Loss Function Curves

Here, we can see various kinds of curves that is produced by the loss functions. These functions can determine the amount of losses or failures of a research or test. By looking at those curves, we can see that, the loss function curves are gradually decreasing. That means the accuracy of this prediction is very high and efficient. The loss function curve of trained dataset and validation dataset are reduced significantly.

Loss functions measure the discrepancy between predicted values and ground truth labels. Training and validation loss curves show how the loss changes during training. As the model learns, the loss should generally decrease. Common loss functions include mean squared error (MSE), categorical cross-entropy, and binary cross-entropy.

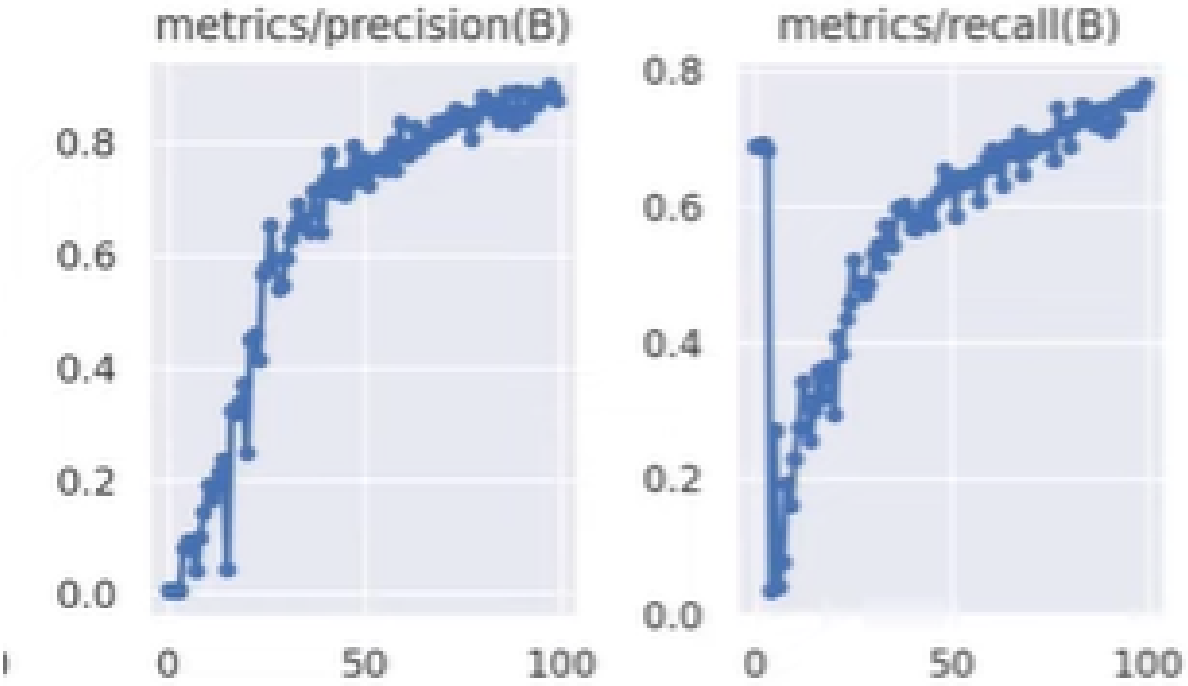


Figure 6.8: Curves of Metrics/Precision and Metrics/Recall

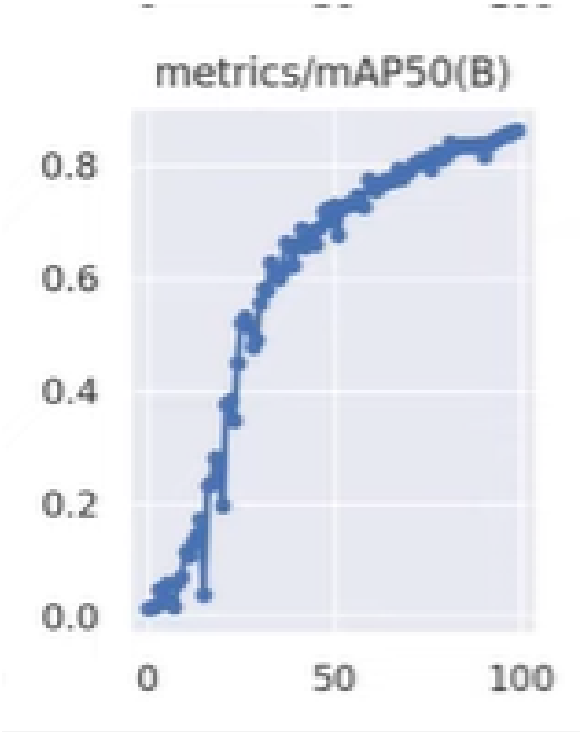


Figure 6.9: The curve of metrics/mAP50

The above figures are the representation of different metrics according to its precision, recall and mAP values of the tested and predicted data samples.

Table 6.1: Test Results based on Accuracy

	Detecting Faces in all angles	**Occlusion**	**Expression**
Viola-Jones	45%	27%	86%
YOLO-V3	90%	89%	98%
YOLO-V8	93%	94.5%	98.7%

6.3 Result Analysis

In this paper, a comparison was made between VJ and YOLO v8 algorithms for face detection in terms of speed and accuracy. Accuracy is considered one of the very important criteria as it depends on the rotation of the head at different angles, expressions and tilt. After performing the real-time video test, it was found that the ability of VJ algorithm to detect face is limited up to the angle of 30 by 98%, occlusion 27% and expressions 86%. The Python programming language was used to make the comparison.

It shows the results obtained by moving head in different directions. Among the 20 snapshots that were taken for the video, there are eleven shots in which the face was not detected; VJ algorithm achieved good results in detecting the front face, in addition to the movement of head in some different directions and angles up to the 30 angle. This means that it is weak at detecting faces at angles of 45, 60 ... etc. In addition to its inability to detect when the head is tilted. That is, the detection accuracy rate is 45% relative to the head movement (rotation).

