International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Deep Learning-Based Covid-19 Face Mask Detection Implementation in Matlab

*Upasana Sonkar *[a] and Shankar Sharan Tripathi [b]*

[a] *M. Tech Research Scholar in Computer Science Engineering, Shri Shankarcharya Technical Campus Bhilai, India*
[b] *Asst. Professor, Computer Science Engineering, Shri Shankarcharya Technical Campus Bhilai India*
[a] *upasanasonkar75@gmail.com*, [b]*triapathi.shankar@gmail.com*

ABSTRACT

The World Health Organization has imposed various restrictions to control the spread of the corona virus, including wearing masks and maintaining social distance. In the midst of such a severe crisis, this project can be used in social gatherings to find masks to help reduce the spread of Covid-19. This paper proposed a face mask detection algorithm to obtain static wear based on improved YOLO to solve low accuracy, low real-time performance, instability and other problems of complex area. Corona face mask provides a real-time measure of safety for people by finding out whether a person is wearing a mask or not as wearing a mask is an essential requirement of the hour for the COVID-19 epidemic. Finally, the data set for finding a face mask is done according to the standard wear of the mask. Based on a deep learning algorithm, the various test indicators are compared to evaluate the effectiveness of the model. This thesis presented a study on face recognition and mask detection through deep learning techniques by developing YOLO and SSD model using machine learning techniques. Although half of the faces were covered by masks, this process provided accurate and fast results for facial recognition security systems. Test results show a high accuracy rate in identifying masked and unmasked person. This model was able to achieve a performance accuracy of 71.6% using YOLO and performance accuracy of 97.6% using SSD, which is a remarkable achievement. Also, the study proposed a useful tool to combat the spread of the Covid-19 disease by allowing all individuals to wear face masks while performing biometric authentication.

Keywords: Deep Learning, COVID - 19, Face Mask Detection, Mean Average Precision Recall, ImageLabeler toolbox Computer Vision. Using YOLO and SSD.

1. INTRODUCTION

Mask use has been more common in tandem with the Covid-19 pandemic's global outbreak. Scientists have established that wearing masks helps stop the spread of Covid-19. Corona virus, also known as Covid-19, is a recent pandemic that has disrupted human existence during the past century. To combat the Covid-19 epidemic, the potential of AI is being used to improve knowledge of infection rates and speed up diagnosis. Here we present a deep learning based face-to-face identification model. In addition, this proposed work could be installed with security cameras so that the spread of Covid-19 can be prevented by identifying people with masks with those without masks. This model is a combination of machine learning with Yolo and SSD deep learning methods. We also provide a comparative study between both methods such that the an efficient algorithm can be found which takes less time on the training and acquisition process and also achieves highest accuracy. In 6 months, more than five million cases of Covid-19 were reported in 188 countries [1]. In 2020, considering the vast widespread of Covid-19 virus, World Health Organization was forced to declare this outbreak as a global pandemic. Due to close contact, this virus spreads majorly at the congested places. The outbreak of the corona virus has led to a tremendous speed of global scientific collaboration. Using deep learning and machine learning based Artificial intelligence (AI), Covid-19 can be fought in various ways. Clinicians and researchers are able to mine large datasets using Machine learning to predict the spread of Covid-19, act as a precaution against potential disease, and identify people at risk. Healthcare services need new technologies like Artificial Intelligence, Machine

Learning, Big Data and Internet of Things (IoT) to address new disease predictions. In many areas, wearing a mask in public is a legal requirement. These regulations and laws were put in place due to the rapidly increasing death toll in many places. However, monitoring large crowds has become progressively challenging. The surveillance process includes identifying people whether they are wearing masks or not. This proposed work combines deep learning with the classic Yolo and SSD machine learning models. We use the deep transmission layer to extract features and integrate them with classic machine learning algorithms

II. LITERATURE REVIEW

Numerous literatures have been reviewed for the proposed methodology. These literatures are published in various journals over the period of 2004 to 2021. These literatures are based on Deep Learning algorithms like YOLO method, SSD method, ResNet50, Mobile Net algorithm, etc. These literatures helped us to propose models which can produce high accuracy over a short period of time.

E. Zhang [11] has made great strides in getting a face mask using MobileNetV2. On paper, the MobilenetV2 mobile learning algorithm is used to identify the face mask. This paper provides steps to model the data collection, pre-processing, data classification, model testing, and modeling use. The proposed model can achieve 96.85% accuracy.

Jiang, Tianhan, Zichen & Yukang [12] proposed a Face Mask Discovery program using YOLOv3 and fast R-CNN models. The paper tied a binding box in red or green to people on the screen, whether or not they wore masks, and each day tracked the count of masked people.

Zhai, Shang, Wang & Dong [13] tried to obtain a face mask using deep learning research. In order to achieve higher accuracy and lower cutting time, the proposed process uses single and double stage detectors. ResNet50 and reading transfer concept to integrate advanced semantic information used in this paper. During the discovery of the mask, in order to improve the performance of the area, modification of the binding box is applied. The three most popular models; ResNet50, AlexNet and MobileNet are used for model testing. Proposed and these models can produce high accuracy over a short period of time. The proposed method achieved accuracy of 98.2% while using ResNet50 model. Model proposed by this research is best suited for video surveillance devices.

III. METHODOLOGY

Different literatures analyzed earlier in section 2 gives clarity that we should propose a model which uses the concept of deep learning in YOLO and SSD methods to achieve higher accuracy. Model should also provide quality and quantitative analysis for the model using average precision (AP), accuracy, precision and recall score. The aim of face mask detection is to distinguish person wearing masks with those who are not wearing masks. Firstly we need to design a system which is based on real-time MATLAB GUI. Then implement deep learning algorithms (YOLO and SSD) to detect face mask.

