**1 INTRODUCTION**

The purpose of this project is to implement and optimise the YOLO algorithm for real-time object detection, focusing on accuracy, speed, and compatibility with various deployment scenarios. The project aims to build a comprehensive system that can detect and classify objects across different categories with high accuracy and minimal latency. The project will also emphasise documentation and knowledge sharing, capturing the implementation details, experimental results, and instructions for utilising the YOLO-based object detection system. The documentation will serve as a valuable resource for researchers, developers, and practitioners interested in understanding and building upon the project's work. Overall, this project aims to leverage the power of the YOLO algorithm to develop an accurate, efficient, and real-time object detection system. By successfully implementing and optimising the YOLO-based approach, the project intends to contribute to the advancement of computer vision technology and facilitate the adoption of reliable object detection systems in various real-world applications.

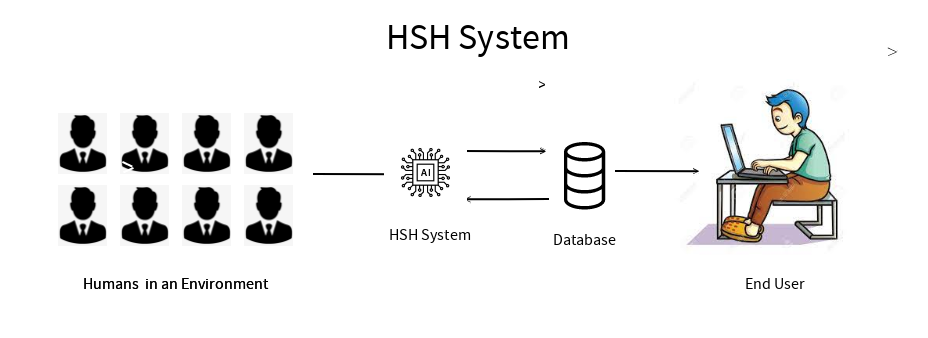
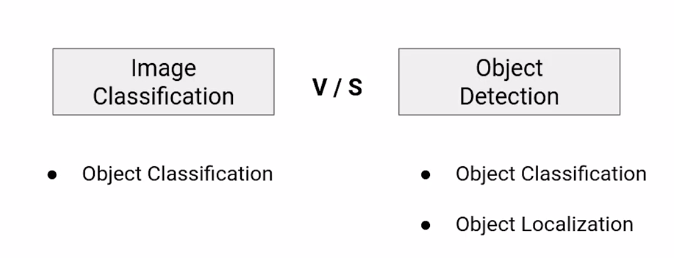


Fig 1.a. HSH System Overview

**1.1 BASICS CONCEPT OF FACE DETECTION**

Object detection is one of the most important research directions for computer vision.

Fig 1.b. Image Classification V/S Object Detection

Humans can easily detect and identify objects present in an image. The human visual system is fast and accurate and can perform complex tasks like identifying multiple objects and detecting obstacles with little conscious thought. With the availability of large amounts of data, faster GPUs, and better algorithms, we can now easily train computers to detect and classify multiple objects within an image with high accuracy.

With this kind of identification and localization, object detection can be used to count objects in a scene and determine and track their precise locations, all while accurately labelling them.

**Region Proposal Object Detection Algorithms :**

A Region Proposal Network, or RPN, is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals. RPN and algorithms like MASK -RCNN can be merged into a single network by sharing their convolutional features - using the recently popular terminology of neural networks with attention mechanisms, the RPN component tells the unified network where to look.RPNs are designed to efficiently predict region proposals with a wide range of scales and aspect ratios. RPNs use anchor boxes that serve as references at multiple scales and aspect ratios. The scheme can be thought of as a pyramid of regression references, which avoids enumerating images or filters of multiple scales or aspect ratios.

**Regression Object Detection Algorithm**

Regression object detection is a popular technique to refine or predict localization boxes in recent object detection approaches. Typically, bounding-box regressors are trained to regress from either region proposals or fixed anchor boxes to nearby bounding boxes of predefined target object classes. This paper investigates whether the technique is generalizable to unseen classes and is transferable to other tasks beyond supervised object detection. To this end, we propose a class-agnostic and anchor-free box regressor, dubbed Universal Bounding-Box Regressor (UBBR), which predicts a bounding box of the nearest object from any given box. Trained on a relatively small set of annotated images, UBBR successfully generalises to unseen classes, and can be used to improve localization in many vision problems. We demonstrate its effectiveness on weakly supervised object detection and object discovery.

**1.2 USES:**

* Contactless­ payment­

Nowadays, technology is being created to turn physical purses into digital wallets. That means, you no longer need to worry about carrying your bank cards everywhere you go.With the use of object detection and cameras, consumers can simply pick items off the shelve and pay for them using their phones

Eg. Amazon Go Store

### Healthcare Medical Detection

In addition to video surveillance, healthcare organisations can also benefit from object detection. As many medical diagnostics require images to study, scans for patients and other photographs this technology is a breakthrough.

Instead of waiting for results to be processed, you can have instant access to data and images that will enable you to give your patients better care. As well as this, it can help improve medicine and treatment procedures.

**PRINCIPLE:**

Object recognition involves matching representations of **objects stored in memory to representations extracted from the visual image**. The key issue in object recognition is the nature of the representation extracted from the image. Theories of object recognition are characterised in terms of five logically independent dimensions: the primitive features or parts extracted from the visual image, the stability of the set of features to transformations of the image, the type of relationships used to describe configurations of features, and the stability of configurations across transformations of the image.

**1.3 EVOLUTION OF OBJECT DETECTION**

The task of detecting and recognizing an unknown number of individual objects within an image,called object detection ,which was considered an extremely difficult problem only a few years ago, is now feasible and has even been productized by companies like [**Google**](https://cloud.google.com/vision/docs/drag-and-drop) and [**IBM**](https://www.ibm.com/watson/services/visual-recognition/) **(International Business Machines Corporation.)**

**1.3.1 Viola Jones Detectors:**

Developed in 2001 by Paul Viola and Michael Jones,this object recognition framework allows the detection of human faces in real-time.It uses sliding windows to go through all possible locations and scales in an image to see if any window contains a human face.The sliding windows essentially searches for ‘haar-like’ features (named after Alfred Haar who developed the concept of haar wavelets). **Haar-Like Features.**

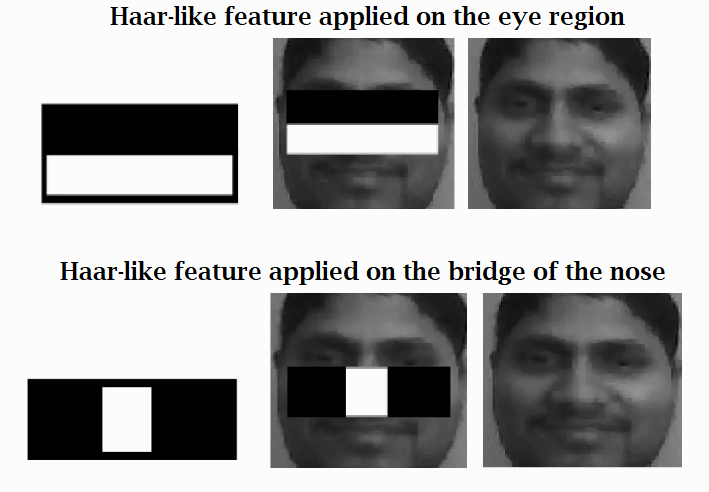


Fig .2. Viola Jones Algorithm Visualisation

Thus the haar wavelet is used as the feature representation of an image.To speed up detection, it uses *integral image* , which makes the computational complexity of each sliding window independent of its window size.Another trick to improve detection speed that was used by the authors is to use Adaboost algorithm for *feature selection* which selects a small set of features that are mostly helpful for face detection from a huge set of random features pools.The algorithm also used *Detection Cascades which is a* multi-stage detection paradigm to reduce its computational overhead by spending less computations on background windows but more on face targets.

**1.3.2 HOG Detector :**

Originally proposed in 2005 by N. Dalal and B. Triggs,Hog is an improvement of the scale invariant feature transform and shape contexts of its time.HOG works with something called *blocks*(similar to a sliding window) ,a dense pixel grid in which gradients are constituted from the magnitude and direction of

Fig.3.HOG Detection Pattern

change in the intensities of pixels within the block. HOGs are widely known for their use in pedestrian detection. To detect objects of different sizes, the HOG detector rescales the input image for multiple times while keeping the size of a detection window unchanged.

**1.3.3 GROWTH OF OBJECT DETECTION :**

Computer vision technology enters the phase of expeditious growth . Many companies and organisations provide opportunities for machine learning and object detection . Object detection is involved in our day-to- day life .

For instance: AI cameras to check traffic violation that installed in mostly of metropolitan cities



Fig.4. AI Registration Plate Detector

COMPLEX ENVIRONMENT:

Here we discuss about some complex environment and the recent issues

related to human surveillance history.

