

FACE MASK DETECTION USING YOLOV5

A PROJECT REPORT

Submitted by

RESHMA MOHAN (TKM19MCA021)

to

The APJ Abdul Kalam Technological University

In partial fulfillment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS



**Thangal Kunju Musaliar College of Engineering
Kerala**

DEPARTMENT OF COMPUTER APPLICATIONS

MAY 2022

DECLARATION

I undersigned hereby declare that the project report on "FACE MASK DETECTION USING YOLOV5", submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Dr.Fousia M Shamsudeen. This submission represents my ideas in my own words and where ideas or words of others have been included,I have adequately and accurately cited and referenced the original sources. I also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University..

Kollam

26/05/2022



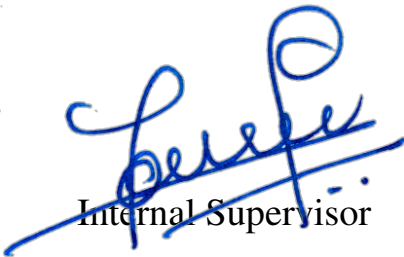
Reshma Mohan

DEPARTMENT OF COMPUTER APPLICATIONS
TKM COLLEGE OF ENGINEERING



C E R T I F I C A T E

This is to certify that, the project report entitled “**FACE MASK DETECTION USING YOLOV5**” is submitted by **RESHMA MOHAN (TKM19MCA021)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications, is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.


Internal Supervisor


Head of the Department

External Examiner

ACKNOWLEDGEMENT

First and foremost I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely grateful to **Dr. Fousia.M.Shamsudeen**, Head of the Department, for providing me with best facilities.

I would like to thank my project guide **Dr.Fousia.M.Shamsudeen**, Department of Computer Applications, who motivated me throughout the work of my project.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project.

Reshma Mohan

ABSTRACT

Global pandemic COVID-19 circumstances emerged in an epidemic of dangerous disease in all over the world. Masks play a crucial role in protecting the health of individuals against respiratory diseases, as is one of the few precautions available for COVID-19 in the absence of immunization. Wearing a face mask will also help prevent the spread of infection . But most of the people are not wearing face masks in public places which increases the spread of viruses. Therefore, automated and efficient face detection methods are essential for such enforcement. In this paper, a face mask detection model has been presented which classifies the images as “with mask”, “without mask” and ”incorrect mask”. The model is trained and evaluated using the Kaggle dataset. The object detection algorithm YOLOV5 based on deep learning is used for detecting mask. The experimental results show that the algorithm proposed in this paper can effectively recognize face masks and realize the effective monitoring of personnel. The experimentation mAP of YOLOV5 exhibited average of 85.7% which proves the effectiveness of face mask detection.

Contents

1	Introduction	1
1.1	Problem Definition	3
1.2	Objective	3
2	Related Works	4
3	Face Mask Detection System	8
3.1	Challenges of Face Mask Detection	8
3.2	Face Mask Detection Using Yolov5	9
4	Methodology	11
4.1	Dataset	12
4.2	Data Preparation and Preprocessing	12
4.3	Selecting a pre-trained model of yolov5	14
4.4	Training the model	14
4.5	Testing the model	14
4.6	Deploying the model using Gradio	15
4.7	YOLOV5 Architecture	15
5	Experimental setup and results	18
5.1	Software Requirement	18
5.2	Training and Validation results	22
5.3	Performance Metrics	24
5.4	Confusion Matrix	27

6 Conclusion	32
6.1 Advantages	32
6.2 Future Enhancement	33
References	34

List of Figures

3.1	Block Diagram for Prediction	10
4.1	System Architecture	12
4.2	Dataset.yaml file	13
4.3	YOLOV5 Architecture	17
5.1	Training Result	22
5.2	Overall Result	23
5.3	Testing Result	23
5.4	Precision of the developed model	26
5.5	Recall graph of the developed model	27
5.6	f1 Score graph of the developed model	28
5.7	Confusion matrix	29
5.8	upload Image	30
5.9	Detected Images of with and without Mask	30
5.10	Detected Incorrect Mask	31

Chapter 1

Introduction

The coronavirus pandemic is responsible for producing an atmosphere of terror as this disease can transmit through the respiratory system. This virus has killed more than a million people around the globe. The world has not yet fully recover from this pandemic and the vaccine that can effectively treat Covid-19 is yet to be discovered. However, to reduce the impact of the pandemic on the country's economy, several governments have allowed a limited number of economic activities to be resumed once the number of new cases of Covid19 has dropped below a certain level. As these countries cautiously restarting their economic activities, concerns have emerged regarding workplace safety in the new post-Covid-19 environment. Currently, there is not any specific single medicine or vaccine in hand to fight against this virus. Therefore, the only option left is to take the utmost care from our side to stay away from the disease. For example, you can maintain the social distancing, wash your hand regularly, and wear a mask. To take part in the protection against the pandemic, here the aim is to design a Face Mask Detection program using the famous Deep Learning technique. This technique is useful to find out who is not wearing the facial mask. The WHO report points out that there are two ways of coronavirus spread i.e. the respiratory droplets and any type of physical contact. The droplets are produced through the respiratory system in case an infected person is coughing or sneezing. If a human is present closer than 4 feet, there are high chances he can inhale these infection-causing droplets. These droplets can stick on those surfaces where the virus can live for days. This way the infected person's surroundings can become a big reason for virus spread. To prevent the virus from the spread, medical masks are the best bet. In the research, medical masks mean surgical as well as the procedure masks and maybe look like

a cup shape or folded. These masks can be attached to the head with cords. They are examined well to control the filtration, easy breathing, and some time for water resistivity. The research examines the collection of video as well as the images to find out those persons who are wearing those medical masks that are according to the govt. guidelines. This way, it can greatly assist the govt. to do action against those people who are not wearing the right type of masks. Using the mask in public has been a normal thing in countries like China and other Asian countries with the very start of the pandemic. Currently, the USA is under the grip and severe pandemic outbreak and cases are increasing day by day along with the confirmed deaths. The mask acts as a physical barrier to prevent the entry of the virus. Many individuals unknowingly infected many other people with the Covid-19.

This paper under study will make deep learning about using masks to prevent the spread of deadly coronavirus. It will also learn the facemask detection by deep learning strategy. This is a useful technique of learning being used these days. Deep learning is a subfield of machine learning. We study a hierarchy of features and functions of the subject under study with the help of input data. The researchers are found to use this technique intensively for their research work. They used it in their research related to image classification, speech recognition, signal processing, and natural language processing. Deep learning is similar to machine learning. It builds a hierarchy of features from top to bottom. The systematic and arranged features are easy to understand and explain to others as well. This method of learning can learn the features at any level stigmatically. The help of human-made techniques is not required. The best thing about deep learning is that all the models have deep architectures. Being the best opposite of shallow architecture which has few hidden layers, deep architecture has several layers. The Regression, Classification, dimensionality reduction, modelling motion, modelling textures, information retrieval, natural language processing, robotics, fault diagnosis, and road crack detection are some important fields that are found deriving benefit from deep architecture techniques.

Deep learning has gained more attention in object detection and was used for human detection purposes and develop a face mask detection tool that can detect whether the individual is wearing mask or not. This can be done by evaluation of the classification results by analyzing real-time streaming from the webcam. In deep learning projects, we need a training data set. It is the actual dataset used to train the model for performing various actions.

1.1 Problem Definition

Many people have come up with creative ideas for detecting face masks most of them solved this issue using simple CNN models and uses two stage detectors like Faster R-CNN considering it as a binary classification problem. Although the two-stage detectors have good detection performances, their training phases are complex, and the testing speeds are usually slow, so they are not suitable for the real-time requirement. This problem needs to be handled using object detection models for detecting multiple people in the frame, then putting bounding boxes of particular colour around them depending on whether they are wearing masks, not wearing masks and wearing mask in an incorrect manner. Most recent and advanced face mask detection approaches are designed using deep learning object detection. In this system, object detection model YOLOv5 is used to achieve this task.