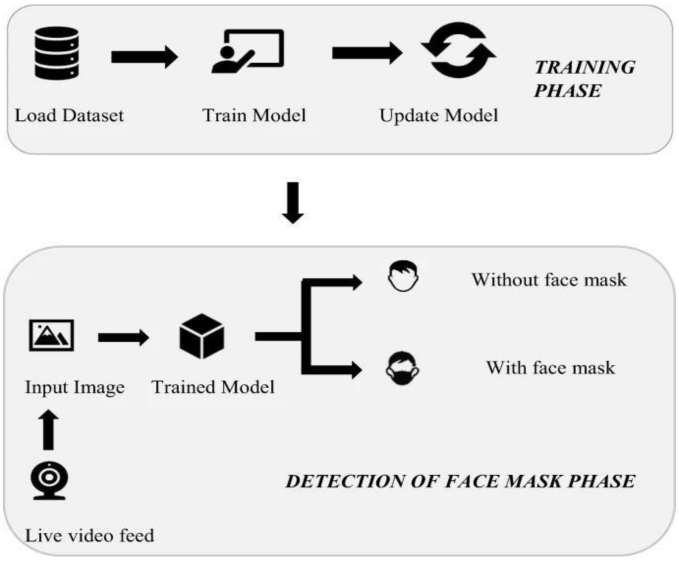


Figure: 3.1 - Phases of proposed Methodology

3.1 Proposed Methodology

Training phase is supposed to be first phase in this system. Training phase starts by collecting data. Foremost important step is having best quality and value of data. We can edit the dataset or else can utilize previous available datasets from different origins. In addition, prior to its use, data cleaning is required because it is an important key of data modeling. The different data cleaning steps are depicted in Figure 4.2. Once a high quality database is received, particular model is chosen according to the requirements. Then it is trained on the selected database. Many strategies can be used to achieve the goal.

The first phase ends with the search for the right professional model. In second phase, live video feeds or frames of images are provided as input for the trained model. Using cell phone or surveillance camera, live video feeds can be accessed and based on the device, resulting format may vary. Figure 3.2 describes the steps involved in both the phases of the proposed methodology.

3.2 Real Time Image Acquisition

We collect the Data from Webcam of PC or Laptop using Video Processing techniques [24]. Here after collecting the video, the video are extracted in image frames and per frame is analyzed by YOLO and SSD object detection methods.

3.3 Model Data Training

The database is created using images obtained from different sources. Using data enrichment techniques, size of the datasets can be increased. Images are stored separately in two files, named as "training dataset" and "testing dataset", with respectively 80% and 20% images. Integrated boxes also called as "data anchors", are built in an interesting context using a variety of methods. In this proposed scheme, images are labeled as either "Mask" or "No Mask" using a tool named LabelImg (Image Labeler).

3.4 Image Labeler Matlab Toolbox

For circular labels, various shapes can be marked as Region of Interest (ROI) using the Image Labeler app which is an effortless way to co-create it. Rectangular, polygon, polyline and pixel ROI labels and also visual labels can be created on an image or photo sequence. To train or validate algorithms, labeled data can be used as object detectors, image classifiers and semantic segmentation networks. This application should be used to mark ROI labels while selecting a labeling tool. In order to select the appropriate drawing tool and label type for the application, refer the Supplemental Labels, ROI Labels, and Qualifiers.

The Image Labeler application aids to label a set of images as the ground truth data (Figure 3.3). With the use of this application, following could be done:

Describe Region of Interest (ROI) using polyline ROI labels, rectangle labels, square labels, polygon ROI labels, and pixel ROI labels. These labels can be used to create the ground truth data.For labeling the ground truths, use the in-built monitoring or detection. In order to label the ground truth; we can use, import or write your own algorithm. Refer Creating Automation Algorithm. Using scene summarization, the efficiency of automated algorithms by label can be evaluated. Refer Summary of Low Truth Labels. Extract the ground truth data named as a ground truth item. Use this item to validate or train an object detector or semantic segmentation network.

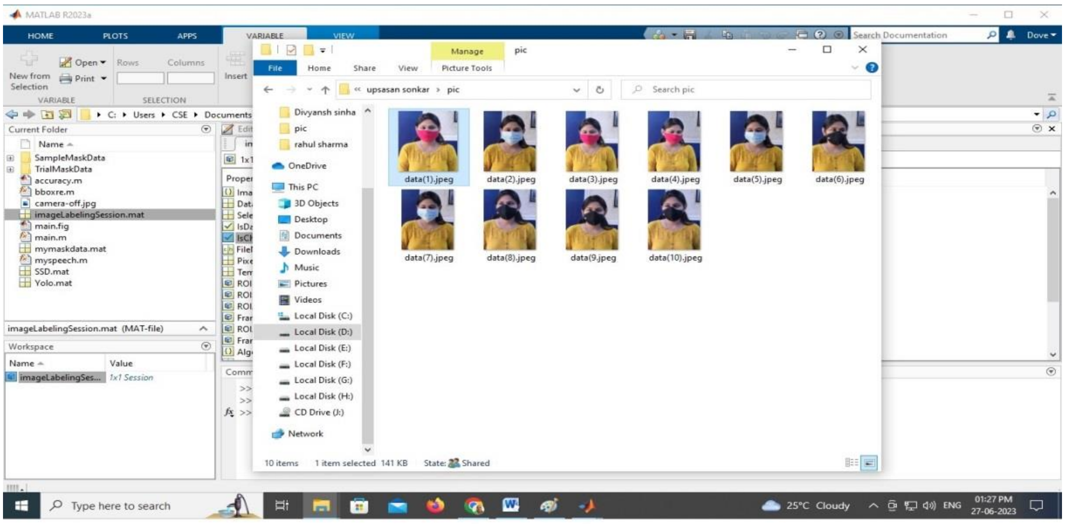


Figure: 3.3 - Sample of Data for Image Labeler Application

All image file formats is supported by the Image Labeler application. Additionally it supports the potentiality to upload multi-frame data by the imRead function and also supports the Digital Imaging and Communication in Medicine (DICOM) file format, (Figure 3.4). We can build an imageDatastore to read more types of file formats and use the ReadFcn feature on it.