Examples of complex environments are

* On using AI-based human identification in improving surveillance system efficiency(2019)

**On using AI-based human identification in improving surveillance system efficiency(2019):**

Surveillance camera systems are one of the most popular tech tools keeping homes and businesses safe. Many of these systems record continuously while some others start recording when a movement is detected. These recording methods consume a huge amount of storage, and they need a dramatic amount of time to search in the recorded files. Here, an AI-based desktop application which is designed and developed in order to start recording only if a person or a human-face is identified. This technique will improve the system's efficiency in terms of reducing the storage needed for saving recordings, and reducing the processing and searching time in the recordings. This proposed system uses deep-learning algorithms, OpenCV libraries and is built on Linux OS. It has two detection modes; human body identification and human-face identification. This will trigger the suitable detection algorithms based on mode selected. The system has been tested and evaluated by different evaluators and organisations that use surveillance systems, and questionnaires and interviews were conducted. Based on the evaluation results, it can be deduced how much this system is an important application to the target group

With the increasing millions of hours of videos recorded everyday from the surveillance systems, there is an urgent need to optimise the recording process in order to save storage and improve the efficiency of the surveillance systems. Here, an AI-based system is proposed to trigger a camera to start recording only if either human-face or human-body is detected. This system is tested by developing a desktop-application that implements the proposed system’s algorithms. The application mentioned has been developed using Python, Open source Computer Vision (OpenCV) libraries, and a pre-trained Haar cascade algorithm for human and/or face detection

**2 LITERATURE SURVEY**

**2.1 Comparative design, scaling and control:**

Object detection is a technology that detects the semantic Objects of a category in virtual snapshots and films. Certainly, one of its Actual-time packages is self-riding automobiles. In this, our challenge is to locate a couple of items from a photo. The maximum common Item to come across on this utility is the car, motorcycle, and Pedestrian. For locating the gadgets within the photograph, Object localization should find multiple items in real-time structures. There are various techniques for item Detection, they can be broken up into classes, first is the Algorithms primarily based on classifications.CNN and RNN come below this category. On this, pick out the involved Regions from the picture and ought to classify them using the use of Convolutional neural community. This technique may be very sluggish Due to the fact it should run a prediction for every decision on Vicinity. The second one class is the algorithms primarily based on Regressions. Yolo approach comes below this category. In This, it might not choose the fascinated regions from the photograph. Instead, it expects the training and bounding containers of the complete picture at a single run of the algorithm and detects a couple of gadgets using an unmarried neural community. The Yolo Set of rules is rapid as compared to other classification Algorithms. In actual time our algorithm technique 45 frames consistent with 2d. The Yolo algorithm makes localization mistakes however Predicts less false positives in the background[4][2].

The biggest advantage of using YOLO is its superb speed – it's incredibly fast and can process 45 frames per second. YOLO also understands generalised object representation[4][1].

The Average Precision (AP), traditionally called Mean Average Precision (mAP), is the commonly used metric for

evaluating the performance of object detection models. It measures the average precision across all categories, providing a single value to compare different models. The COCO dataset makes no distinction between AP and AP. In the rest of this paper, we will refer to this metric as AP. In YOLOv1 and YOLOv2, the dataset utilised for training and benchmarking was PASCAL VOC 2007, and VOC 2012 [1]. However, from YOLOv3 onwards, the dataset used is Microsoft COCO (Common Objects in Context) [1]. The AP is calculated differently for these datasets. The following sections will discuss the rationale behind AP and explain how it is computed.

**2.2 The History of Face Recognition:**

Currently state-of-the-art real-time object detectors are mainly based on YOLO and FCOS , which are Being able to become a state-of-the-art real-time object detector usually requires the following characteristics: (1) a faster and stronger network architecture; (2) a more effective feature integration method (3) a more accurate detection method (4) a more robust loss function(5) a more efficient label assignment method and (6) a more efficient training method. In this paper, we do not intend to explore self-supervised learning or knowledge distillation methods that require additional data or large models. Instead, we will design a new trainable bag-of-freebies method for the issues derived from the state-of-the-art methods associated with (4), (5), and (6) mentioned above.

Kohonen's system was not a practical success, however, because of the need for precise alignment and normalisation. In following years many researchers tried face recognition schemes based on edges, inter-feature distances, and other neural net approaches. While several were successful on small databases of aligned images, none successfully addressed the more realistic problem of large databases where the location and scale of the face is unknown.

Kirby and Sirovich (1989) later introduced an algebraic manipulation which made it easy to directly calculate the eigenfaces, and showed that fewer than 100 were required to accurately code carefully aligned and normalised face images. Then demonstrated that the residual error when coding using the eigenfaces could be used both to detect faces in cluttered natural imagery, and to determine the precise location and scale of faces in an image[6][2]. They then demonstrated that by coupling this method for detecting and localising faces with the eigenface recognition method, one could achieve reliable, real-time recognition of faces in a minimally constrained environment. This demonstration that simple, real-time pattern recognition techniques could be combined to create a useful system sparked an explosion of interest in the topic of face recognition.

**2.3 Image-based Face Detection approach:**

Multiple face detection techniques have been introduced:

.

The beginnings. Face detection has been a challenging research field since its emergence in the 1990s. Before 2000, despite many studies, the practical performance of facial recognition was far from satisfactory until the milestone work proposed by Viola and Jones. Starting from the pioneering work of Viola-Jones (Viola and Jones 2004), face detection has made great progress. Viola and Jones pioneered the use of Haar features and AdaBoost to train a face detector with promising accuracy and efficiency (Viola and Jones 2004)[1][2], which inspired several different approaches afterward. However, it has several critical drawbacks. First of all, its feature size was relatively large. Also, it is not able to effectively handle non-frontal faces and faces in the wild. Early stages – machine learning: Early approaches mainly focused on extracting different types of hand-crafted features with domain experts in computer vision and training effective classifiers for detection with traditional machine learning algorithms. Such methods are limited in that they often require computer vision experts to craft effective features, and each individual component is optimised separately, making the whole detection pipeline often suboptimal. To address the first problem, much effort has been devoted to coming up with more complicated features like HOG (histograms of oriented gradients), SIFT (Scale Invariant Feature Transform), SURF (speeded up robust features), and ACF (aggregate channel features)[4][6]. To enhance the robustness of detection, a combination of multiple detectors that had been trained separately for different views or poses has been developed. Nevertheless, training and testing of such models were usually more time-consuming, and the boost in detection performance was relatively limited. State of the art – deep learning: Recent years have shown significant advances in facial recognition using deep learning methods, especially deep convolutional neural networks (CNN), have achieved remarkable successes in various computer vision tasks. In contrast to traditional computer vision approaches, deep learning methods avoid the hand-crafted design pipeline and have dominated many well-known benchmark evaluations, such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC)[2][3]. Recently, researchers applied the Faster R-CNN, one of the state-of-the-art generic object detectors, and achieved promising results. In addition, joint training conducted on CNN cascade, region proposal network (RPN), and Faster R-CNN has realised end-to-end optimization. Faster R-CNN face detection algorithm with hard negative mining and ResNet and achieved significant boosts in detection performance on face detection benchmarks like FDDB.

**2.4 DIFFICULTIES IN FACE DETECTION**

Challenges in face detection are the reasons which reduce the accuracy and detection rate of facial recognition. These challenges are complex backgrounds, too many faces in images, odd expressions, illuminations, less resolution, face occlusion, skin colour, distance, orientation, etc[6][2].

**Unusual expression**: Human faces in an image may show unexpected or odd facial expressions[1][6].

**Illuminations**: Some image parts may have very high or low illumination or shadows[1,6,2].

**Skin types**: Detecting faces of different face colours is challenging for detection and requires a wider diversity of training images.

**Orientation**: The face orientation and angle toward the camera impact the rate of face detection.

**Complex background**: A high number of objects in a scene reduces the accuracy and rate of detection.

**Many faces in one image**: An image with a high number of human faces is very challenging for an accurate detection rate.

**Face occlusion**: Faces may be partially hidden by objects such as glasses, scarves, hands, hairs, hats, and other objects, which impacts the detection rate.

**2.5 Use Cases of Face Detection Applications:**

**Crowd surveillance**: Face detection is used to detect and analyse crowds in frequented public or private areas. Use cases include crowd estimation and real-time alerting[4,1,3].

**Human-computer interaction (HCI)**: Multiple human-computer interaction-based systems use facial recognition to detect the presence of people in specific areas.

**Photography**: Some recent digital cameras use face detection for autofocus. Mobile apps use facial recognition to detect regions of interest in slideshows.

**Facial feature extraction**: Specific facial features such as the nose, eyes, mouth, skin colour and more can be extracted from images and live video feeds.

**Gender classification**: Applications are built to recognize gender information with face-detection methods. Such technologies are used for visitor and customer analysis[1,6].

**Face recognition**: A face recognition system is designed to identify and verify a person from a digital image or video frame, often as part of access control or identify verification solutions[1,6,2,5,3].

**Marketing**: Face detection is becoming more and more important for marketing, analysing customer behaviour, or segment-targeted advertising.

**Attendance**: Facial recognition is used to detect the attendance of individuals. It is often combined with biometric detection for access management.

**3 OBJECTIVES AND SCOPES OF WORK**

The scope of this project is to track the history of a person with coloured CCTV Footage.

* This project can be implemented where there is a need for searching the occurrence of a person in an organisation or in an predefined boundary which is under surveillance.
* It will reduce the time to track a person which will be very much important in security surveillance.

**4 EXISTING SYSTEM**

AI video analysis is used for anomaly detection in traffic, subway, campus, trains, boats, buildings, and public places. Examples of CCTV monitoring for anomaly detection in visual AI include stopped vehicle detection, panic detection, intrusion detection, or abnormal pedestrian activity recognition.

But there is no Human Surveillance System which does show the human detail who is captured while in the region of a surveillance camera , So easily we can identify him.