1.2 Objective

The main goal of project is:

- To develop a system that will draw bounding boxes around the faces of people, based on whether a person is wearing a mask or not.
- Effectively recognize face masks and realize the effective monitoring of personnel.
- To train the model on a dataset that consists of images of people of three categories that are with ,without and incorrect face masks.
- Deploying the model with higher accuracy using Gradio.

Chapter 2

Related Works

In this section, several studies of Face mask detection techniques utilizing both machine learning and deep learning are discussed.

Hiten Goyal et al.[1] used Convolution Neural Network layers (CNN) as its backbone architecture to create different layers. Along with this, libraries such as OpenCV, Keras, and Streamlit are also used. Gathered data-set comprises approximately about 4,000 pictures and attained a performance accuracy rate of 98 %,but leads to problems such as poor generalization ability, redundant information, poor real-time performance.

Xueping Su et al.[2]proposes a new algorithm for face mask detection that integrates transfer learning and Efficient-Yolov3, using EfficientNet as the backbone feature extraction network, and choosing CIoU as the loss function to reduce the number of network parameters and improve the accuracy of mask detection but the training process is prone to overfitting, which reduces the robustness of the model.

Sunil Singh et al.[3] discussed two state-of-the-art object detection models, namely, YOLOv3 and faster R-CNN. The authors have trained both the models on a dataset that consists of images of people of two categories that are with and without face masks.Although the two-stage detectors have good detection performances, their training phases are complex.YOLOv3 is much faster than Faster R-CNN but the quality of faster R-CNN output is indeed greater than YOLO's output.

Harish Adusumalli et al.[4] discussed a method uses machine learning classification using OpenCV and Tensorflow to detect facemasks on people. But the dataset is small, so accuracy is comparatively low.

Mohammad Marufur Rahman et al[5] proposes a deep learning architecture which is trained on a dataset that consists of images of people with and without masks collected from various sources. The developed architecture is trained for 100 epochs since further training results cause overfitting on the training data. Training accuracy increases but test accuracy decreases.

Retinal Face Mask Detector proposed by Mingjie Jiang, Xinqi fan and Hong[6], here introduces a Retinal Face Mask Detector. It is a One-stage object detector. The dataset contained 7959 images. The ResNet and mobile Net used as BACKBONE. But ResNet is considered as standard backbone. The detection network includes a backbone, a neck and head modules. As a result the ResNet accuracy is very much higher than the Mobile Net, but took more time for detection .

Chun et al. [7] analyzed the performance of YOLOV3 for face detection in complex environment using WIDER FACE. The experimental analysis reveals that the confidence set in YOLOV3 is relatively large due to being trained on COCO dataset. Since WIDER FACE contains images of small and medium sized faces mainly so high confidence setting in YOLOV3 missed detection of many small faces. However, multiple clustering adjustments on priori box had improved the accuracy in face detection from 57.9% to 80.5% but limited by low detection speed.

Nguyen, N. D., Do, T., Ngo, T. D., Le, D. D.[8] introduces Fast R-CNN, an extension over R-CNN and SPPNet. It introduces a new layer named Region of Interest (RoI) pooling layer between shared convolutional layers to fine-tune the model. Moreover, it allows to simultaneously train a detector and regressor without altering the network configurations. Although Fast-R-CNN effectively integrates the benefits of R-CNN and SPPNet but still lacks in detection speed compared to single stage detectors

Roy et al. [9] compared the performance of YOLOV3, YOLOV3-tiny and R-CNN using Moxa3K benchmark dataset for monitoring of people wearing masks. In his work, Inception V2 is taken

as the backbone with R-CNN using Tensor flow. The mAP@50 (Mean Average Precision) score obtained by R-CNN with inception V2 was 63.99%, YOLOV3 configured in Darknet was 60.5% and YOLOV3-tiny with Darknet was only 56.57%. Although R-CNN yields high performance for facemask detection but limited by low detection speed.

In an attempt to track and enforce compliance to health protocols, one application of facial recognition was implemented by Li et al.[10] using CNN (Convolution Neural Network) to identify whether an individual was wearing a mask or not. They developed the HGL method to classify head poses using masks for faces, which made use of the analysis of colour texture in photographs and line portraits. The front accuracy was 93.64%, meanwhile the side-to-side accuracy was measured at 87.17%. The model achieves reasonably good accuracy on small dataset but requires more memory.

The authors of [11] recommended the use of a pre-trained MobileNet with the global pooling block that can be used for facial recognition and detection. The pre-configured MobileNet creates a multi-dimensional component map from a shaded image. Over-fitting is not a problem in the model proposed since it uses an overall pooling block. But the model struggles to detect face mask under certain conditions such as detection of small faces with masks and confusing masks

Performance evaluation of intelligent face mask detection system with various deep learning classifiers proposed by C. Jagadeeswari, M.Uday Theja.[12] Here the performance of face mask detection using different deep learning classifiers can be analyzed mobileNet V2, ResNet 50, VGG 16, ADAM, SGD. These are the classifiers used for it. For each classifier followed by 3 optimizer and evaluate the performance. The optimizers are used here such as ADAM, ADAGRAD, SGD (Stochastic Gradient Descent). As a result ADAM optimizer performance is very good and also observed that MobileNet V2 classifier has best result with high accuracy but Not very cost efficient and maintenance seems to be high and frequent.

Qin and Li [13] constructed a face mask recognition design using the condition identification process. The problem was broken down into four parts in the paper: preprocessing the picture, cropping the facial regions, super-resolution operation, and predicting the end condition. The

primary innovation during this study was that it used super-resolution to improve low-quality image performance. The proposed method by the author used SRCNet and detected face masks and the mask's position with an accuracy of 98.7%. But the training time and testing time required is more compared to other models.

Facial Mask Detection using Semantic Segmentation proposed by Toshihanlal Meenpal, Ashuthosh Balakrishnan and Amit Verma [14] used a facial mask detection based on semantic segmentation. Here the class labels are named as face or non-face. The convolutional neural network VGG-16 architecture followed by fully convolutional network is used for segmentation. As a result it recognizes multiple faces. This method is useful for frontal faces as well as non-frontal faces. As a result it is focused on removal of erroneous prediction. But the model takes enough time for processing.

Implementation of Principle Component Analysis on Masked and Non-Masked Face Recognition proposed by Md. Sabbir Ejaz and Rabiul Islam [15], here analyzed a masked and non-masked face recognition accuracy by using a principle component analysis. The dataset used is Olivetti and Oracle Research Laboratory (ORL) face database. Here PCA is used for feature extraction. The steps are used in this work includes Facial Image Acquisition and Facial Feature Extraction using PCA and Eigen Vector Calculation. As a result it gives high recognition rate of face without mask but the accuracy reduces significantly

Chapter 3

Face Mask Detection System

A face mask detection is a technique to find out whether someone is wearing a mask or not. It is similar to detect any object from a scene. Many systems have been introduced for object detection. Deep learning techniques are highly used in medical applications. Recently, deep learning architectures have shown a remarkable role in object detection.

3.1 Challenges of Face Mask Detection

In object detection, face mask detection is a crucial task. Detection is the first part of identity authentication and pattern recognition. Deep learning-based algorithms for object detection have evolved in the last years at a high rate. The algorithms can be subdivided into two general parts. These are one-stage detectors such as YOLO and two-stage detector such as Faster R-CNN (Faster Region Based Convolutional Neural Networks). Though YOLO and its varieties aren't as good in terms of accuracy as the two-stage detectors, they outmarch their correspondents in speed by a large margin. When facing standard sized objects, YOLO does well but it is unable to detect small objects. When dealing with objects with faces that seem to have large scale changing properties, the accuracy reduces significantly.