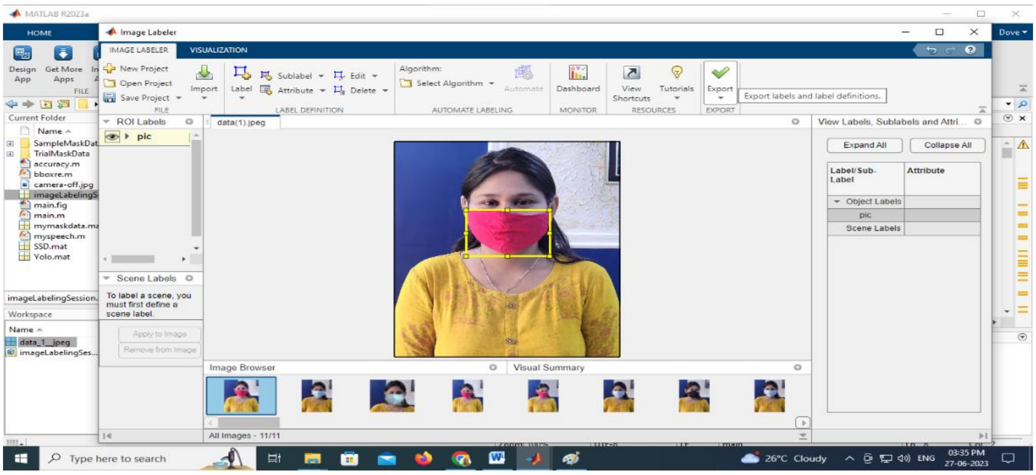


Figure: 3.4 – ImageLabeler toolbox in MATAB application

While uploading a photo, if the size of the image is more than 8000 pixels or image resolution, the imageLabeler app provides the choice to convert that image into a block image. A block image contains a large image segmented into smaller blocks which can be stored in memory. ImageLabel converts a large image into a block, which can be processed in the app just like any normal image. Blocked Images helps to process those images in the application whose processing was difficult due to various limitations. Refer the LargeImages label under photo label for more information on this.

3.5 Deep Learning

A dynamic set of learning techniques using neural networks is called deep learning. Neural networks are biologically inspired paradigm that allows computers to read data. Deep learning is a subset of machine learning that imitates human brain to gain knowledge such as learning by example [8].

Deep learning is the prime technique for working of non-motorized vehicles, helping them recognize stop signs or distinguish pedestrians from traffic lights. This is mandatory in voice assisted devices like TVs, tablets, cell phones, speaker phones, etc. Recently, deep learning got great deal of recognition. We will reap benefits like never before.

Deep learning teaches computer models to perform tasks that directly distinguish between text, sound or images. Deep learning models can attain cutting-edge accuracy and sometimes surpass even the efficiency level of humans. The proposed work is trained with a massive labeled data along with multiple layers of neural network architecture as shown in Figure3.5

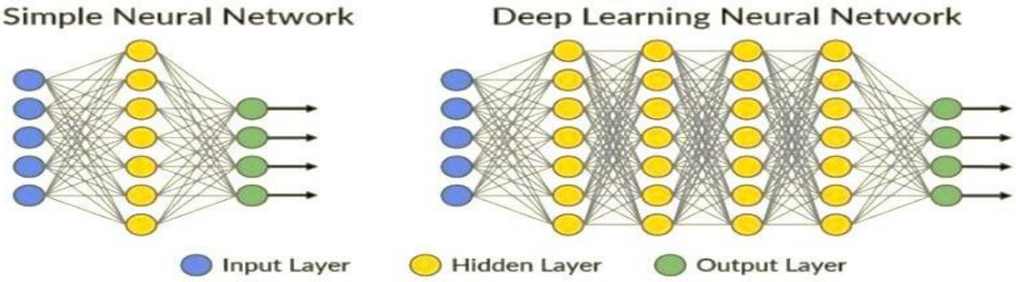
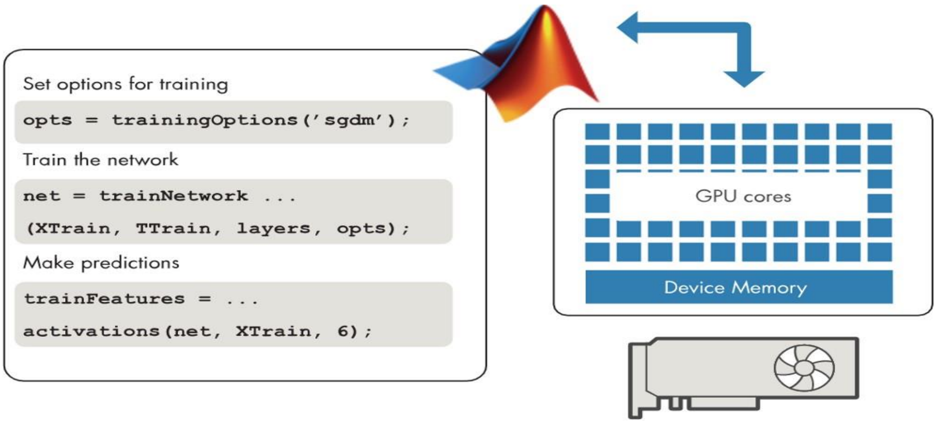


Figure: 3.5 - Deep Learning Model Architecture

3.6 Accelerating Deep Learning Models with GPU's

To train a model of deep learning requires prolonged time, which might take days or even weeks. Providing acceleration will significantly make the process faster with the use of Graphics Processing Unit (GPU). If GPU is used along with MATLAB, time will be reduced for training a network thus eventually helps in reducing the time required for training of image segmentation and can reduce from weeks to days also up to hours. While deep learning model is trained, GPU (when available) is used by MATLAB even by not knowing the GPU organization.



```
opts = trainingOptions('sgdm');
net = trainNetwork ...
(XTrain, TTrain, layers, opts);
trainFeatures = ...
activations(net, XTrain, 6);
```

Figure: 3.6 - Deep learning in Matlab

Figure 3.6 uses toolbox of deep learning to train CNN from scratch.

3.7 Applications of Deep Learning

Varied applications in which deep learning networks can be used [9] are natural language processing, visual recognition, self-driving cars, fraud detection, Google's virtual assistant, detecting delay in the development of a child, healthcare, text into image translation, automated image recognition, earthquake forecasting, providing sound effects in silent movies, image to image synthesis, market-rate prediction, image colorization, news summation and many more.

3.9 YOLO Method

YOLO-v4 proposed in 2020 is a real-time with high-precision regression based object detection method which uses only single stage. It has attained optimality with respect to trade-off of detection accuracy and detection speed. Figure 4.7 shows the structure of this model. This model is divided into three sections: Backbone, Neck, and Prediction. YOLO-v3 merged the residual module with itself and integrated the properties of the ResNet structure to obtain Darknet53. Then YOLO-v4 designed the CSPDarkNet53 based on Darknet53 by taking into account of deep learning feature of CSPNet (Cross Stage Partial Network) [14].