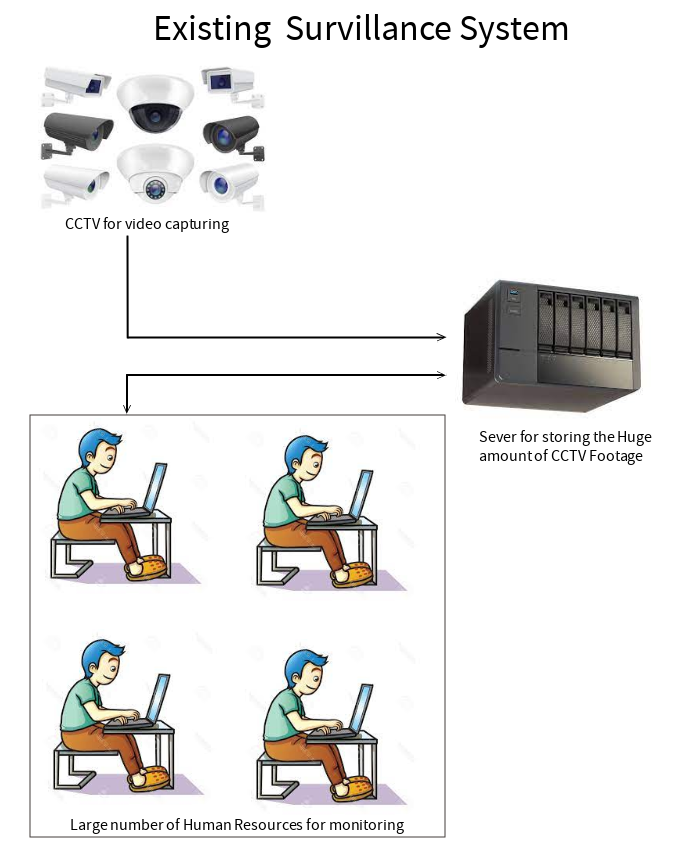


Fig.5 Existing System

**5 PROPOSED SYSTEM**

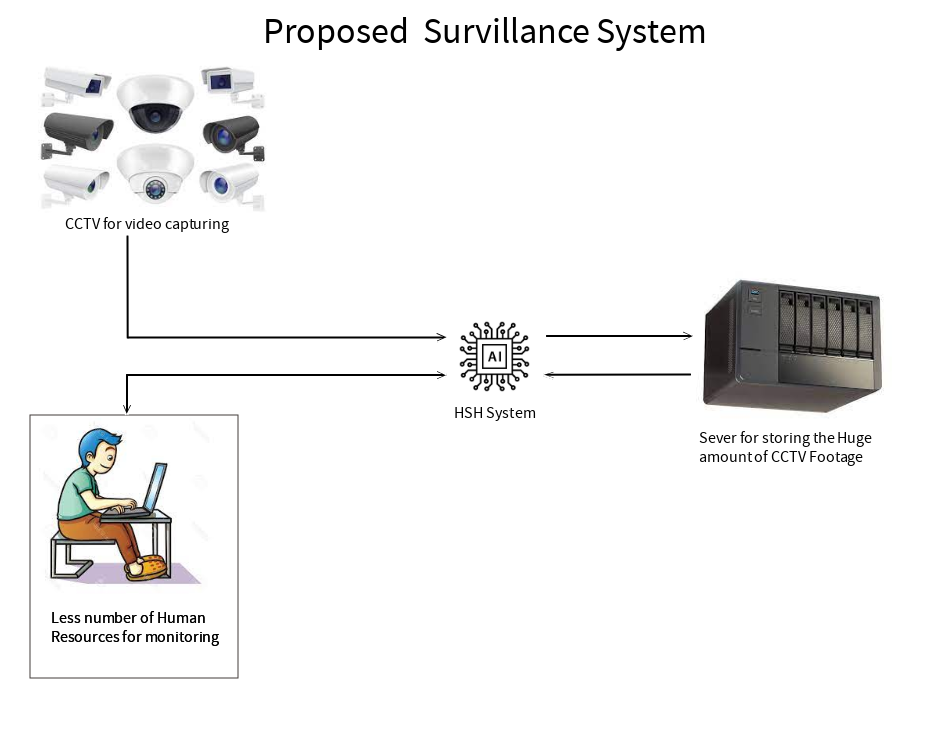
The proposed system has the main concept of gatheringthe information from pre-recorded footage.

Fig.6. Proposed System

Normally, Algorithm detects the object based on classification like humans , Dogs , and cats . but in this algorithm categories are based on human names which are pre-labeled by the developer .

**DISADVANTAGE OF EXISTING SYSTEM** :

1. Existing System never classifies human , It justs detects the human
2. There are a lot of workers required .
3. It requires more time to identify the person in manually

**5.1 Gathering information from Pre-recorded footage:**

**Annotated Faces in the Wild Dataset (AFW)**: The AFW dataset is built using Flickr images. It includes 205 images with 473 labelled faces. For each face, image annotations include a rectangular bounding box, 6 landmarks, and the pose angles.

**PASCAL Face Dataset (PASCAL FACE)**: This dataset is used for facial recognition and face recognition; it is a subset of the PASCAL VOC and contains 1‘335 labelled faces in 851 images with large face appearance and pose variations.

**MIT Face Dataset (CBCL Face Database)**: The MIT-CBCL face recognition database contains a training set (2’429 faces, 4’548 non-faces) and a test set (472 faces, 23’573 non-faces).

**Face Detection Data Set and Benchmark (FDDB)**: The dataset contains 5‘171 faces annotated in 2‘845 images with a wide range of difficulties, such as occlusions, difficult poses, and low image resolutions. These images are used to train with large appearance changes, heavy occlusions, and severe blur degradations that are prevalent in detecting a face in unconstrained real-life scenarios.

**CMU Multi-PIE Database (PIE)**: The CMU Multi-PIE Face Database contains 41’368 images of 68 people, each person under 13 different poses, 43 different illumination conditions, and 4 different expressions.

**Surveillance Cameras Face Database (SCface Dataset)**: SCface is a database of static images of human faces. The images were taken in an uncontrolled indoor environment using five video surveillance cameras of various qualities. The dataset contains 4’160 static images (visible and infrared spectrum) of 130 subjects.

**WIDER FACE dataset (WIDER)**: The face detection benchmark dataset includes 32’203 images and 393’703 labelled faces with a high degree of variability in scale, pose, and occlusion, making face detection extremely challenging. Also, the WIDER FACE dataset is organised based on 61 event classes.

**5.2 Overview of Application:**

5.2.1 Collecting the image data:

Collecting classification images is usually done manually using a photo editing software to crop and resize photos. Furthermore, PCA and LDA require the same number of pixels in all the images for the correct operation. This time consuming and a laborious task is automated through an application to collect 50 images with different expressions. The application detects suitable expressions between 300ms, straightens any existing tilt and saves them. The Flow chart for the application is shown in figure 5