3.2 Face Mask Detection Using Yolov5

To overcome the limitations of current object detection methods, YOLOv5 ultralytics open-source object-detection model based on deep learning is used to detect the images. YOLO is a clever convolutional neural network (CNN) for doing object detection in real-time. It can deal with the difficulty of detection of varying face scales, hence improving the performance of face mask detection. The current technique involves the use of anchor boxes that are more suitable for face detection and a more accurate regression loss function. The enhanced detector improved accuracy significantly while maintaining a fast detection speed.

The algorithm applies a single neural network to the full image, and then divides the image into regions and predicts bounding boxes and probabilities for each region. The proposed model can be integrated with CCTV or laptop cameras allowing it to detect people who are wearing masks, not wearing masks and incorrectly wearing masks. However, deep learning typically requires a large number of labelled samples for training, which is time consuming. Transfer learning, where a model is pre-trained for a task on an existing labelled database is adapted to be reused for a different but related task, is a common workaround to this issue. Yolov5s model is proposed based on the applications of deep learning and convolution neural networks based on transfer learning that are capable of classifying automatically that the individuals wear mask, not wearing or incorrectly wearing mask.

Face Mask Detection using YoloV5, this project mainly aims improve the accuracy and inference speed. The main advantage is the system will accurately detect the images into three categories that are "with mask", "without mask" and "incorrect masks". This paper focuses on improving the detection accuracy of small faces in surveillance images. Because of the comparison of the four versions of YOLOv5 including YOLOv5m, YOLOv5l, YOLOv5x, and YOLOv5s, the YOLOv5s model is smaller and easier to deploy quickly. Therefore, this research is based on the YOLOv5s model.

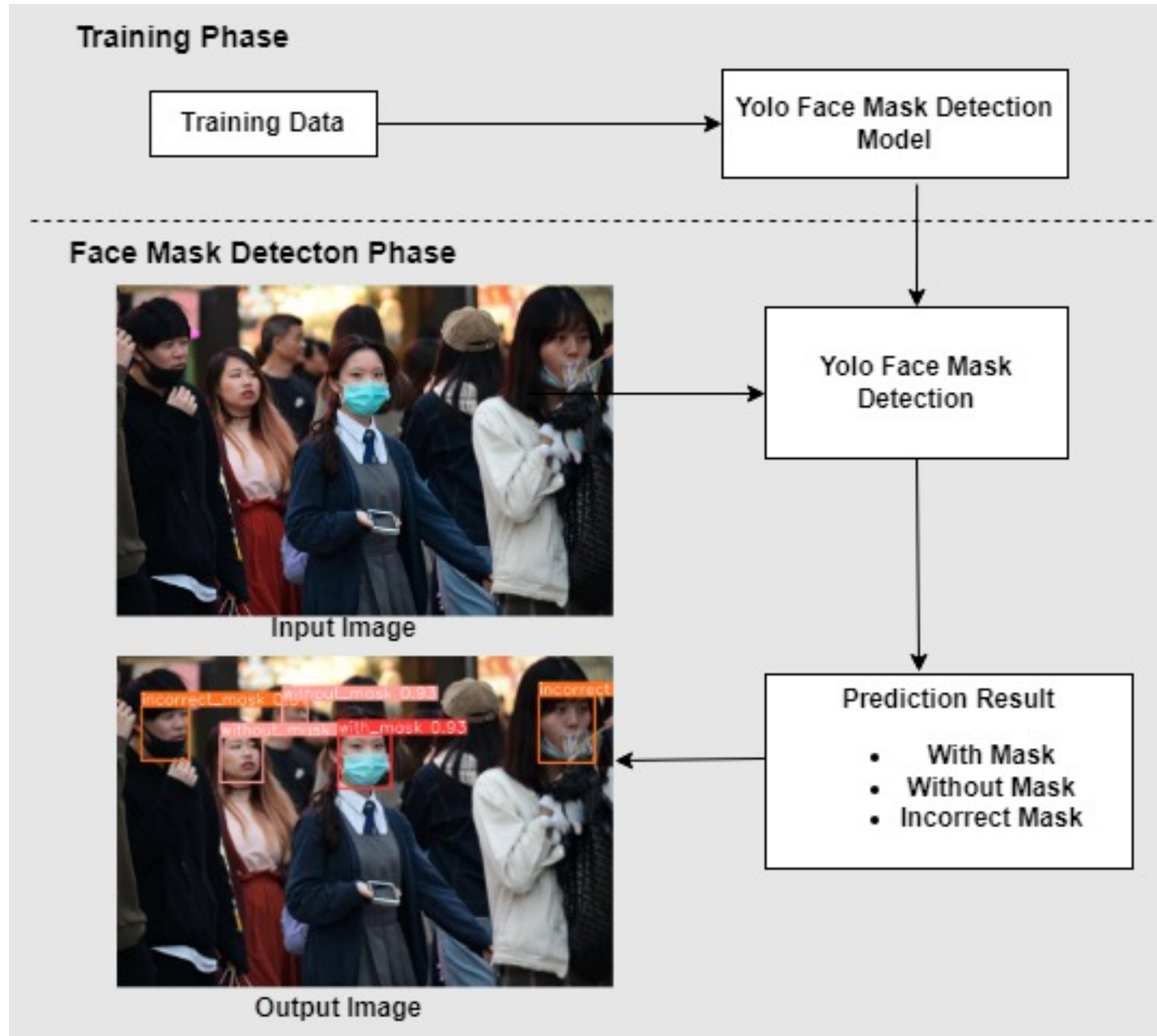


Figure 3.1: Block Diagram for Prediction

Chapter 4

Methodology

The main purpose of the project is the detection of face mask. This system is capable to train the dataset of persons wearing masks, without wearing masks and incorrectly wearing mask. After training the model the system can predicting whether the person is wearing the mask or not. It also can access the webcam and predict the result.

The proposed system consist of four major phases:

- Preparation & Pre-Processing of Dataset.
- Selecting a pre-trained model of yolov5
- Training the model.
- Testing the model
- Deploying the model using Gradio

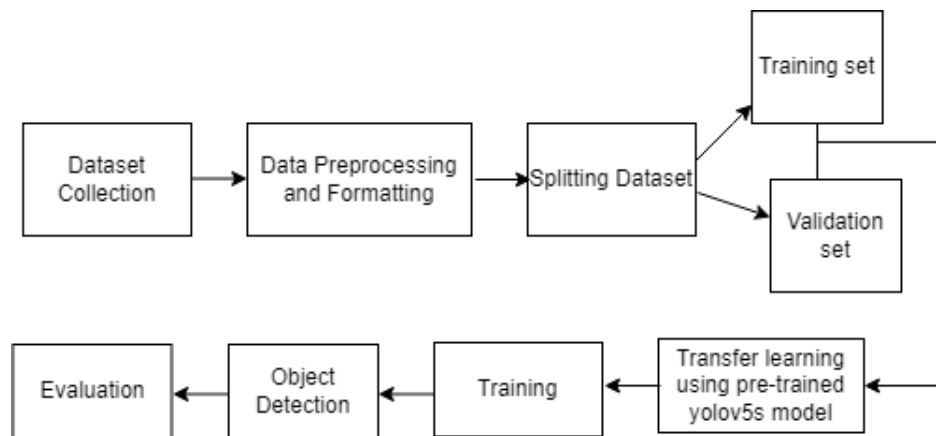


Figure 4.1: System Architecture

4.1 Dataset

The dataset used is from Kaggle website. With this dataset, it is possible to create a model to detect people wearing masks, not wearing them, or wearing masks improperly. This dataset contains 853 images belonging to the 3 classes, as well as their bounding boxes in the PASCAL VOC format. The classes are:

- With mask;
- Without mask;
- Mask worn incorrectly.

The data is modified into the training and validation set to enhance the system and increase efficiency. The dataset can be downloaded from <https://www.kaggle.com/datasets/andrewmvd/face-mask-detection>

4.2 Data Preparation and Preprocessing

Before training the data with the model, the annotation data in PASCAL VOC XML format have to be changed to YOLO format, with one .txt file per image,

- One row per object
- Each row is class x_center y_center width height format.
- Box coordinates must be in normalized xywh format (from 0–1). If your boxes are in pixels, divide x_center and width by image width, and y_center and height by image height.
- Class numbers are zero-indexed (start from 0).