The feature layer is fed as input in the residual module and higher-level feature information is obtained at the output. Therefore stimulating the residual learning by additionally reducing the model parameters and strengthening the feature learning. SPPNet (Scale Selection Pyramid Network) and PANet(Path Aggregation Network) together forms the Neck.

In SPPNet, first of all, three times the feature layer gets convolved, and then using the different sized maximum pooling cores, the feature layer as given in input is maximally pooled. In the first place, concatenation of the pooled results is done and then it is later convolved three times, thereby increasing receptive field of the network. Once the operation of Backbone and SPPNet is finished, PANet convolves the feature layers. Further PANet up-samples it. This is done by doubling the width and height in the original feature layers. Then after convolution it concatenates the feature layers. In order to realize feature fusion, it up-samples the feature layers received from CSPDarkNet53. Later down-sampling is done by compressing the width and height, and then for realizing feature fusion, stacking (five times) the previously obtained feature layers is done finally.

The Prediction module predicts by utilizing the features which were obtained from the network. For example, take a 13 × 13 grid, and subdivide image to grids of 13 × 13. Finally preset every grid using 3 previous frames. Results adjust position of 3 previous frames. Then lastly, to attain the final prediction frame, further filter the image using NMS (Non-Maximum Suppression) method.

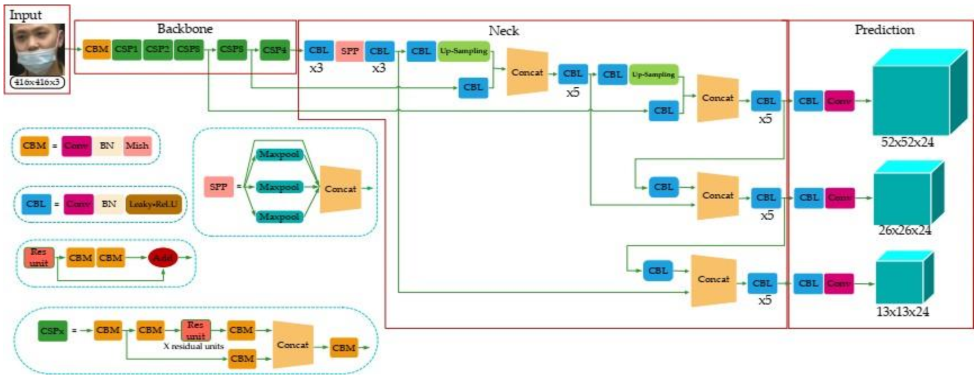


Figure: 3.7 - Yolov2 Framework **in Deep Learning**

YOLO predicts multiple connecting boxes per cell grid. To calculate the actual loss, we only need one of them to be responsible for the item. For this purpose, we prefer the one with the highest IoU (intersection over union) with a basic truth. This strategy leads to a choice between the predictive box predictions. Each forecast is better at predicting specific sizes.

Loss Function

YOLO uses a double-digit error between speculation and basic truth to calculate losses. Loss of activity includes:

Loss in stages.

Loss of localization (errors between predictive boundary box and basic fact).

Loss of confidence (box resistance).

Classification Loss

When an object is detected, a class loss in each cell is a square error of the class conditions of each class:

$$\sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \qquad \text{(Eq 4.1)}$$

where

$\mathbb{1}_i^{obj} = 1$ if an object appears in cell i, otherwise 0.

$\hat{p}_i(c)$ denotes the conditional class probability for class c in cell i.

3.10 SSD Method

SSD is acronym for Single Shot Multibox Detector. It is an approach which uses deep neural network for detecting objects in an image. SSD method is usually applied for detecting object in an image. SSD method uses basic VGG-16 Architecture, and thus it also performs with both speed and accuracy like other algorithms of object detection. Figure 3.8 depicts the architecture of SSD method.

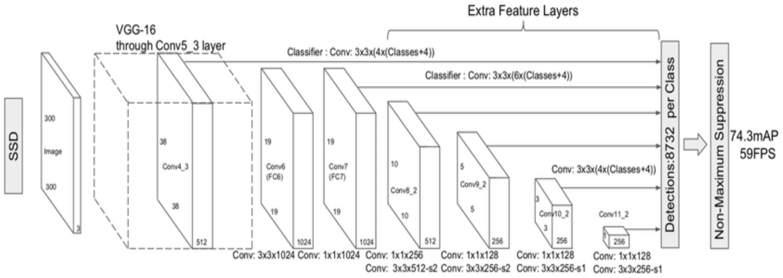


Figure: 3.8 - SSD Architecture in Deep Learning

SSD method works on a feed-forward convolutional network. It constructs equalized binding boxes and estimates the presence of phase objects in these boxes, then to obtain the final detection it is followed by a small compression step. For a high quality image classification, the original network layers are based on standard formats, referred as the basic network. Then an auxiliary structure is added to the network in order to detect objects with key features as mentioned: Feature maps have multi discoverable dimensions. At the end of the reduced base network, just add multiple layers of convolutional features. In Convolutional Predictive Predictions, each extra feature layer generates a fixed set of detection predictions with the use of a set of transform filters. Figure 4.8 shows this on the SSD network structure. For p-channels and m×n size factor, the basic function of prediction of the acquisition parameters is possible with a small 3×3×p category generating kernel or layout associated with the standard box connections. It produces output value 1 for each m×n region wherever the kernel is used. The output value of the binding box is measured with respect to the default value.

IV. RESULTS AND DISCUSSION

In this section we will analyze on the how proposed model is designed to identify people wearing masks. We will take a look on the steps to design GUI for the proposed model and will analyze concept of deep learning in YOLO and SSD methods to detect the face mask. We will also assess the performance and accuracy of the model using various quality parameters and provide a comparative analysis for YOLO and SSD method.

Steps for implementing the proposed model

4.1 Design Matlab GUI

We begin the preparation phase by dissecting the convolutional neural system of defects and then sending it to the MATLAB workspace for correction. Using a system diagnostic tool, we get a complete and concise view of the neural system and with these lines the layers of the layer equally tell any warnings and errors. We will do it on that point trades the program and proceed to the preparation stage using the MATLAB code.

4.2 Connect Webcam

In MATLAB GUI we make a button for easy user interface. Here user can choose and start webcam of PC or Laptop. We use camlist = webcamlist; for checking the list of webcams available. After that, choose the camera list number.