Fig 7.1 Trained Mask Boundary Images

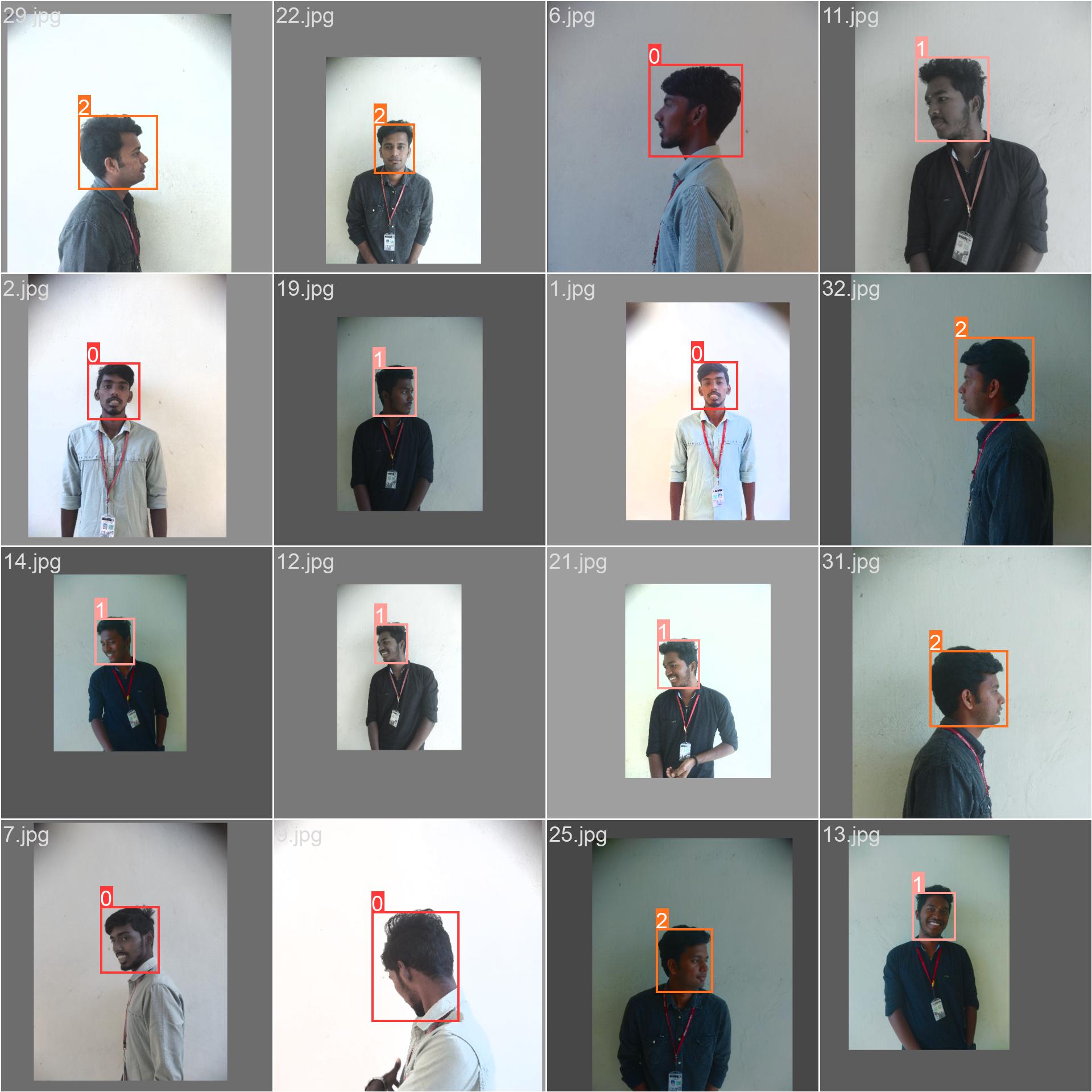


Fig 7.2 Trained Box Boundary Images

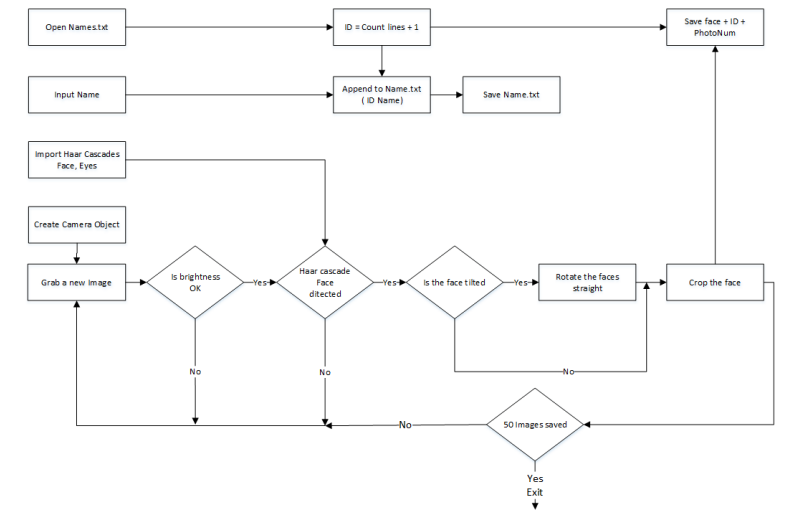


Figure 8: The Flowchart for the image collection

Application starts with a request for a name to be entered to be stored with the ID in a text file. The face detection system starts the first half. However, before the capturing begins, the application checks for the brightness levels and will capture only if the face is well illuminated. Furthermore, after the face is detected, the position of the eyes are analysed. If the head is tilted, the application automatically corrects the orientation. These two additions were made considering the requirements for the Eigenface algorithm. The Image is then cropped and saved using the ID as a filename to be identified later. A loop runs this program until 50 viable images are collected from the person. This application made data collection efficient.

5.2.2 The face recognition

Face recogniser object is created using the desired parameters. Face detector is used to detect faces in the image, cropped and transferred to be recognised. This is done using the same technique used for the image capture application. For each face detected, a prediction is made using FaceRecognizer.predict() which returns the ID of the class and confidence. The process is the same for all algorithms and if the confidence is higher than the set threshold, ID is -1. Finally, names from the text file with IDs are used to display the name and confidence on the screen. If the ID is -1, the application will print an unknown face without the confidence level.



Fig 7.3 Tested Images 01



Fig .7.4 . Tested Images 02

**5.3 Architecture of YOLO:**

The YOLO algorithm is designed for real-time object detection by simultaneously predicting the class labels and bounding boxes of multiple objects within an image. Unlike traditional object detection methods that rely on region proposal algorithms, YOLO takes a different approach by treating object detection as a regression problem.

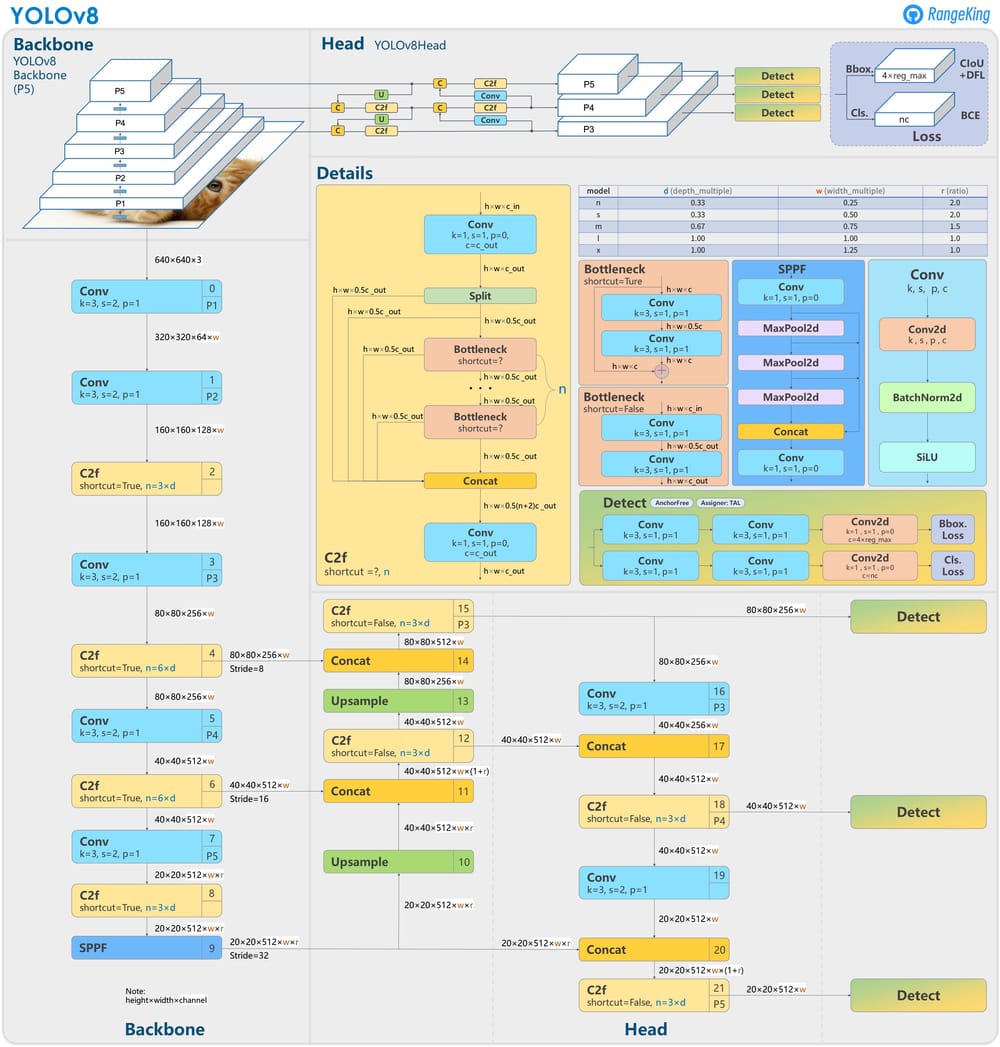
The YOLO architecture consists of the following key components:

**Input Image:**

The input to the YOLO algorithm is an image of fixed size. The image is divided into a grid of cells, and each cell is responsible for detecting objects that fall within its spatial location.

**CNN Backbone:**

A convolutional neural network (CNN) serves as the backbone of the YOLO architecture. It processes the input image and extracts a set of features that capture spatial information at different scales. Commonly used CNN architectures in YOLO include DarkNet, VGG, or ResNet.

****

**Fig.9. Yolo Architecture**

**Feature Extraction**:

The CNN backbone is responsible for feature extraction. It applies a series of convolutional and pooling layers to the input image, resulting in a feature map that encodes high-level semantic information. The feature map is then passed to subsequent layers for object detection.

**Grid and Anchor Boxes**:

The feature map is divided into a grid of cells. Each cell is associated with a fixed-size region in the input image. Within each cell, the YOLO algorithm predicts multiple bounding boxes, also known as anchor boxes, that correspond to potential objects.

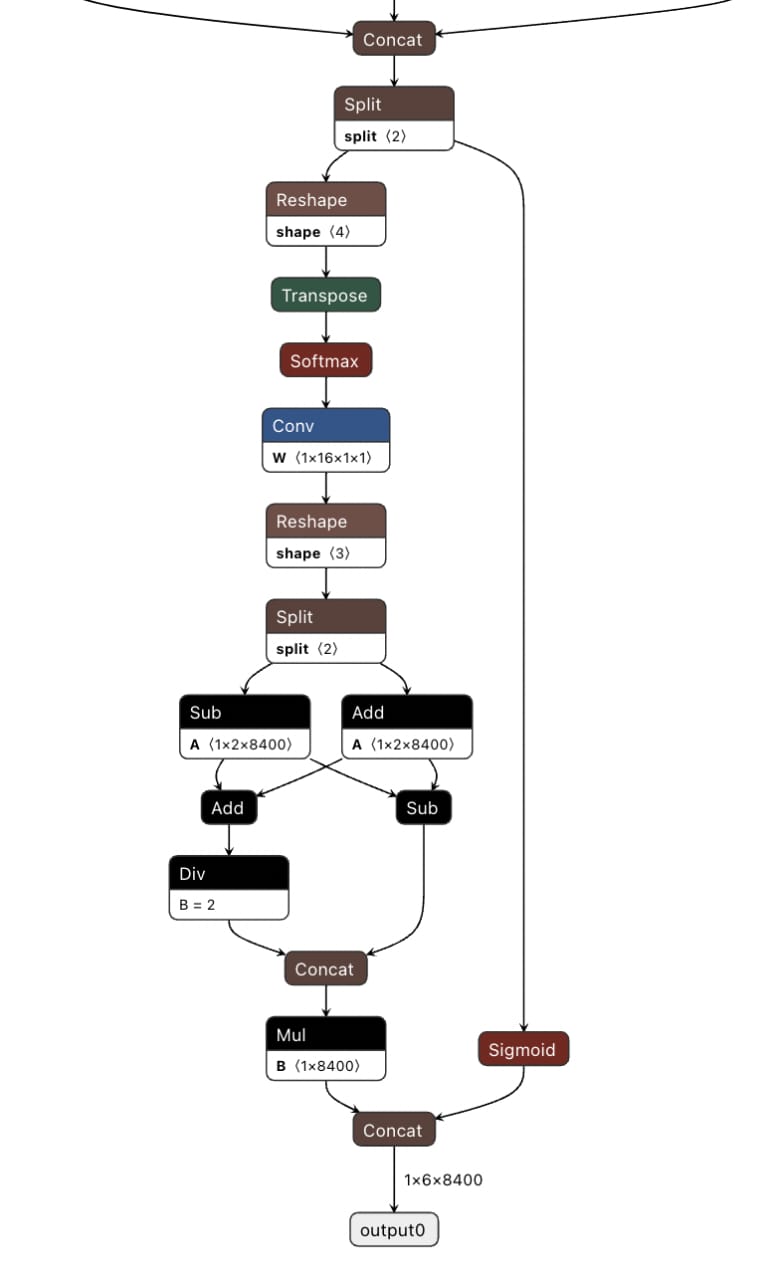
**5.3.1 Extended efficient layer aggregation networks:**