Data preprocessing is an integral step in Deep Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we preprocess our data before feeding it into our model. Data preprocessing is a process of preparing the raw data and making it suitable for our model. It is the first and crucial step while creating a machine learning model. For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner.

Before training the model, we will need to create a project data.yaml file, specifying where the training and validation images are and the number of labels as well as the label names of our training data. The file should be structured as shown:

```
# Train/val/test sets as
## 1) dir: path/to/imgs, 2) file: path/to/imgs.txt, or 3) list: [path/to/imgs1, path/to/imgs2, ...]

train: /content/face_data/train/images
val: /content/face_data/val/images
# test: # test images (optional)

# Classes
nc: 3 # number of classes
names: [ 'with mask', 'without mask', 'incorrect mask' ] # class names
```

Figure 4.2: Dataset.yaml file

4.3 Selecting a pre-trained model of yolov5

Ultralytics supports several YOLOv5 architectures, named P5 models, which varies mainly by their parameters size: YOLOv5n (nano), YOLOv5s (small), YOLOv5m (medium), YOLOv5l (large), YOLOv5x (extra large). These architecture are suitable for training with image size of 640*640 pixels. In this paper, YOLOv5s version is used, because it is the simplest of all. It is the small model in the yolov5 family with around 7.2 million parameters and is ideal for inference on the CPU

4.4 Training the model

Another performance enhancing method in deep models especially in CNNs which is named as transfer learning. Transfer learning is the idea of overcoming the isolated learning paradigm and utilizing knowledge acquired for one task to solve related ones. In this process is a training process using a custom YOLO v5-s model and the results of the training or Yolov5 weight model are used for the detection process. That is all the layers of the small model are trained. This means that, although the pretrained weights will be loaded, the entire model will be fine tuned on a new dataset. After training the model is saved as pt file and the data from the training can be tested using photos and videos.

4.5 Testing the model

The selected data has been partitioned into two parts as training data and testing data. The training data has been used to build a deep learning model and the testing data has been used to validate the model. The model which was built on the training data has been tested on the test data to validate the performance of model based on the accuracy. In this stage, the developed model has been tested using the test dataset. The system processes the data test dataset similar to the training dataset. The testing phase using photos and videos. Input can be photos or videos for testing. The detection process is carried out by loading the model that has been built, then classification and prediction are carried out using bounding boxes and confidence scores. The results displayed are in the form of prediction boxes, confidence values, and object classes. Based on that, the system classifies

the object (facemask). Here we have trained the model for finding the person with and without a facemask.

4.6 Deploying the model using Gradio

Here, GUI for the model is developed using Gradio where user can upload or capture image from webcam which is to be detected and the system predicts whether it is with mask, without mask or incorrect mask accurately. First, the input is from a face captured by the camera. Then the detection process will load the model that has been built, then perform classification and prediction using bounding box and confidence score. The results displayed are in the form of prediction boxes, confidence values, and object classes.

4.7 YOLOV5 Architecture

Object detection is one of the primary tasks in computer vision which consists of determining the location on the image where certain objects are present, as well as classifying those objects. In 2015, the YOLO (You Only Look Once) algorithm was born with a new approach, reframing object detection as a regression problem and performing in a single neural network. That made the object detection field explode and obtained much more remarkable achievements than just a decade ago. So far, combining with many of the most innovative ideas coming out of the computer vision research community, YOLO has been upgraded to five versions and assessed as one of the outstanding object detection algorithms. The 5th generation of YOLO, YOLOv5, is the latest version not developed by the original author of YOLO. However, the performance of the YOLOv5 is higher than the YOLOv4 in terms of both accuracy and speed. This thesis investigates the most advanced inventions in the field of computer vision which were integrated into YOLOv5 as well as the previous versions. Using the Colab platform to implement object detection in the face mask dataset contains 853 images. Subsequently, the YOLOv5 model will be evaluated and configured for improvement based on the results.

The YOLO family of models consists of three main architectural blocks

- Backbone
- Neck
- Head

YOLOv5 Backbone: The backbone model is used to extract important features from a given input image. YOLOv5 uses the CSP (Cross Stage Partial) Network as the backbone to extract informative features from the input image. Feature extraction process uses DenseNet CSP. CSP (Cross Stage Partial) is based on the same principle from DenseNet except that the CSP input is separated into 2 portions, with some of it being forwarded through dense blocks which will perform convolution and part of it being sent directly to the next stage without being processed. Then the part of the block that performs the convolution will be combined (concat). Before entering the neck using FPN, additional blocks are added, namely the SPP (Spatial Pyramid Pooling) layer

YOLOv5 Neck: Neck style on YOLOv5 using PANet which adopts the Feature Pyramids Network (FPN) structure. The PANet model is used to help the model generalize well for scaling objects. This is very helpful in identifying the same object with different sizes and scales. It uses PANet to perform aggregation on the features and pass it to Head for prediction.

YOLOv5 Head: Head model used to perform the final detection, this model applies anchor boxes to the features and produces a final output vector with class probabilities, objectivity scores, and bounding boxes. YOLOv5 uses leaky ReLU and Sigmoid as activation functions. The Leaky ReLU activation function is used in the middle layer or hidden layers while Sigmoid is used in the final detection layer. The activation function is a function to calculate the number of input weights and biases in the artificial neural network so that it can activate and deactivate neurons. The loss value in YOLOV5 is calculated based on objectivity scores, class probability scores, and bounding box regression scores.

Apart from this YOLOv5 uses the below choices for training :-

Activation and Optimization: YOLOv5 uses leaky ReLU and sigmoid activation, and SGD and ADAM as optimizer options. Loss Function: It uses Binary cross-entropy with logits loss.