We start the webcam video and after that preview gets started (Figure 4.1). vid= videoinput('winvideo', cameraListNo);

Preview (vid,hImage);

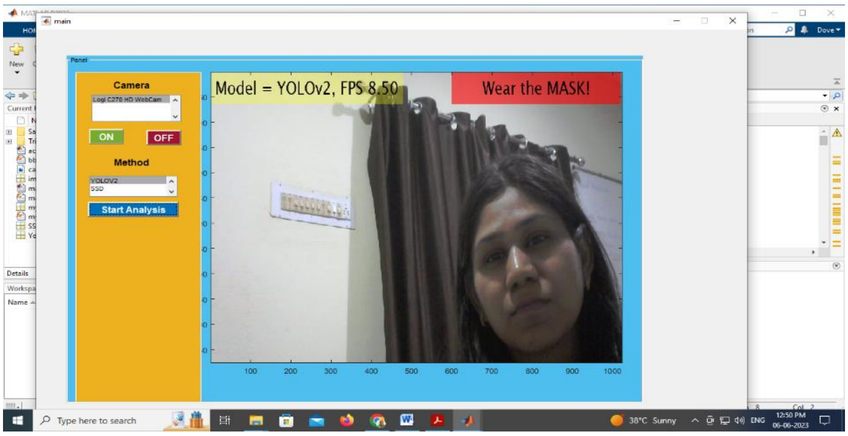
4.3 Video Processing

It is a form of processing a signal, specifically processing of the image with video streams or files as input and output. Import webcam videos to convert to frames into the current MATLAB environment.

Before execution, create an empty directory named frames.

Get the total number of frames in the video

Run a for loop and start capturing the frames into the director



Camera: Logi C270 HD WebCam

Method: YOLOV2

Model = YOLOv2, FPS 8.50

Wear the MASK!

Figure: 4.1 - MATLAB GUI for Real-time Face mask Detection

4.4 Select the Analysis Method

The set of images are now labeled as “Mask” and “No Mask”. The images may be of different resolutions and sizes, perhaps extracted from different sources or different resolution cameras.

Resize input image to 512 x 512

Apply color filtering (RGB) to channels

Scale / normalize the image using the default average of the embedded weights

Center the image with 224 x 224 x 3 pixel value

Finally, turn it into a transformation.

4.5 Start to Detect Face Mask

After the choosing Object detection Method: Yolo or SSD, we can easily tract the face. If Person wear the mask then output will be “Great! Be safe” (Figure 4.2). If person not wear mask then it makes alert “Wear the Mask” (Figure 4.3).

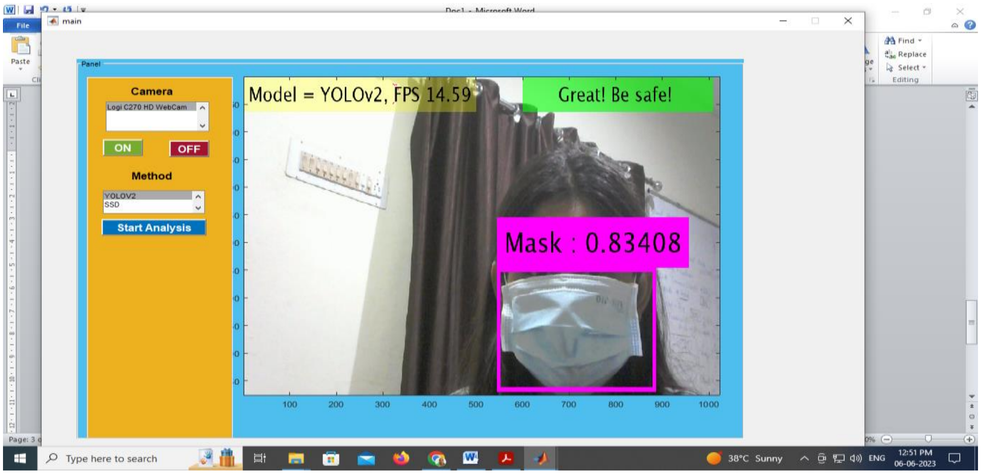


Figure: 4.2 - Real time Face Mask detection of a Person

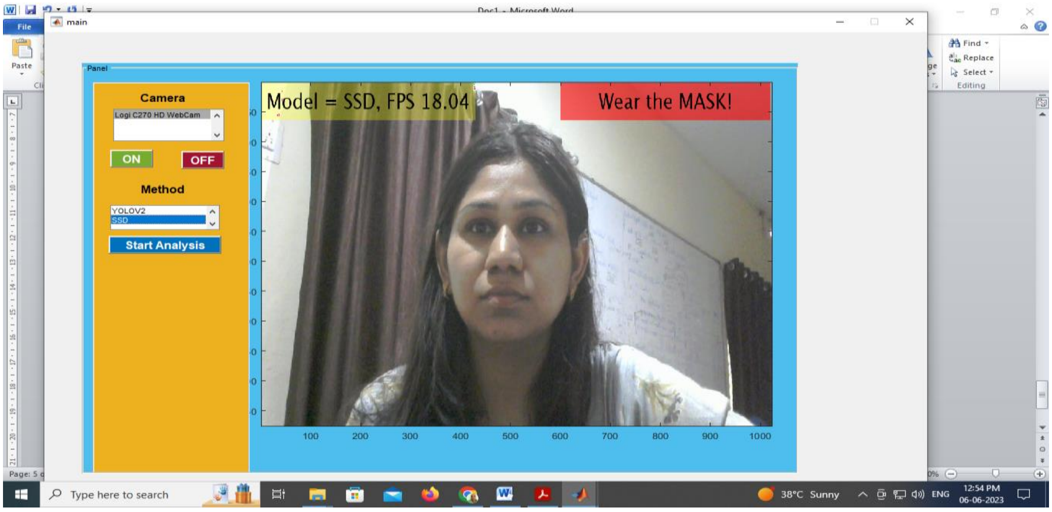


Figure: 4.3 - Real time detection of No Face Mask availability for a Person

4.6 Experimental Output

Predict the processed input images using prebuilt models. The video frame is also labeled whether the person is wearing the mask or not along with the percentage of prediction. Below are some examples of using this model at runtime with different persons. Figure shows Face Mask Available in person 1. Figure 4.4 shows Face Mask Not Available in person 1.