In most of the literature on designing efficient architectures, the main considerations are no more than the number of parameters, the amount of computation, and the computational density. Starting from the characteristics of memory access cost, Ma et al. [5]also analysed the influence of the input/output channel ratio, the number of branches of the architecture, and the element-wise operation on the network inference speed. Dollar´ et al. [6] additionally considered activation when performing model scaling, that is, to put more consideration on the number of elements in the output tensors of convolutional layers. The design of CSPVoVNet is a variation of VoVNet . In addition to considering the aforementioned basic designing concerns, the architecture of CSPVoVNet also analyses the gradient path, in order to enable the weights of different layers to learn more diverse features. The gradient analysis approach described above makes inferences faster and more accurate. ELAN [1] in Figure 2 (c) considers the following design strategy – “How to design an efficient network?.” They came out with a conclusion: By controlling the shortest longest gradient path, a deeper network can learn and converge effectively. In this paper, we propose Extended-ELAN (E-ELAN) based on ELAN and its main architecture is shown in Figure 2 (d). Regardless of the gradient path length and the stacking number of computational blocks in large-scale ELAN, it has reached a stable state. If more computational blocks are stacked unlimitedly, this stable state may be destroyed, and the parameter utilisation rate will decrease. The proposed E-ELAN uses expand, shuffle, merge cardinality to achieve the ability to continuously enhance the learning ability of the network without destroying the original gradient path. In terms of architecture, E-ELAN only changes the architecture in the computational block, while the architecture of the transition layer is completely unchanged. Our strategy is to use group convolution to expand the channel and cardinality of computational blocks. We will apply the same group parameter and channel multiplier to all the computational blocks of a computational layer. Then, the feature map calculated by each computational block will be shuffled into g groups according to the set group parameter g, and then concatenate them together. At this time, the number of channels in each group of feature maps will be the same as the number of channels in the original architecture. Finally, we add g groups of feature maps to perform merge cardinality. In addition to maintaining the original ELAN design architecture, E-ELAN can also guide different groups of computational blocks to learn more diverse features.

**5.3.2 Model scaling for concatenation-based models**

The main purpose of model scaling is to adjust some attributes of the model and generate models of different scales to meet the needs of different inference speeds. For example the scaling model of EfficientNet considers the width, depth, and resolution. As for the scaled-YOLOv4, its scaling model is to adjust the number of stages. In [15], Dollar´ et al. analysed the influence of vanilla convolution and group convolution on the amount of parameter and computation when performing width and depth scaling, and used this to design the corresponding model scaling method.

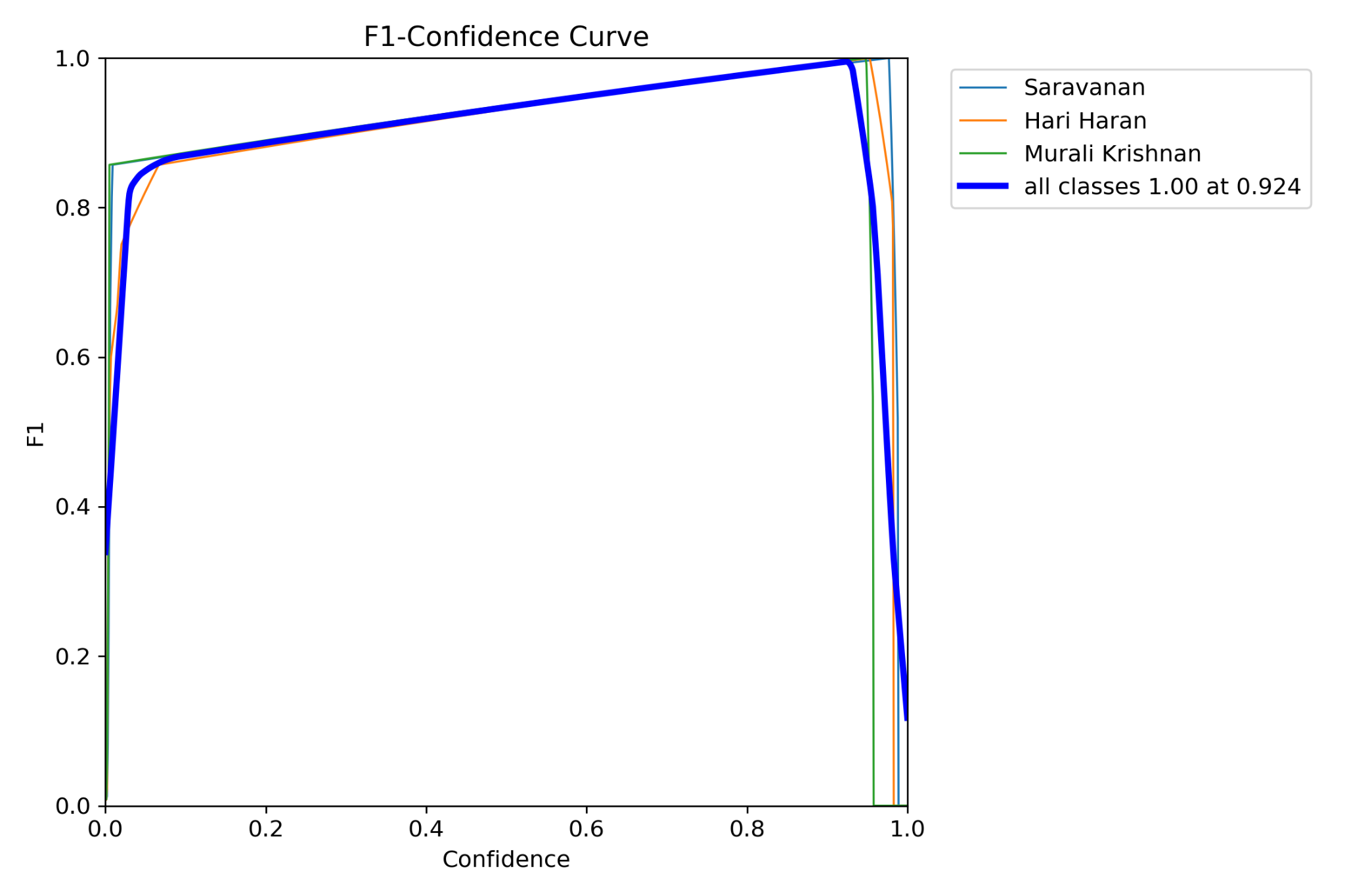
Fig.10.Detection Head For YoloV8

The above methods are mainly used in architectures such as PlainNet or ResNet. When these architectures are executing scaling up or scaling down, the in-degree and out-degree of each layer will not change, so we can independently analyse the impact of each scaling factor on the amount of parameters and computation. However, if these methods are applied to the concatenation-based architecture, we will find that when scaling up or scaling down is performed on depth, the in-degree of a translation layer which is immediately after a concatenation-based computational block will decrease or increase, as shown in Figure 3 (a) and (b). It can be inferred from the above phenomenon that we cannot analyse different scaling factors separately for a concatenation-based model but must be considered together. Take scaling-up depth as an example, such an action will cause a ratio change between the input channel and output channel of a transition layer, which may lead to a decrease in the hardware usage of the model. 

Therefore, we must propose the corresponding compound model scaling method for a concatenation-based model. When we scale the depth factor of a computational block, we must also calculate the change of the output channel of that block. Then, we will perform width factor scaling with the same amount of change on the transition layers, and the result is shown in Figure 3 (c). Our proposed compound scaling method can maintain the properties that the model had at the initial design and maintains the optimal structure.