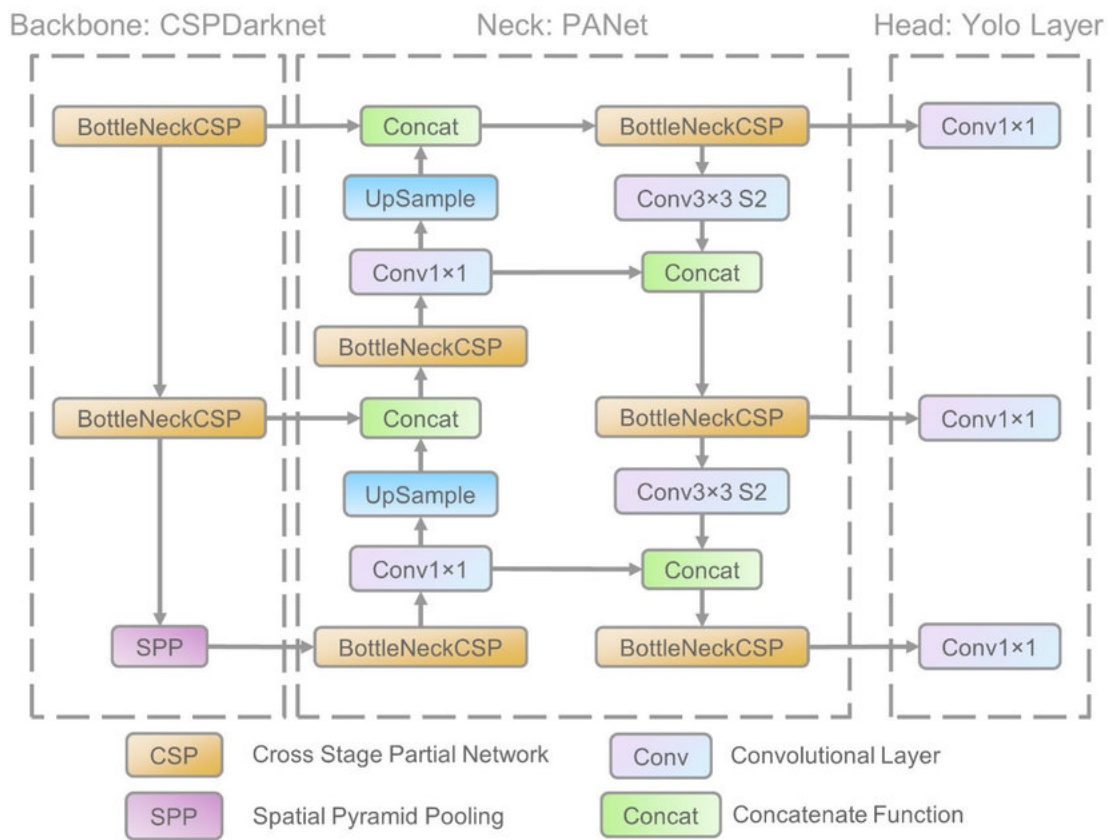


Figure 4.3: YOLOV5 Architecture

Chapter 5

Experimental setup and results

In this section of this paper, the results of the developed model will be discussed. The developed model works fairly on the artificially developed test data. The model accurately classified the persons who wear a mask as well as persons who are not wearing masks. The developed model has been classified people who worn mask and without mask accurately.

5.1 Software Requirement

The software used for the project:

- Python
- Google Colab
- Gradio

1. PYTHON

Python is an object-oriented language control users to manage and management data structures or objects to make and run programs. Everything in Python is, in fact, top-notch. All objects, data types, functions, methods, and classes take an equal position in Python. Programming languages are created to satisfy the requirements of programmers and users for an efficient tool to develop applications that impact lives, lifestyles, economy, and society.

they assist build lives better by increasing productivity, enhancing communication, and rising potency. Languages die and become obsolete once they fail to live up to expectations and are replaced and superseded by languages that are more powerful. Python programming language artificial language that has stood the test of time and has remained relevant across industries and businesses and among programmers, and individual users. it's a living, thriving, and extremely helpful language that's extremely recommended as a primary programming language for those that want to dive into and experience programming.

Browser incompatibilities

When a user receives a page which incorporates JavaScript, the JavaScript interpreter of his browser kicks in and tries to execute the script. Currently, the main downside here is that the assorted browsers each use their own interpreter, which generally browser vendors have chosen to not implement a bit of JavaScript. Their reasons were typically associated with a business advantage over the competitors. therefore the dreaded browser incompatibilities. In addition, every new browser version understands more JavaScript and permits more and more components of the HTML page to be modified by scripts. This results in even more incompatibilities. It is best to resolve compatibility issues on a case by case basis. In fact, most pages on this website have been written precisely thanks to browser incompatibilities. So scan on to know more. But I warn you: you need to digest quite a ton of information. thus it is best to unravel the problem at hand and leave the rest of the knowledge alone till you need it.

- Python is Interpreted : Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive : You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented : Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- Python is a Beginners Language : Python is a great language for the beginnerlevel programmers and supports the development of a wide rangrnditemizee of applications from simple text processing to WWW browsers to games.
- an easy and intuitive language just as powerful as those of the major competitor
- open source, so anyone can contribute to its development
- code that is as understandable as plain English

2. Google Colab

Google Colab was developed by Google to provide free access to GPU's and TPU's to anyone who needs them to build a machine learning or deep learning model. Google Colab can be defined as an improved version of Jupyter Notebook. Jupyter Notebook is an application that allows editing and running Notebook documents through a web browser or an Integrated Development Environment (IDE).

Google Colab provides tons of exciting features that any modern IDE offers, and much more. Some of the most exciting features are listed below.

- Interactive tutorials to learn machine learning and neural networks.
- Write and execute Python 3 code without having a local setup.
- Execute terminal commands from the Notebook. Import datasets from external sources such as Kaggle.
- Save your Notebooks to Google Drive.
- Import Notebooks from Google Drive.
- Free cloud service, GPUs and TPUs.
- Integrate with PyTorch, Tensor Flow, Open CV.
- Import or publish directly from/to GitHub.

3. Gradio

The user interface(UI) is the graphical blueprint of an application. One major and important characteristic to keep in mind while building a data science project is a smooth user interface.

Gradio was released in February 2019 as an open-source library in python for building user interfaces that are straightforward to use and can be easily customized for any machine learning or deep learning model or any arbitrary function. It makes it simpler to play with models in the web browser by just dropping and dragging images, text, recording, etc. and recognizing live output interactively.

Gradio helps build an online GUI in a coding format that is very suitable for exhibiting the model presentation. It is quick, easy to set up, ready to operate, and accessible widely as a public connection. Moreover, it can distantly and parallelly run the model in a machine. Gradio works with a broad range of media-text, pictures, video, and sound. Besides ML models, it can very well be utilized as python code embeddings.

- Demos can be created effortlessly of machine learning codes that are useful for clients/ users/ team members.
- During expansion, we can interactively debug model
- Developers can get feedback on model performance from users. This helps in improving the model easily and faster.
- It can be incorporated with TensorFlow and PyTorch models for better comprehension.

Gradio consists of three parameters:

- fn :- performs the major operation of the user interface
- inputs :- the input component type
- outputs :- the output component type

Components like buttons, text boxes, checkboxes, file dialogs can be customized in Gradio. Moreover, it promotes multiple inputs and outputs. Gradio can also be used to conduct comparative analysis for diverse models.

5.2 Training and Validation results

After setup the model the next step is to train the model. YOLOv5 repository is used as the basic framework. all repositories and dependencies of YOLOv5 must be fulfilled. Dataset training for the face mask detection system was carried out on 853 images that had been labeled. The object in the image is resized to 640x640. The training was carried out for 50epochs with a total of 32 batches. The basic model used is yolov5s.yaml. The training process was completed in 1.013 hours hours for 853 images labeled with 50epochs. The resulting weight model is 14.9MB. The training process is a time and resource-consuming process in deep learning. Because the overall result of the process mainly depends on the quality of the training process. In this project, we already developed the training dataset which contains the information like persons who wear the facemasks correctly and persons who don't wear the facemask correctly as well as the persons who wear the facemask partially.

```
Validating runs/train/exp2/weights/best.pt...
Fusing layers...
Model summary: 213 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs
      Class      Images  Labels    P      R    mAP@.5  mAP@.5:.95: 100% 2/2
      all         120     477  0.963  0.806  0.857    0.612
      with_mask   120     392  0.973  0.903  0.961    0.684
      without_mask 120      76  0.933  0.737  0.833    0.499
      incorrect_mask 120      9  0.983  0.778  0.776    0.654
Results saved to runs/train/exp2
```

Figure 5.1: Training Result

Based on the results of the training, it can be seen that the level of minimized return is achieved when the graph starts to form an elbow (elbow method) in 50 epochs at mA@0.5, mAP@0.5:0.95, Precision, and Recall. The results can be seen in Figure 5.2 and 5.3