When testing the video in real time using the Yolo v8 algorithm, the accuracy rate was about 93% in detecting the face and in all angles, occlusion 94.5% and expressions by 98.7% .It also achieved good results if the head was tilted left or right. It shows the results obtained by moving the head in different directions. Among the 20 snapshots that were taken for the video, there were two shots in which the face was not exposed, in which the movement of the head is up and down strongly.

6.4 Chapter Summary

Face detection and locating it in the video is one of the topics that has gained a great importance. VJ and Yolov8 algorithms are two successful algorithms for detecting faces; they have been used in many different applications. In this paper comparison between these algorithms is done, the comparison is focused on cases of head movement in many angles. Based on the results obtained in this paper, VJ algorithm obtained an accuracy rate of 45%, while the Yolov8 deep learning algorithm obtained an accuracy rate of 90%, these rates refer to the VJ algorithm that can achieve good results in detecting the front and close faces of the camera, as well as in expressions, but its results is low in the case of occlusion, while for the YOLO v3, it achieved good results in the case of head movement at different angles, as well as in expression and occlusion. Additionally from practical view, it has been found that the VJ is faster, but its accuracy rate is less, unlike YOLO v8, it is slower but has more accuracy rate. For the future work, and through this study, the results of testing the two algorithms showed that each algorithm has advantages and disadvantages in terms of speed and accuracy.

Therefore, it is possible to use Viola algorithm in the case of speed and also it is possible to use algorithm Yolov8 in the case of accuracy, or improve the algorithms, or combine them in order to obtain a fast and accurate algorithm to be suitable for work.

Chapter 7

Conclusion

In this chapter, we will discuss about the further approaches that can be taken for the development of this thesis work.

7.1 Discussion

Feature extraction is a huge concept in terms of modern machine learning algorithms.The maximum efficiency of feature selection using YOLO-V8 is 95.6 percent whereas in case of viola jones, the percentage is only 85. For the image extraction process in PCA, the accuracy was 89.3 percent.