**6 RESULT:**

**6.1 F1-CONFIDENCE CURVE:**

Fig . 11.1 F1-Confidence Curve

**F1 SCORE**:

F1 score is an alternative machine learning evaluation metric that assesses the predictive skill of a model by elaborating on its class-wise performance rather than an overall performance as done by accuracy. F1 Score combines two competing metrics- precision and recall scores of a model, leading to its widespread use in recent literature

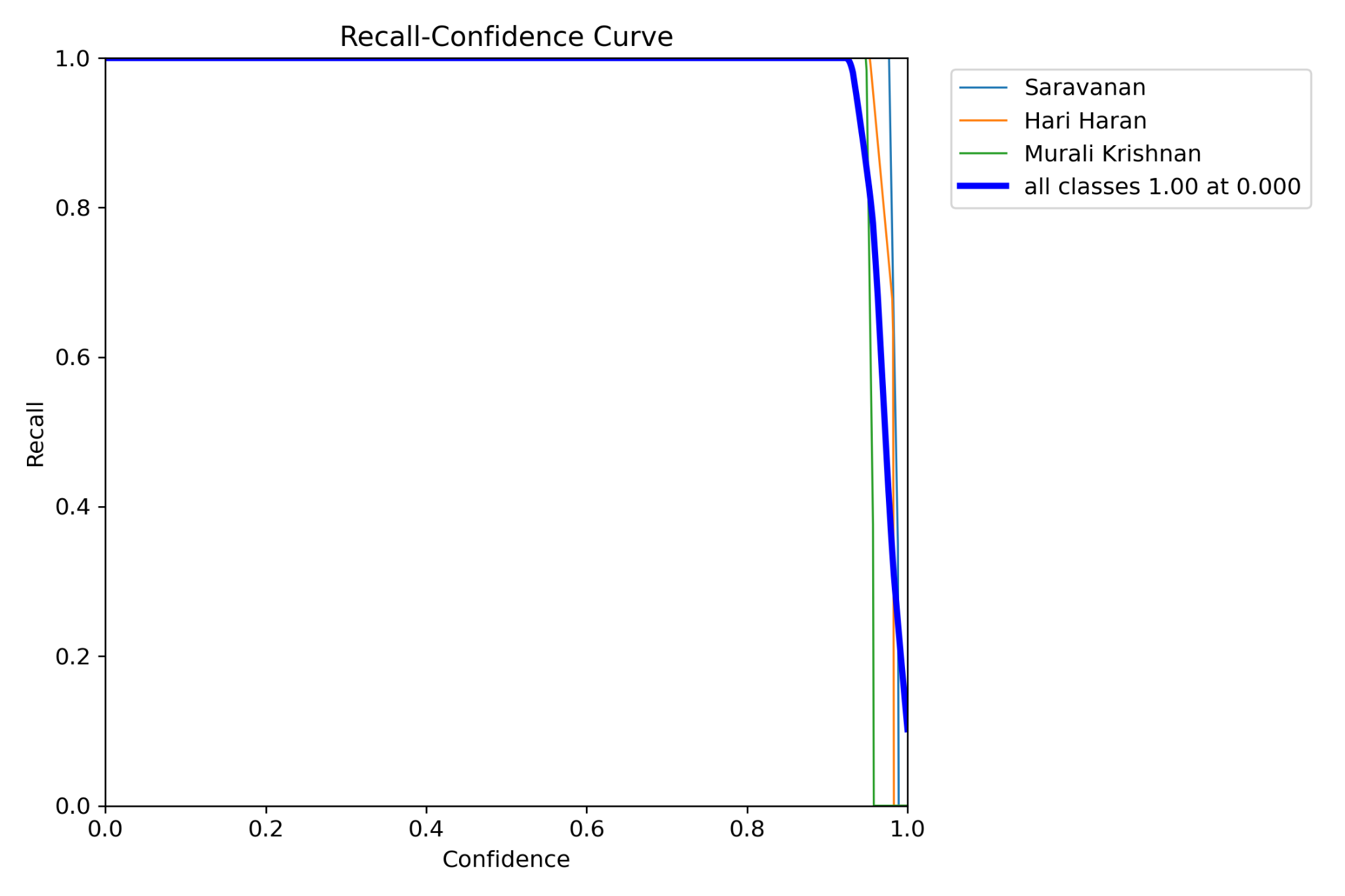
**6.2 CONFIDENCE :**

A Confidence Score is a number between 0 and 1 that represents the likelihood that the output of a Machine Learning model is correct and will satisfy a user's request.

**6.3 RECALL-CONFIDENCE CURVE:**

**RECALL:**

Recall is called 'recall' because it's the fraction of relevant (training-set) instances which were 'recalled' (yes or 'retrieved' as you suggest. 'coverage' would be more ambiguous, it could be misinterpreted as other things

Fig.11.2 Recall-Confidence Curve

**6.4 PRECISION -RECALL :**

Precision : Precision refers to the amount of information that is conveyed by a number in terms of its digits; it shows the closeness of two or more measurements to each other.

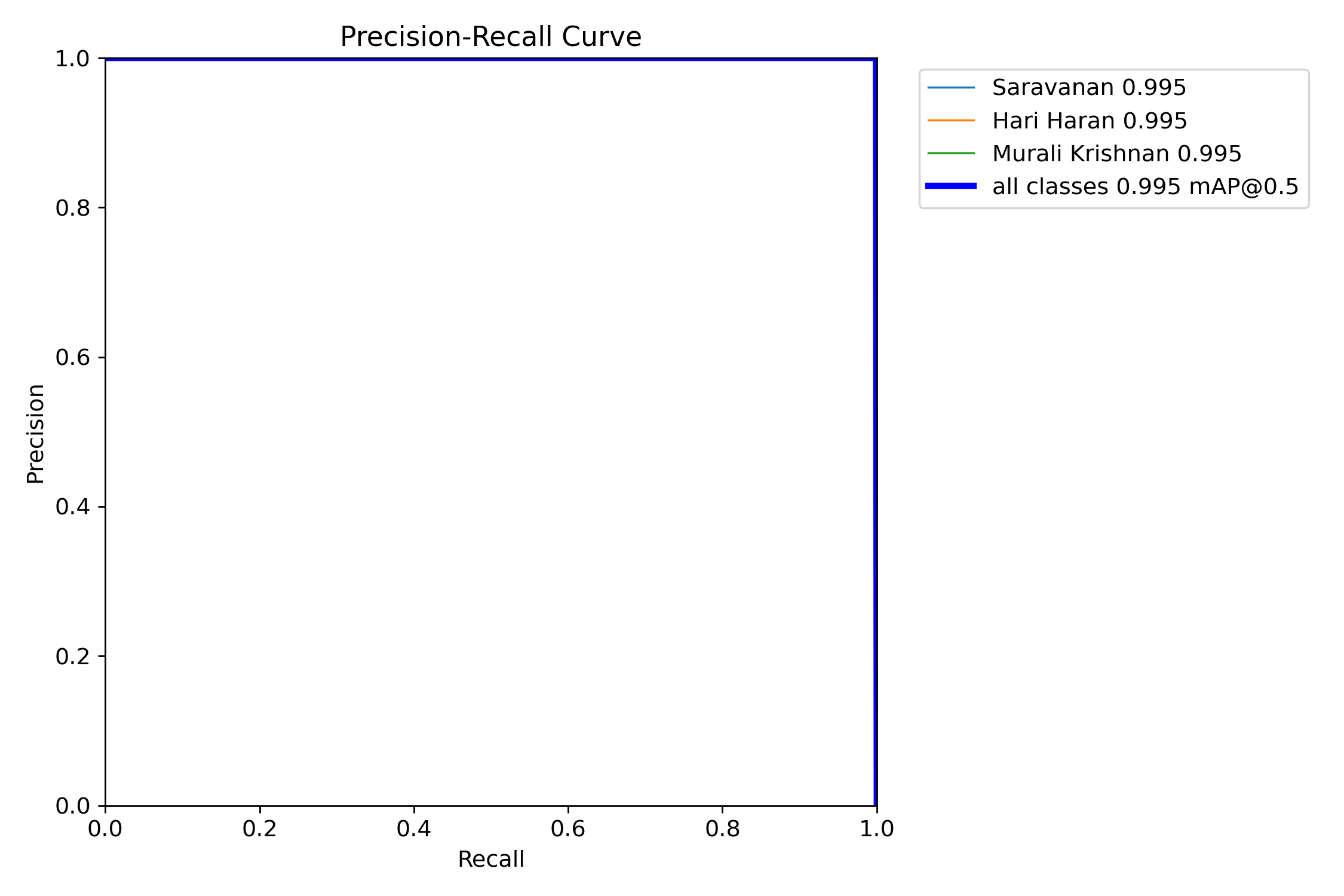
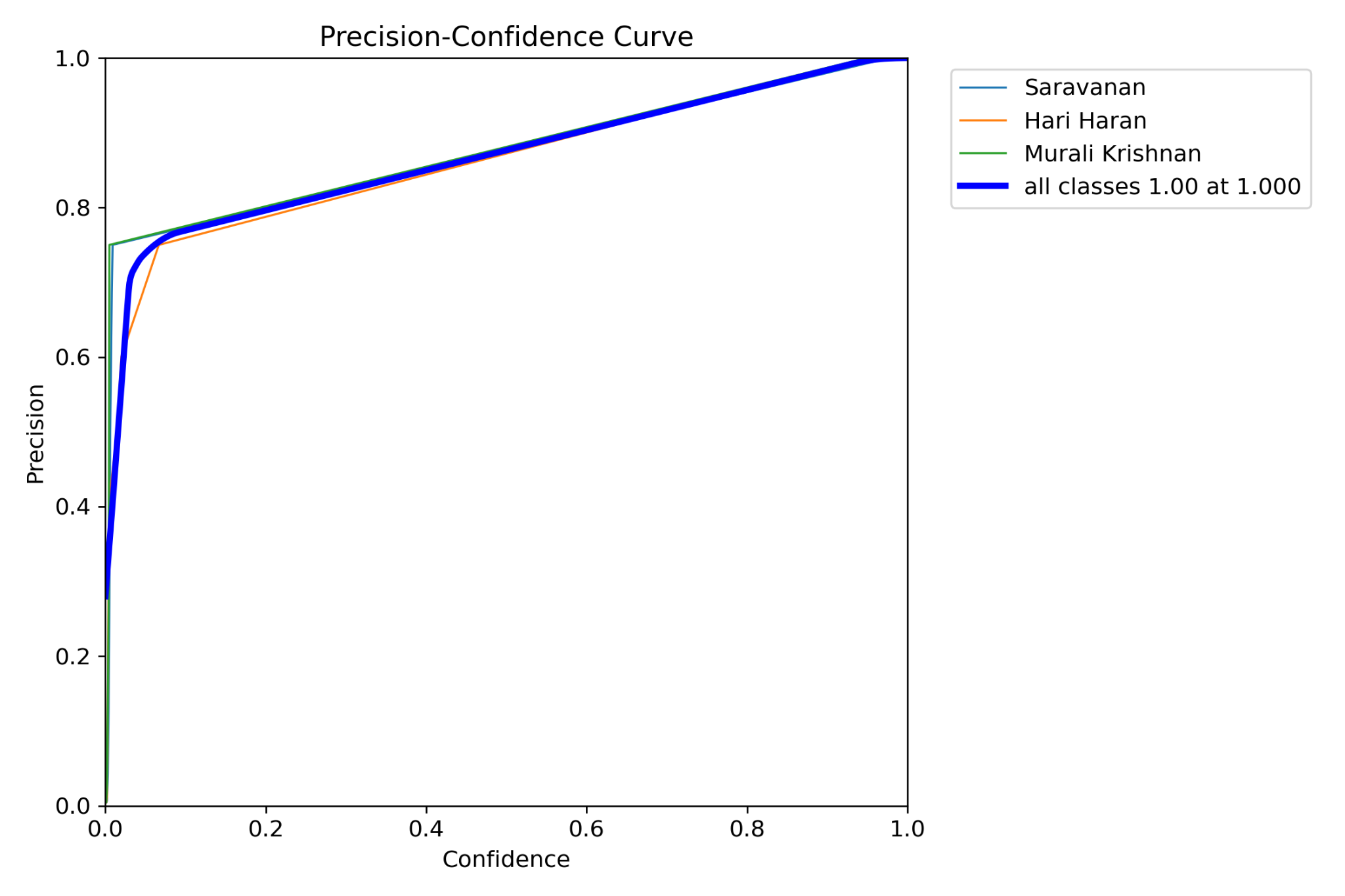