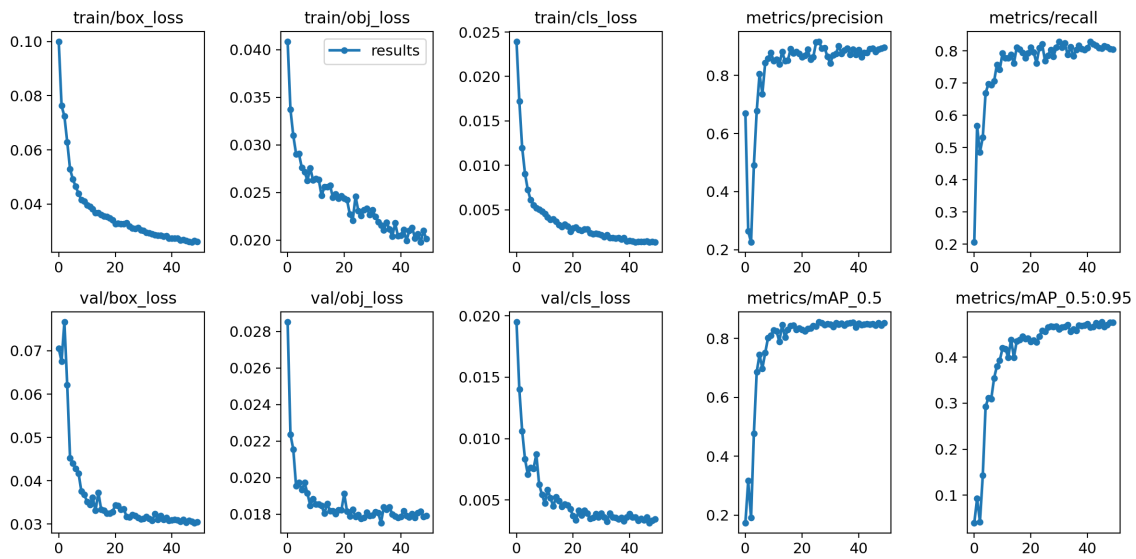


Figure 5.2: Overall Result

```

Model summary: 213 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs
val: Scanning '/content/train_data/labels/val.cache' images and labels... 120 found, 0 missing, 0
Class  Images  Labels  P      R      mAP@.5  mAP@.5:.95:  0% 0/4 |
Class  Images  Labels  P      R      mAP@.5  mAP@.5:.95:  25% 1/4 |
Class  Images  Labels  P      R      mAP@.5  mAP@.5:.95:  100% 4/4 |
  all      120     477    0.968  0.8    0.858    0.612
  with_mask 120     392    0.975  0.897  0.962    0.685
  without_mask 120     76     0.932  0.724  0.834    0.497
  incorrect_mask 120     9      0.998  0.778  0.776    0.654
Speed: 0.6ms pre-process, 17.5ms inference, 7.9ms NMS per image at shape (32, 3, 640, 640)
Results saved to runs/val/xxx
    
```

Figure 5.3: Testing Result

5.3 Performance Metrics

The classification report visualizer displays the precision, recall, F1, and mAP scores for the model. The metrics are defined in terms of true and false positives, and true and false negatives. Positive and negative in this case are generic names for the classes of a binary classification problem. In the example above, we would consider true and false occupied and true and false unoccupied. Therefore a true positive is when the actual class is positive as is the estimated class. A false positive is when the actual class is negative but the estimated class is positive. Using this terminology the metrics are defined as follows:

- Precision

Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class it is defined as the ratio of true positives to the sum of true and false positives.

Precision = Accuracy of positive predictions

$$Precision = TP / (TP + FP) \quad (5.1)$$

- Recall

Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives. Said another way, “for all instances that were actually positive, what percent was classified correctly?”

Recall = Fraction of positives that were correctly identified

$$Recall = TP / (TP + FN) \quad (5.2)$$

- f1 Score

The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. Generally speaking, F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.

f1Score = What percent of positive predictions were correct

$$f1Score = 2(Recall * precision) / Recall + Precision. \quad (5.3)$$

- mAP

mAP(Mean Average Precision) is a combined measure of recall and accuracy used to identify object accuracy. It is derived using a mean precision value for recall greater than 0 to 1 and IOU that is the intersection of the two values between 0.5 and 0.95.

Considering the precision of the developed model. Precision is a ratio between the number of positive results and the number of positive results predicted by the classifier. From the precision graph it can be said that model is performing well as the precision value increase overtime in the training process.

The below-given graph shows the recall value of the developed model. Here the recall is the ratio between the correct positive results and all samples founded to be positive.

The method used to measure the Mean Average Precision (mAP) on multiple object detection for the face mask detection is mAP@0.5 and mAP@0.5:0.95. The system on mAP@0.5 has a high level of precision. The highest value as can be seen in Figure 4.6 reached was 0.857

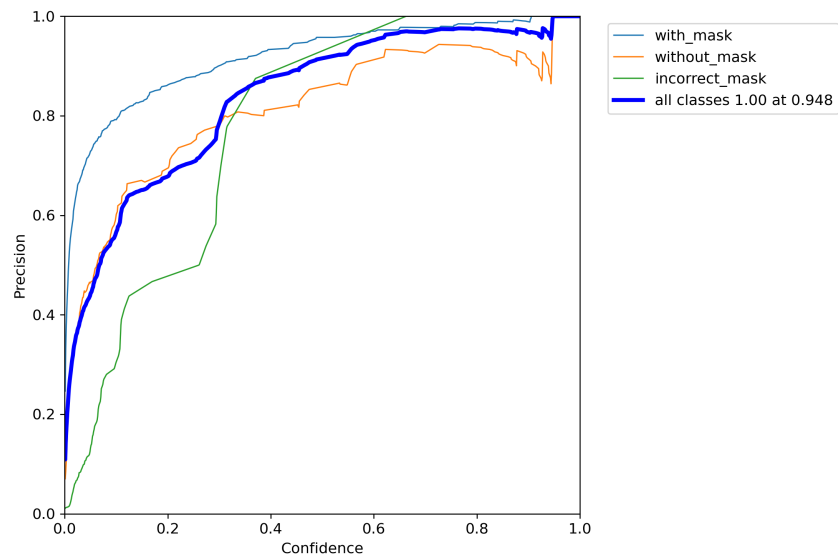


Figure 5.4: Precision of the developed model

The model has been tested using YOLOv5s. Training process has been completed by running the model through Google Colab (Virtual machine developed by Google which provides high RAM and GPU memory for executing machine learning programs). Using Google Colab was a necessity for me as my personal laptop is very old and does not have enough RAM and GPU memory to execute this model.

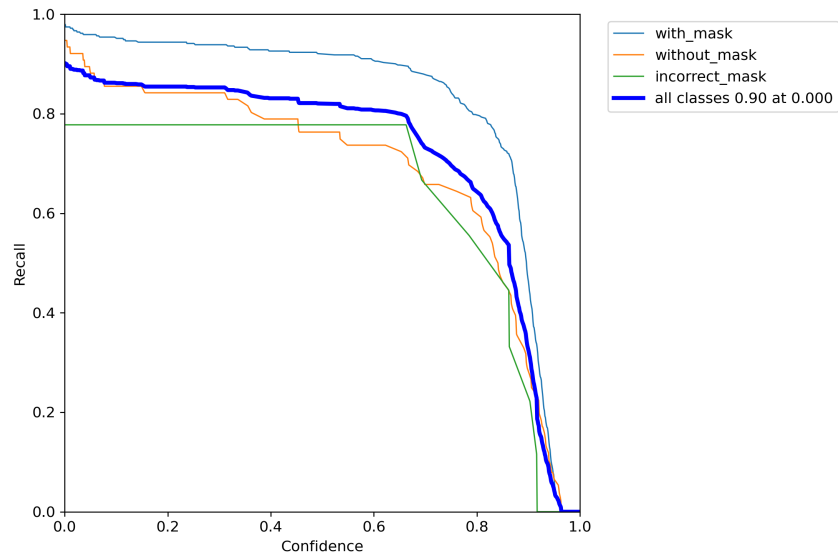


Figure 5.5: Recall graph of the developed model

5.4 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. It allows easy identification of confusion between classes e.g. one class is commonly mislabeled as the other. Most performance measures are computed from the confusion matrix. The important terms included in confusion matrix are as following:

- True Positive (TP) : Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.