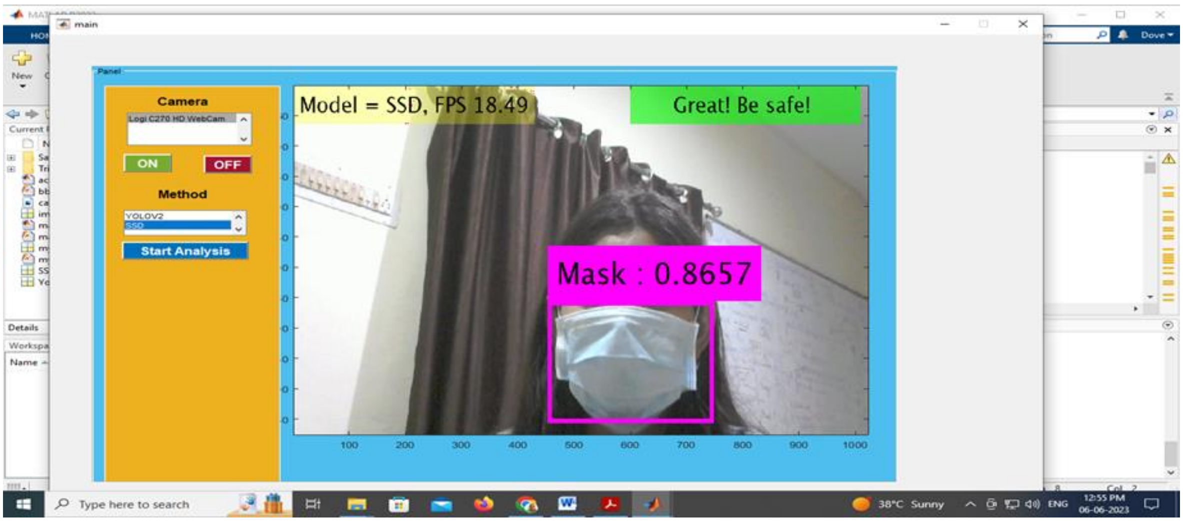


Figure: 4.4 - Face Mask Available in Person 1

4.7 Performance Evaluation

In this module, we evaluate performance of the model based on below four quality feature parameters for the image quality using both YOLO and SSD methods.

4.8 MSE - Mean Squared Error

MSE is a distortion measure. It means the difference between referenced image and processed image. Higher MSE value means larger difference between referenced images with respect to processed image. This means that lower value of the MSE value signifies that it is better; and 0 being the perfect model.

4.9 PSNR - Peak Signal to Noise Ratio

PSNR is the measure of reconstruction quality. It is the ratio of max signal strength with respect to the corrupt noise signal strength. Typically its value for a video and image compression ranges from 30 to 50 dB, for bit depth such as 8 bits, with the higher the better.

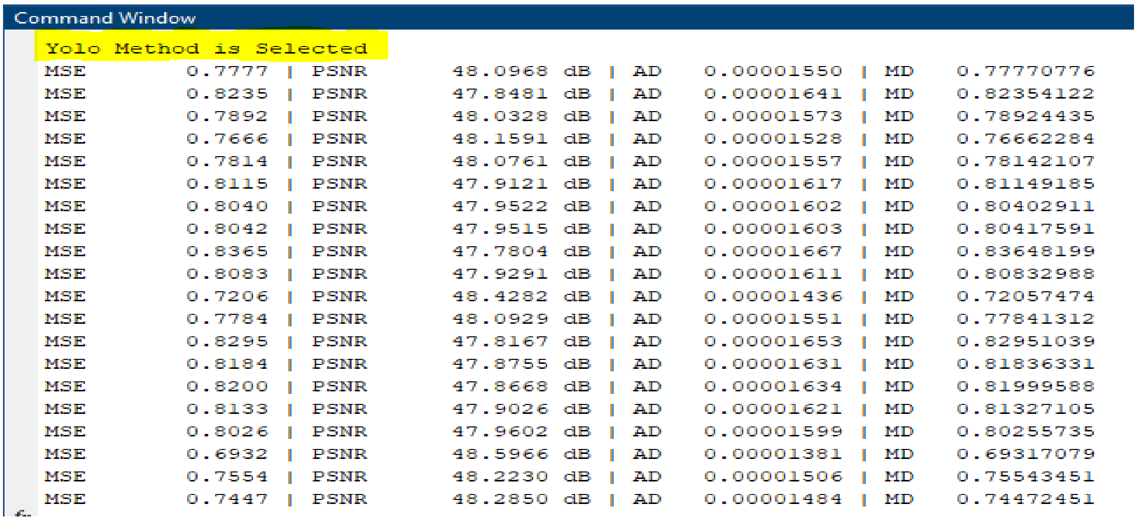
4.10 AD - Average Difference

Average difference is defined as the difference in the pixels between the filtered image and its respective degraded image. A larger value of Average Difference indicates the poor image quality.

4.11 MD - Maximum Difference

Maximum Difference is directly proportional to the contrast giving the dynamic range of an image. The higher Maximum Difference value specifies the poor quality of image.

Result of Quality factors with YOLO Method



```
Yolo Method is Selected
MSE       0.7777 | PSNR     48.0968 dB | AD   0.00001550 | MD   0.77770776
MSE       0.8235 | PSNR     47.8481 dB | AD   0.00001641 | MD   0.82354122
MSE       0.7892 | PSNR     48.0328 dB | AD   0.00001573 | MD   0.78924435
MSE       0.7666 | PSNR     48.1591 dB | AD   0.00001528 | MD   0.76662284
MSE       0.7814 | PSNR     48.0761 dB | AD   0.00001557 | MD   0.78142107
MSE       0.8115 | PSNR     47.9121 dB | AD   0.00001617 | MD   0.81149185
MSE       0.8040 | PSNR     47.9522 dB | AD   0.00001602 | MD   0.80402911
MSE       0.8042 | PSNR     47.9515 dB | AD   0.00001603 | MD   0.80417591
MSE       0.8365 | PSNR     47.7804 dB | AD   0.00001667 | MD   0.83648199
MSE       0.8083 | PSNR     47.9291 dB | AD   0.00001611 | MD   0.80832988
MSE       0.7206 | PSNR     48.4282 dB | AD   0.00001436 | MD   0.72057474
MSE       0.7784 | PSNR     48.0929 dB | AD   0.00001551 | MD   0.77841312
MSE       0.8295 | PSNR     47.8167 dB | AD   0.00001653 | MD   0.82951039
MSE       0.8184 | PSNR     47.8755 dB | AD   0.00001631 | MD   0.81836331
MSE       0.8200 | PSNR     47.8668 dB | AD   0.00001634 | MD   0.81999588
MSE       0.8133 | PSNR     47.9026 dB | AD   0.00001621 | MD   0.81327105
MSE       0.8026 | PSNR     47.9602 dB | AD   0.00001599 | MD   0.80255735
MSE       0.6932 | PSNR     48.5966 dB | AD   0.00001381 | MD   0.69317079
MSE       0.7554 | PSNR     48.2230 dB | AD   0.00001506 | MD   0.75543451
MSE       0.7447 | PSNR     48.2850 dB | AD   0.00001484 | MD   0.74472451
```