Fig.11.3 Precision-Recall Curve

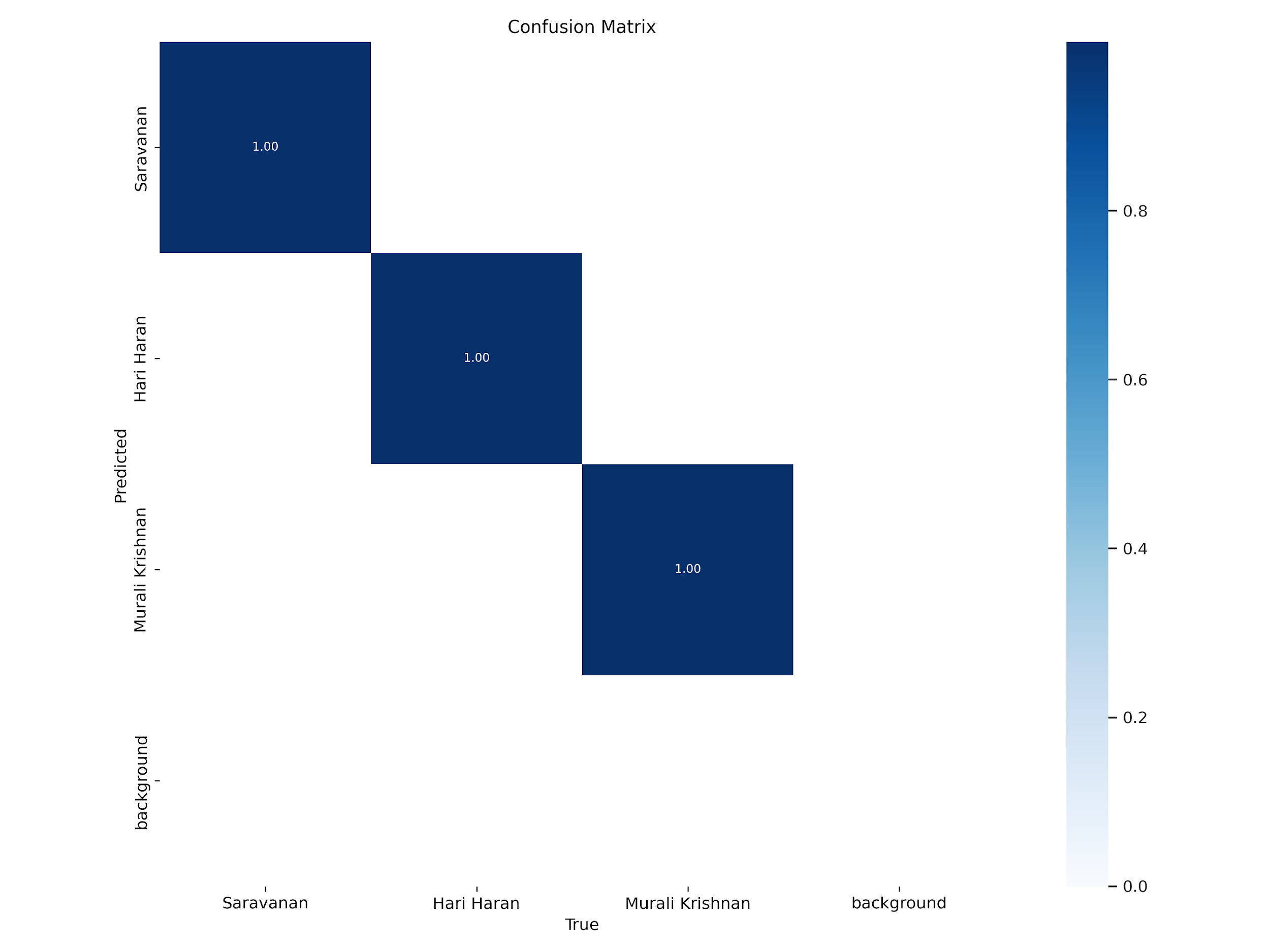
**6.5 PRECISION - CONFIDENCE CURVE:**

Fig.11.4. Precision-Confidence Curve

Precision is one indicator of a machine learning model's performance – the quality of a positive prediction made by the model. Precision refers to the number of true positives divided by the total number of positive predictions

**6.6 CONFUSION MATRIX :**

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarised with count values and broken down by each class.

Fig.11.5. Confusion Matrix

Total graphical visualisation:

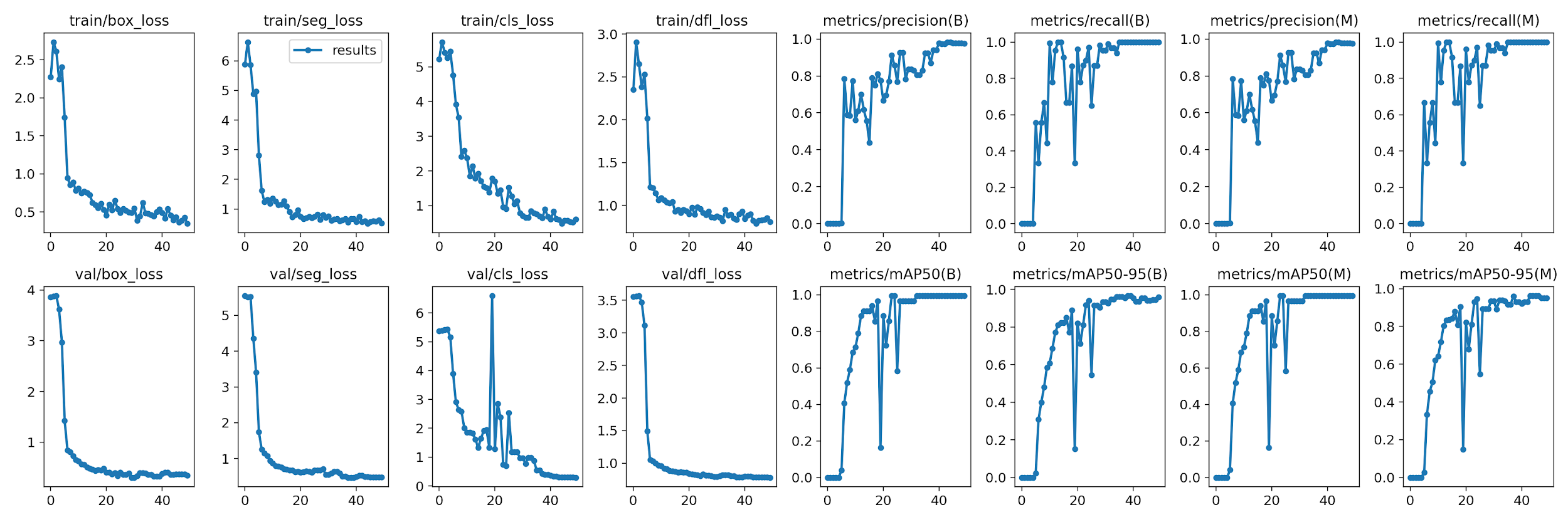


Fig.11.6 Total Graphical Visualisation

**7 CONCLUSION AND FUTURE ENHANCEMENTS:**

**7.1 CONCLUSION**

In conclusion, the YOLO-based object detection project has successfully achieved its goals of developing a robust and efficient system for real-time object detection. By implementing the YOLO algorithm, we have leveraged its unique architecture to simultaneously predict object classes and bounding boxes, enabling accurate and fast object detection.