Here, there were 2 class labels, that are representing the 2 stages of the churn as following:

- 0: No

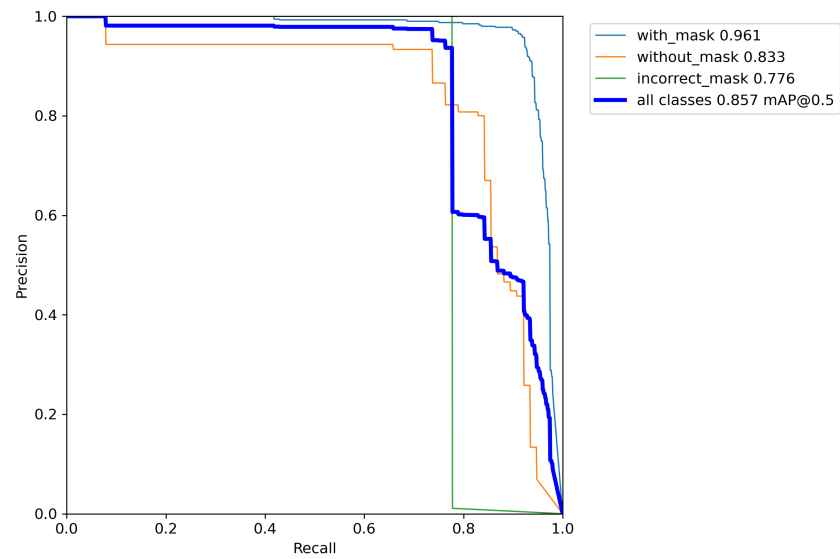


Figure 5.6: f1 Score graph of the developed model

- 1: Yes

The confusion matrix represent like this

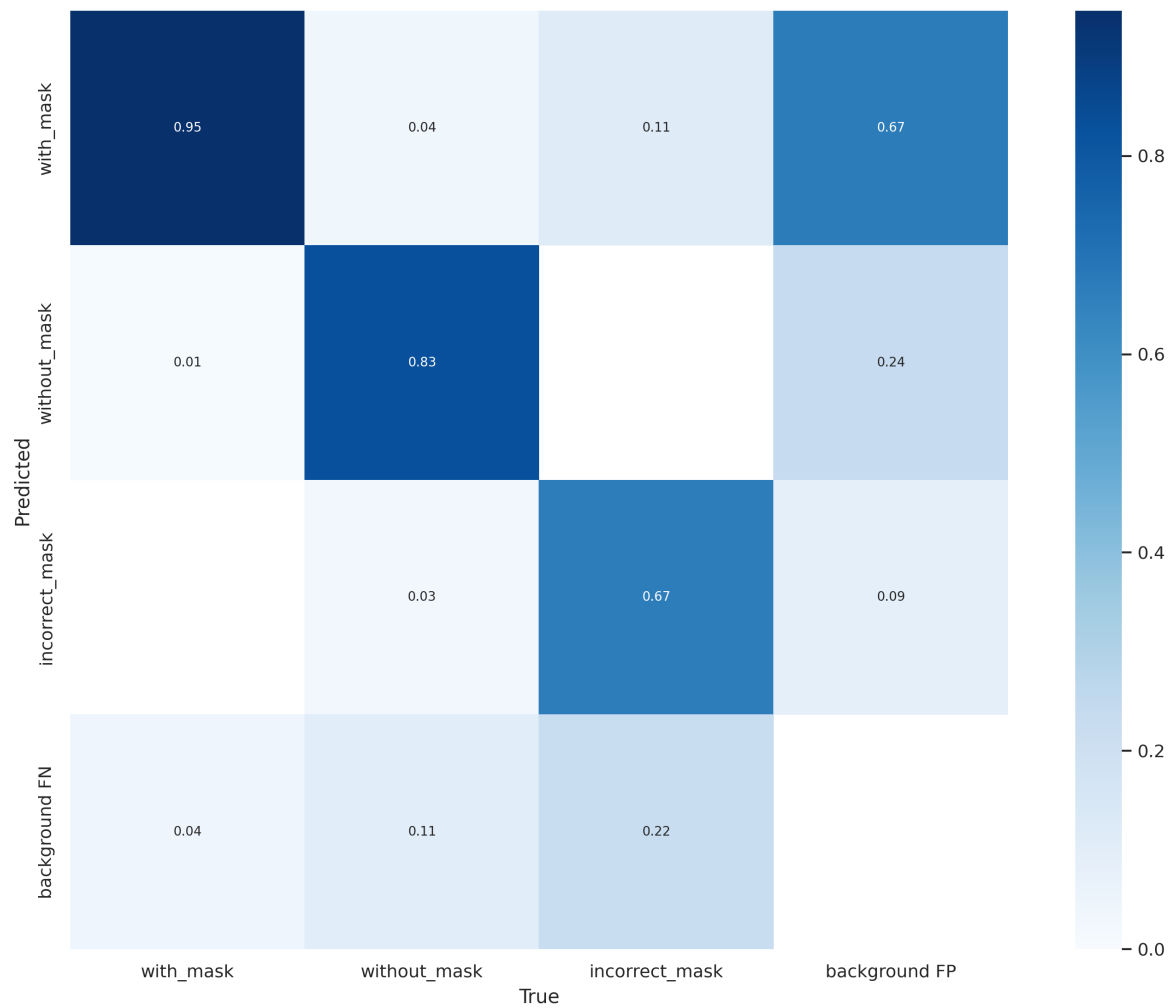


Figure 5.7: Confusion matrix

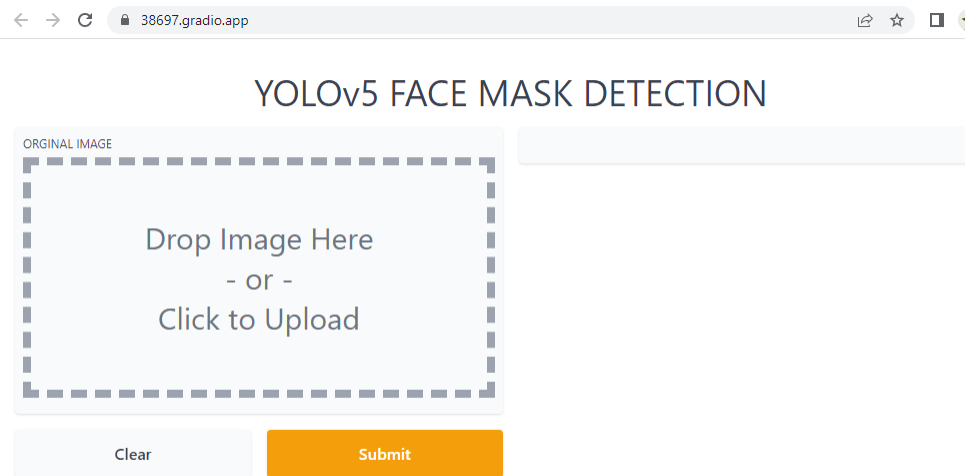


Figure 5.8: upload Image

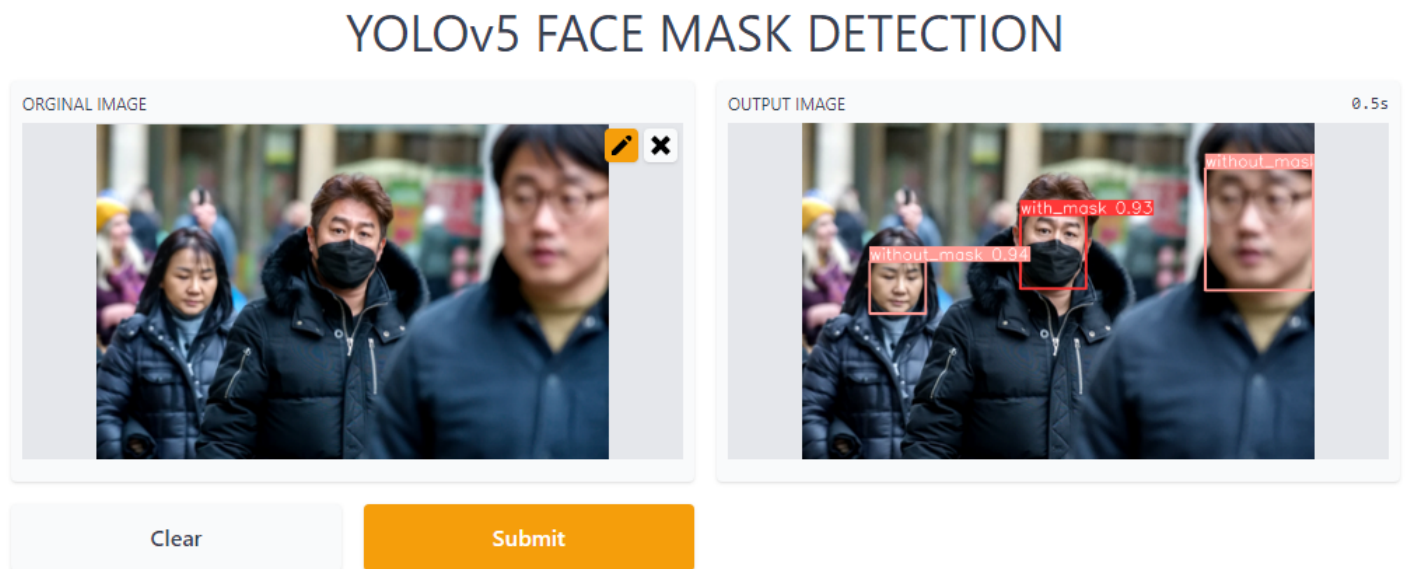


Figure 5.9: Detected Images of with and without Mask

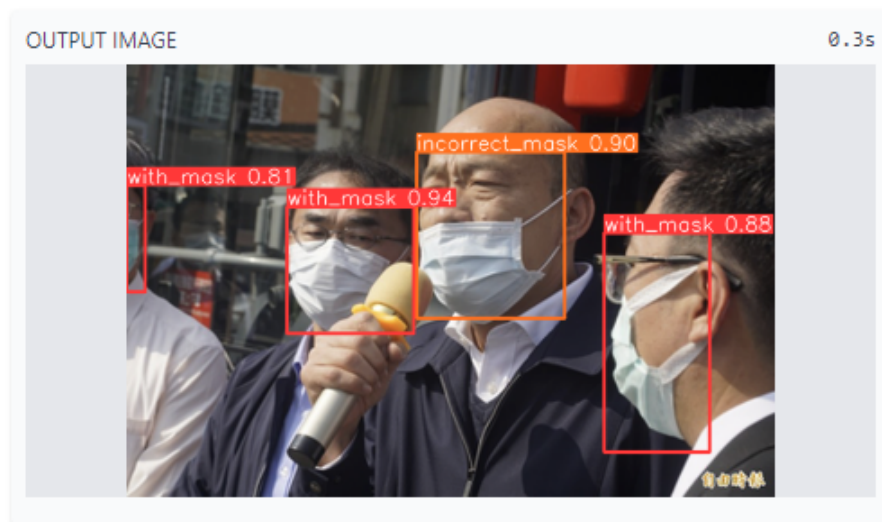


Figure 5.10: Detected Incorrect Mask

Chapter 6

Conclusion

As we know nowadays mask detection is a very challenging task. The applications of Facial Mask Detection are used especially for the prevention of spreading Corona Virus, tracking identifying criminals and anti-spoofing etc. By using a Deep Convolutional Neural Network Object Detection Algorithm, we can easily detect the face mask. The test results show a high accuracy rate in identifying individuals wearing a face mask, not wearing a face mask, and wearing a face mask but in an incorrect manner. The model was able to achieve a 85.7 % of Average Precision, which is a significant achievement. Moreover, the study presented a useful tool in fighting the spread of the COVID-19 disease by allowing all individuals to wear a face mask .

6.1 Advantages

The key Features are :

- The model accurately detect the person with mask, without mask and wearing mask in an incorrect manner.
- The system provide more accuracy than the existing system
- Faster Inference speed

6.2 Future Enhancement

The future works include alarm if somebody is not wearing a face mask properly, and detection of the social distancing. Apart from detecting the face mask, it will also compute the distances among each individual and see any possibility of coughing or sneezing. And improve the situation where masks covered by hands or other shielding objects cannot be recognized, making it more convenient for people to enter the shopping mall in this special environment, and the identification system will be more intelligent.

References

- [1] Goyal, H., Sidana, K., Singh, C. et al. A real time face mask detection system using convolutional neural network. *Multimed Tools Appl* 81, 14999–15015 (2022). <https://doi.org/10.1007/s11042-022-12166-x>.
- [2] Su, X., Gao, M., Ren, J. et al. Face mask detection and classification via deep transfer learning. *Multimed Tools Appl* 81, 4475–4494 (2022). <https://doi.org/10.1007/s11042-021-11772-5>.
- [3] Singh, S., Ahuja, U., Kumar, M. et al. Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment. *Multimed Tools Appl* 80, 19753–19768 (2021). <https://doi.org/10.1007/s11042-021-10711-8>