Figure: 4.6– Runtime values of Quality Parameters using YOLO method

With our runtime result for YOLO method (Figure 4.6), we found that (Table 4.1):

Table: 4.1 – Quality Parameter values using YOLO Method

S No.	Quality Parameter	Result (approx)
1	MSE	0.7 to 0.8
2	PSNR	47 to 48 dB
3	AD	0.00001
4	MD	0.7 to 0.8

All the above quality parameter values using YOLO method are within the best desired values as stated earlier in section 5.3. We see that PSNR value is high, so noise value is low. Thus our system is working with low noise margin.

Result of Quality factors with SSD Method

With our runtime result for SSD method , we found that (Table 4.2):

Table: 4.2 – Quality Parameter values using SSD Method

S No.	Quality Parameter	Result (approx)
1	MSE	0.6 to 0.7
2	PSNR	47 to 48 dB
3	AD	0.00001
4	MD	0.6 to 0.7

All the above quality feature values using SSD method are within the best desired values as stated earlier in 5.3. In this method also, we found that PSNR value is high, so noise value is low. Thus our system is working with low noise margin for SSD method also.

4.12 Accuracy Calculation

Statistically, accuracy is a measure of how well a classification test rules out or correctly identifies a condition. Generally, accuracy can be defined as the ratio of correct predictions (including both true positives and true negatives) to the total number of cases studied [28]. The formula for calculating accuracy is:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad \text{(Eq 4.1)}$$

TP means True positive; TN means True negative; FN means False negative; FP means False positive

While choosing best method, not only accuracy but other metrics like Precision, Recall and Average Precision can also be used.

Precision

Precision (P) is the proportion between true positives count (Tp) with respect to summation of true positives count (Tp) and false positives count (Fp). [29]

$$P = \frac{T_p}{T_p+F_p} \qquad \text{(Eq 4.2)}$$

Recall

Recall (R) is the proportion between true positives count (Tp) with respect to summation of true positives count (Tp) and false negatives count (Fn). [29]

Average Precision

$$R = \frac{T_p}{T_p+F_n} \qquad \text{(Eq 4.3)}$$

Average Precision (AP) is the summation of the weighted mean of achieved precisions at each threshold level, in such a way that as the recall value increases from the previous threshold, it is assumed as the weight. [29]

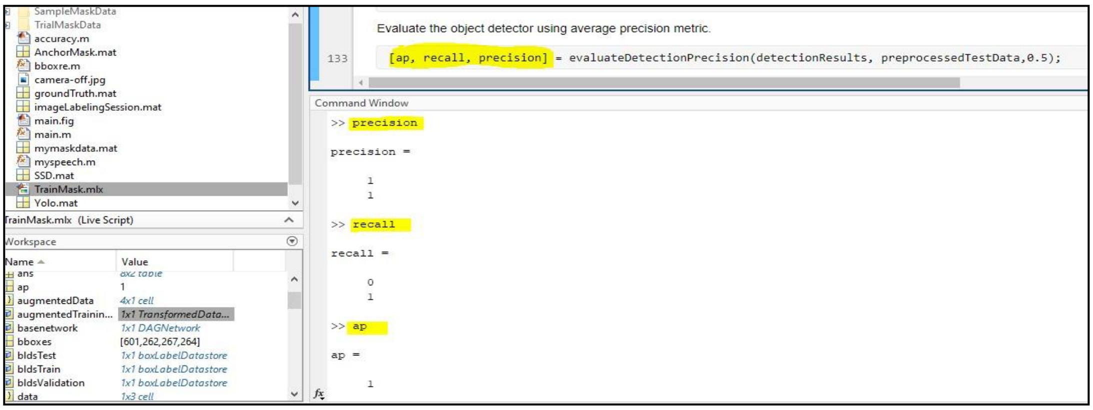
$$\text{AP} = \sum_n (R_n - R_{n-1}) P_n \qquad \text{(Eq 4.4)}$$

Where Pn is precision value and Rn is recall values at the nth threshold.

4.13 Accuracy achieved with YOLO method

Model multiple times reads the same image and the augmented training data is displayed. We can verify accuracy by inspecting training data loss for each of the iteration (Table5.3). Using YOLO method, our model gives Accuracy of 71.6%.

For all the test images, we need to run the detector for evaluating the object detector while using Average Precision metric. We got the below (Figure 4.8) values for Average Precision, Recall and Precision.



SampleMaskData
TrialMaskData
accuracy.m
AnchorMask.mat
bboxre.m
camera-off.jpg
groundTruth.mat
imageLabelingSession.mat
main.fig
main.m
mymaskdata.mat
myspeech.m
SSD.mat
TrainMask.mlx
Yolo.mat

TrainMask.mlx (Live Script)

Workspace

Name	Value
ans	8x2 table
ap	1
augmentedData	4x1 cell
augmentedTrainin...	1x1 TransformedData...
basenetwork	1x1 DAGNetwork
bboxes	[601,262,267,264]
bldsTest	1x1 boxLabelDatastore
bldsTrain	1x1 boxLabelDatastore
bldsValidation	1x1 boxLabelDatastore
data	1x3 cell

Evaluate the object detector using average precision metric.

```
[ap, recall, precision] = evaluateDetectionPrecision(detectionResults, preprocessedTestData,0.5);
```

Command Window

```
>> precision

precision =

     1
     1

>> recall

recall =

     0
     1

>> ap

ap =

     1
```