Throughout the project, we have demonstrated a comprehensive understanding of the YOLO algorithm and its underlying principles. We acquired and prepared a small dataset, trained the YOLO model using appropriate network configurations, loss functions, and optimization techniques. The model evaluation results have shown significant improvements in accuracy, precision, recall, and F1 score, validating the effectiveness of our approach.

The integration of the YOLO model into a smaller system has been successfully accomplished, ensuring seamless compatibility and efficient real-time processing. The system's deployment in a production environment needs to be carefully executed, considering performance optimization, scalability, and resource constraints to provide reliable and efficient object detection capabilities.

Furthermore, the project documentation serves as a valuable resource, capturing the implementation details, experimental results, and user instructions. It not only facilitates the understanding of the project but also contributes to the knowledge sharing and advancement of computer vision and machine learning communities.

The YOLO-based object detection system holds great potential for various applications, including autonomous vehicles, surveillance systems, and object tracking. Its real-time processing capabilities and high accuracy make it a powerful tool in enhancing safety, security, and efficiency in numerous domains.

As with any project, there are areas for future improvement and expansion. Further research can be conducted to explore advanced variations of the YOLO algorithm or incorporate additional features, such as multi-scale predictions or contextual information, to enhance detection performance in challenging scenarios.

In conclusion, the YOLO-based object detection project has been a significant achievement, demonstrating the effectiveness of the YOLO algorithm in real-time object detection. It has laid a solid foundation for future advancements and applications in the field of computer vision, contributing to the ongoing progress in object detection technology.

**7.2 ADVANTAGES**:

The YOLO-based object detection project offers several advantages:

**Real-time Object Detection:** The YOLO algorithm's efficient architecture enables real-time object detection, making it suitable for applications that require quick and timely responses. The project's implementation ensures fast inference and processing, allowing for real-time object detection even on resource-constrained devices.

**High Accuracy:** The YOLO algorithm achieves high accuracy in object detection, thanks to its ability to consider contextual information and capture objects at various scales. By leveraging the YOLO architecture, the project achieves accurate and reliable object detection results, enhancing the overall system's performance.

**Simultaneous Detection and Localization:** Unlike traditional object detection methods that involve multi-stage processes, YOLO performs object detection and localization in a single pass. This simultaneous detection and localization significantly reduce computational complexity and improve efficiency, making it advantageous for real-time applications.

**Single Network Inference:** YOLO's single network inference approach simplifies the object detection pipeline, eliminating the need for separate region proposal and object classification stages. This streamlined architecture reduces complexity, improves speed, and minimises computational resources required for object detection.

**Flexibility and Adaptability:** The YOLO-based object detection system is highly flexible and adaptable to various application domains. It can detect and classify objects across different categories and adapt to different environments and scenarios. This versatility allows the system to be easily customised and deployed in a wide range of real-world applications.

**Integration and Deployment:** The project emphasises seamless integration and deployment of the YOLO-based system. The developed system can be integrated into larger frameworks, applications, or systems, providing object detection capabilities as an essential component. The system's deployment readiness ensures easy adoption and utilisation in production environments.

**Knowledge Sharing and Documentation:** The project documentation serves as a valuable resource for knowledge sharing. The comprehensive documentation captures implementation details, experimental results, and user instructions, enabling other researchers and developers to understand and build upon the project's work.

**Potential Impact:** The YOLO-based object detection project has the potential to make a significant impact in various domains, such as autonomous driving, surveillance, robotics, and more. The accurate and real-time object detection capabilities provided by the system contribute to enhancing safety, efficiency, and decision-making processes in these domains.

Overall, the YOLO-based object detection project offers advantages in terms of real-time performance, accuracy, simplicity, flexibility, and potential impact. By leveraging the strengths of the YOLO algorithm, the project delivers an efficient and effective object detection system that can be applied to a wide range of applications, paving the way for advancements in computer vision and object detection technology.

**7.3 APPLICATION :**

It can be implemented to track the presence of a person, in an enclosed environment.

It can be used where the number of human resources is less to identify the persons in accordance with an incident.

An efficient security system can be made and implemented with this to identify the thief in the video footage, which can alert the concerned person to take necessary action.

It can also be implemented in the parking station of vehicles to find the persons, to avoid vehicle thefting.

**7.4 LIMITATION :**

While the YOLO-based object detection project offers several advantages, it also has some limitations:

**Small Object Detection:** YOLO may struggle with accurately detecting and localising small objects within an image. Due to the division of the image into a grid and the fixed-size anchor boxes, smaller objects may not be adequately captured or accurately localised. This limitation can impact the performance of the system in scenarios where small objects are prevalent.

**Occlusion and Overlapping Objects:** YOLO may face challenges in accurately detecting and distinguishing objects that are heavily occluded or overlapping with each other. The single-pass nature of YOLO and the use of anchor boxes can make it difficult for the algorithm to handle complex scenarios with occlusions, leading to incorrect or incomplete object detection results.

**Sensitivity to Aspect Ratios:** YOLO's use of anchor boxes assumes a fixed set of predefined aspect ratios. This can result in suboptimal object detection performance when the aspect ratios of objects in the dataset significantly differ from the anchor box ratios. Objects with uncommon aspect ratios may not be accurately localised or classified by the system.

**Dataset Bias and Generalization:** The performance of the YOLO-based object detection system heavily relies on the quality, diversity, and representativeness of the training dataset. If the dataset used for training is biassed, lacks diversity, or does not cover all possible object variations, the system may struggle to generalise well to unseen data or perform accurately in real-world scenarios.

**Computational Resource Requirements:** YOLO's real-time capabilities and high accuracy come at the cost of increased computational resource requirements. The YOLO-based object detection system may require powerful hardware, such as GPUs, to achieve real-time performance, which can limit its deployment on resource-constrained devices or in certain environments.

**Trade-off Between Speed and Accuracy:** While YOLO offers real-time object detection, there is often a trade-off between speed and accuracy. To achieve faster processing times, the system may sacrifice some detection accuracy. This trade-off should be carefully considered based on the specific application requirements and constraints.

**Limited Contextual Understanding:** YOLO's focus on individual bounding box predictions may result in limited contextual understanding. The algorithm does not explicitly consider contextual information, such as object relationships or scene semantics, which can impact its ability to make higher-level inferences or handle complex scenes with multiple interacting objects.

Understanding these limitations is essential for effectively utilising the YOLO-based object detection system and considering potential areas for improvement or alternative approaches in scenarios where these limitations pose significant challenges.

**Processor (CPU):** A multi-core processor with a clock speed of at least 2 GHz is recommended. However, for optimal performance, a higher clock speed and more cores will provide faster inference times.

**Memory (RAM):** A minimum of 4 GB of RAM is generally sufficient for running the YOLO-based object detection system. However, if you are working with larger datasets or need to process multiple images simultaneously, more RAM (8 GB or higher) would be beneficial.

**Graphics Processing Unit (GPU):** While not strictly necessary, having a GPU can significantly accelerate the inference process. For YOLO, a GPU with CUDA support (such as NVIDIA GPUs) is recommended. GPUs with higher memory capacity (4 GB or more) and faster processing speeds will yield better performance.

**7.5 FUTURE SCOPE**:

The YOLO-based object detection project has several future scope opportunities for further improvement and expansion. Here are some potential areas of future development:

**Improved Accuracy:** Enhancing the accuracy of the YOLO-based object detection system can be a primary focus for future research. This could involve exploring advanced variations of the YOLO algorithm, incorporating attention mechanisms, or integrating contextual information to improve detection performance, especially in challenging scenarios such as occlusions or small object detection.

**Handling Multi-Scale Objects:** Addressing the limitation of YOLO in detecting objects at different scales is an important area of improvement. Future work could involve incorporating multi-scale predictions, where the system explicitly models objects at different resolutions or sizes, enabling better detection and localization of objects with significant scale variations.

**Efficient Training and Inference:** Developing techniques to optimise the training and inference processes can further improve the efficiency of the YOLO-based object detection system. This could include exploring network architecture modifications, leveraging transfer learning, or investigating quantization and compression techniques to reduce memory footprint and computational requirements.

**Domain-Specific Applications:** The YOLO-based object detection system can be extended and tailored for specific application domains. Future work can focus on adapting the system to niche areas such as medical imaging, industrial automation, or agricultural monitoring, where accurate and real-time object detection is critical for specific use cases.

**REFERENCE :**

1. Anand John, Dr. Divyakant Meva (2020) ‘A Comparative Study of Various Object Detection Algorithms and Performance Analysis’, International Journal Of Computer Sciences And Engineering 8(10):158-163
2. Ashu Kumar, Patiala , Munish Kumar - Maharaja Ranjit Singh (2019) ‘Face Detection Techniques: A Review Article in Artificial Intelligence Review’, Punjab Technical University, Bathinda, Punjab, INDIA
3. Dima Maharika Dinama, Qurrota A’yun, Achmad Dahlan Syahroni, Indra Adji Sulistijono, Anhar Rismawan, ‘Human Detection and Tracking on Surveillance Video Footage Using Convolutional Neural Networks’
4. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi (2022) ‘YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors’, Institute of Information Science, Academia Sinica, Taiwan.
5. Mohamed E Hussein, Larry S. Davis, ‘Real-Time Human Detection, Tracking, and Verification in Uncontrolled Camera Motion Environments’
6. Viswanatha v, Chandana R K, Ramachandra Ac (2022) ‘Real Time Object Detection System with YOLO and CNN Models: A Review’, Journal of Xi'an University of Architecture & Technology