- [4] H. Adusumalli, D. Kalyani, R. K. Sri, M. Pratapteja and P. V. R. D. P. Rao, "Face Mask Detection Using OpenCV," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 2021, pp. 1304-1309, doi: 10.1109/ICICV50876.2021.9388375.
- [5] M. M. Rahman, M. M. H. Manik, M. M. Islam, S. Mahmud and J. -H. Kim, "An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network," 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2020, pp. 1-5, doi: 10.1109/IEMTRONICS51293.2020.9216386.
- [6] Mingjie Jiang, Xinqi Fan and Hang Yan "Retinal Face Mask Detector". 9 June 2020, Hong Kong University.

- [7] Chun, L. Z., Dian, L., Zhi, J. Y., Jing, W., Zhang, C. (2020). YOLOv3: Face detection in complex environments. *International Journal of Computational Intelligence Systems*, 13(1), 1153–1160. DOI 10.2991/ijcis.d.200805.002
- [8] Nguyen, N. D., Do, T., Ngo, T. D., Le, D. D. (2020). An evaluation of deep learning methods for small object detection. *Journal of Electrical and Computer Engineering*, 1–18. DOI 10.1155/2020/3189691.
- [9] Roy, B., Nandy, S., Ghosh, D., Dutta, D., Biswas, P. et al. (2020). MOXA: A deep learning based unmanned approach for real-time monitoring of people wearing medical masks. *Transactions of the Indian National Academy of Engineering*, 5(3), 509–518. DOI 10.1007/s41403-020-00157-z
- [10] Li S, Ning X, Yu L, Zhang L, Dong X, Shi Y, He W (2020) Multi-angle head pose classification when wearing the mask for face recognition under the covid-19 coronavirus epidemic. In: 2020 International conference on high performance big data and intelligent systems (HPBDIS). IEEE, pp 1–5. <https://doi.org/10.1109/HPBDIS49115.2020.9130585>.
- [11] Venkateswarlu IB, Kakarla J, Prakash S (2020) Face mask detection using mobilenet and global pooling block. In: 2020 IEEE 4th conference on information communication technology (CICT). IEEE, pp 1–5. <https://doi.org/10.1109/CICT51604.2020.9312083>.
- [12] C. Jagadeeswari and M. Uday Theja, “Performance Evaluation of Intelligent Face Mask Detection System with Various Deep Learning Classifiers”. *International Journal of Advanced Science and Technology*. Vol 29, 2020, P P 3074-3082
- [13] Qin B, Li D (2020) Identifying facemask-wearing condition using image super-resolution with classification network to prevent covid-19. *Sensors* 20(18):5236. <https://doi.org/10.3390/s20185236> .
- [14] Toshnall Meenpal, Ashuthosh balakrishnan and AMit Verma. “Facial Mask Detection using Semantic Segmentation”. 2019, IEEE 4th International Conference on Computing, Communication and Security.

- [15] Md. Sabbir Ejaz, Md. Sifatullah, Md. Rabiul islam and Ananya Sarker “Implementation of Principle Component Analysis on Masked and Non-masked Face Recognition”. 2019, IEEE 1st International Conference on Advances in Science, Engineering.