7.2 Limitations

In case of PCA, the dimension of the resized image is critical. If it is too large, the system may break down and the analysis may not further continue. And on the other hand, if it is too small, then the images can not be discriminated from each other. Viola jones does not take time in training but having less accuracy than all the other methods.

7.3 Future Works

In future, we tend to do recognition of emotional facial expression and detecting the accurate human behaviour. We will also add the image segmentation process to increase robustness and efficiency. Furthermore, we have the intention to build a security system by synchronizing the detection models with Close Circuit Camera.

REFERENCES

[1] N. D. Tra, N. C. Tri, and P. D. Hung, "Improving warped planar object detection network for automatic license plate recognition," *arXiv preprint arXiv:2212.07066*, 2022.

[2] D. Garg, P. Goel, S. Pandya, A. Ganatra, and K. Kotecha, "A deep learning approach for face detection using yolo," in *2018 IEEE Punecon*, pp. 1–4, IEEE, 2018.

[3] N. Azmi, *A driver fatigue monitoring and haptic jacket-based warning system*. University of Ottawa (Canada), 2012.

[4] F. Bonilla-Rivas, C. Cárdenas-Angelat, A. Garrido-Díaz, and M. C. Aguayo-Torres, "Adapting yolo to recognition of real numbers in 7-segment digits," in *2023 International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS)*, pp. 19–25, IEEE, 2023.

[5] M. Son and K. Ko, "Multiple projector camera calibration by fiducial marker detection," *IEEE Access*, 2023.

[6] J. Jönsson Hyberg and A. Sjöberg, "Investigation regarding the performance of yolov8 in pedestrian detection," 2023.

[7] S. Abtahi, B. Hariri, and S. Shirmohammadi, "Driver drowsiness monitoring based on yawning detection," in *2011 IEEE International Instrumentation and Measurement Technology Conference*, pp. 1–4, IEEE, 2011.

[8] I. P. Sary, S. Andromeda, and E. U. Armin, "Performance comparison of yolov5 and yolov8 architectures in human detection using aerial images," *Ultima Computing: Jurnal Sistem Komputer*, vol. 15, no. 1, pp. 8–13, 2023.

[9] S. M. Silva and C. R. Jung, "License plate detection and recognition in unconstrained scenarios," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 580–596, 2018.

[10] A. C. Rios, A. R. Cukla, M. A. D. S. L. Quadros, and D. F. T. Gamarra, "Comparison of the yolov3 and ssd models using a balanced dataset with data augmentation, for object recognition in images," in *2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE)*, pp. 288–293, IEEE, 2022.

[11] F. Nowruzi, *Deep Learning for Autonomous and Driver Assistant Systems*. PhD thesis, Université d'Ottawa/University of Ottawa, 2020.

[12] J. S. Murthy, G. Siddesh, W.-C. Lai, B. Parameshachari, S. N. Patil, and K. Hemalatha, "Objectdetect: A real-time object detection framework for advanced driver assistant systems using yolov5," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[13] N. Taherifard, *AI-assisted Anomalous Event Detection for Connected Vehicles*. PhD thesis, Université d'Ottawa/University of Ottawa, 2021.

[14] T. FUJII, R. JINKI, and Y. Horita, "Practical improvement and performance evaluation of road damage detection model using machine learning," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, p. 2022IML0003, 2023.

[15] M. Hussain, "Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection," *Machines*, vol. 11, no. 7, p. 677, 2023.

[16] M. Hussain, "Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection," *Machines*, vol. 11, no. 7, p. 677, 2023.

[17] H. Lou, X. Duan, J. Guo, H. Liu, J. Gu, L. Bi, and H. Chen, "Dc-yolov8: Small-size object detection algorithm based on camera sensor," *Electronics*, vol. 12, no. 10, p. 2323, 2023.

[18] Y. Luo, S. Li, K. Sun, R. Renteria, and K. Choi, "Implementation of deep learning neural network for real-time object recognition in opencl framework," in *2017 International SoC Design Conference (ISOCC)*, pp. 298–299, IEEE, 2017.

[19] V. Teslyuk and B. Borkivskyi, "Models and means of object recognition using artificial neural networks," 2020.

[20] A. R. Kalva, J. S. Chelluboina, and B. Bharathi, “Smart traffic monitoring system using yolo and deep learning techniques,” in *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 831–837, IEEE, 2023.

[21] M. Chetoui and M. A. Akhloufi, “Object detection model-based quality inspection using a deep cnn,” in *Sixteenth International Conference on Quality Control by Artificial Vision*, vol. 12749, pp. 65–72, SPIE, 2023.

[22] J. Terven and D. Cordova-Esparza, “A comprehensive review of yolo: From yolov1 to yolov8 and beyond,” *arXiv preprint arXiv:2304.00501*, 2023.

[23] N. Affes, J. Ktari, N. Ben Amor, T. Frikha, and H. Hamam, “Real time detection and tracking in multi speakers video conferencing,” in *International Conference on Intelligent Systems Design and Applications*, pp. 108–118, Springer, 2022.

[24] M. S. Beg, M. Y. Ismail, M. S. U. Miah, and M. H. Peeie, “Yus-a deep learning algorithm for collision avoidance through object and vehicle detection,” *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 31, no. 1, pp. 226–236, 2023.

[25] S. Shafi, T. P. S. K. Reddy, R. Silla, and M. Yasmeen, “Deep learning based real-time stolen vehicle detection model with improved precision and reduced look up time,” in *2023 3rd International Conference on Intelligent Technologies (CONIT)*, pp. 1–6, IEEE, 2023.

[26] Y. Chang, “Video-based event detection in catheterization laboratory,” 2023.

[27] A. Dodia and S. Kumar, “A comparison of yolo based vehicle detection algorithms,” in *2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1)*, pp. 1–6, IEEE, 2023.

[28] H. H. Nguyen, T. N. Ta, N. C. Nguyen, H. M. Pham, D. M. Nguyen, *et al.*, “Yolo based real-time human detection for smart video surveillance at the edge,” in *2020 IEEE Eighth International Conference on Communications and Electronics (ICCE)*, pp. 439–444, IEEE, 2021.

[29] T. Mikkonen, “Capacity monitoring using object detection algorithms,” 2021.

[30] L. Fekkar *et al.*, *Face recognition based on deep learning techniques*. PhD thesis, University of M'sila, 2023.

[31] A. Melino-Carrero, Á. N. Suárez, C. Losada-Gutierrez, M. Marron-Romera, I. G. Luna, and J. Baeza-Mas, "Object detection for functional assessment applications," in *International Conference on Engineering Applications of Neural Networks*, pp. 328–339, Springer, 2023.

[32] J. Zhang, S. Qian, and C. Tan, "Automated bridge surface crack detection and segmentation using computer vision-based deep learning model," *Engineering Applications of Artificial Intelligence*, vol. 115, p. 105225, 2022.

[33] T.-H. Jung, B. Cates, I.-K. Choi, S.-H. Lee, and J.-M. Choi, "Multi-camera-based person recognition system for autonomous tractors," *Designs*, vol. 4, no. 4, p. 54, 2020.

[34] Y. Li, Q. Fan, H. Huang, Z. Han, and Q. Gu, "A modified yolov8 detection network for uav aerial image recognition," *Drones*, vol. 7, no. 5, p. 304, 2023.

[35] P. Li, "Research on rgb-d slam dynamic environment algorithm based on yolov8," in *2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, vol. 3, pp. 1038–1044, IEEE, 2023.

[36] H. Du, *General Object Detection Algorithm Yolov5 Comparison and Improvement*. PhD thesis, CALIFORNIA STATE UNIVERSITY, NORTHRIDGE, 2023.