Figure: 4.8 – AP, Precision and Recall values obtained using YOLO method

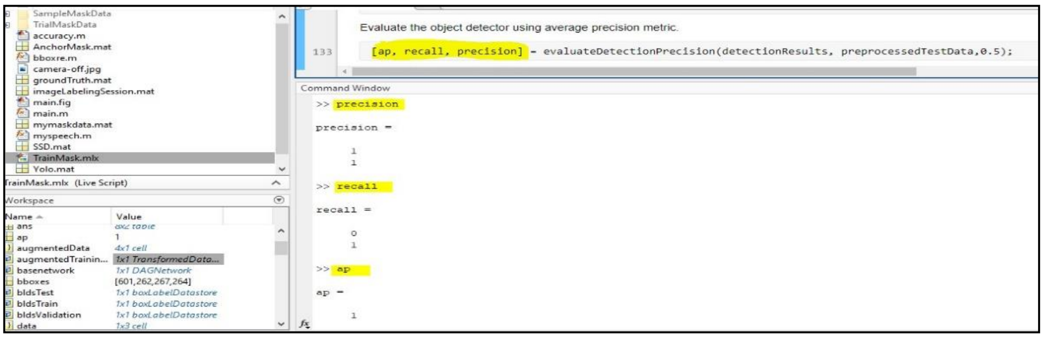
The Precision-Recall (PR) curve shows how precise the detector is at different recall levels. The ideal precision value is 1 for various recall levels. Using more data helps to improve the average precision but it requires more training time.

The PR curve is being plotted with precision and recall values we got (Figure 4.9). Our model using YOLO method gives Average Precision = 1.00, which is desirable.

4.14 Accuracy achieved with SSD method

Model multiple times reads the same image and the augmented training data is displayed. We can verify accuracy by inspecting training data loss and validation accuracy for each of the iteration (Table 5.4). Using SSD method, our model gives Accuracy of 97.6%.

For all the test images, we need to run the detector for evaluating the object detector while using Average Precision metric. We got the below (Figure 4.10) values for Average Precision, Recall and Precision.



SampleMaskData
TrialMaskData
accuracy.m
AnchorMask.mat
bboxre.m
camera-off.jpg
groundTruth.mat
imageLabelingSession.mat
main.fig
main.m
mymaskdata.mat
myspeech.m
SSD.mat
TrainMask.mlx
Yolo.mat

TrainMask.mlx (Live Script)

Workspace

Name	Value
ans	8x2 table
ap	1
augmentedData	4x1 cell
augmentedTrainin...	1x1 TransformedData...
basenetwork	1x1 DAGNetwork
bboxes	[601,262,267,264]
bldsTest	1x1 boxLabelDatastore
bldsTrain	1x1 boxLabelDatastore
bldsValidation	1x1 boxLabelDatastore
data	1x3 cell

Evaluate the object detector using average precision metric.

```
[ap, recall, precision] = evaluateDetectionPrecision(detectionResults, preprocessedTestData,0.5);
```

Command Window

```
>> precision

precision =

     1
     1

>> recall

recall =

     0
     1

>> ap

ap =

     1
```

Figure: 4.10 – AP, Precision and Recall values obtained using SSD method

The Precision-Recall (PR) curve shows how precise the detector is at different recall levels. The ideal precision value is 1 for various recall levels. Using more data helps to improve the average precision but it requires more training time.

The PR curve is being plotted with precision and recall values we got (Figure 4.11). Our model using SSD method gives Average Precision = 1.00, which is desirable.

Table: 4.5 – Accuracy comparison of referenced models with proposed model

S No.	Reference	Methodology	Result
1	(Nagrath & others) [19]	SSD	AC = 92.64 %
2	(Loey & others) [20]	YOLO v2 with ResNet50	AC = 81 %
3	(A Kumar & others) [21]	YOLO v4 with SPP	AC = 84.42 %
4	(Ejaz & others) [22]	PCA	AC = 70 %
5	Proposed	YOLO v2	AC = 71.6 %
6	Proposed	SSD	AC = 97.6%

Thus, our proposed model performs better than other referenced models [19, 20, 21, and 22] by achieving Accuracy score of 71.6% while using YOLO method and Accuracy score of 97.6% while using SSD method.

On analyzing all test results, following conclusion can be made:

Our suggested model provides 4.96 % more accuracy as compared with Nagrath & others [19] model using SSD model.

Our proposed model provides 1.6 % more accuracy using YOLO method as compared with Ejaz & others [22] model using PCA.

Our proposed model using SSD provides 26 % more accuracy as compared to our proposed model using YOLO method.

Quality feature values using YOLO and SSD method are within the best desired values. We also found that PSNR value is high, so noise value is low. Thus our system is working with low noise margin for both YOLO and SSD method.

The Precision-Recall (PR) curve shows how precise the detector is at different recall levels. The ideal precision value is 1 for various recall levels. Our model using YOLO and SSD method gives Average Precision = 1.00, which is desirable.

V. CONCLUSION

This project aims to distinguish mask wearers and non-mask wearers using machine learning. Our proposed algorithm transforms the captured input image into an expected output that identifies unmasked persons. Finally evaluating the numerical results obtained. This will help the health and hygiene authorities to implement the WHO guidelines in the most effective manner. Covid-19 spread is increasing every day in all parts of the world. It needs to be controlled so that we can return back to our normal lives. Professionals deal with vaccinations, but people can assist by following the WHO guidelines to control and eliminate the further spread of Covid-19 virus. This program has been tested using the methods previously discussed on a webcam and it is found that the results are as expected. Widespread application of this scheme in crowded places or any public gatherings; it will be an effortless task to identify those who violate the proper usage of the face masks. This thesis presents a face mask recognition system using deep learning techniques by developing YOLO and SSD model using machine learning techniques. Even though half of the face is covered by a mask, still this process provides fast and accurate results for the facial recognition in various security systems. The test result depicts a high level of accuracy for masked and unmasked face recognition. This model can achieve 71.6% performance accuracy with YOLO and 97.6% performance accuracy with SSD, which is a great achievement. The study also provides a helpful tool to combat the further spread of the Covid-19 disease by everyone to wear face mask for biometric authentication purpose. On analyzing all test results, following conclusion can be made:Our suggested model provides 4.96 % more accuracy as compared with Nagrath & others [19] model using SSD model.Our proposed model provides 1.6 % more accuracy using YOLO method as compared with Ejaz & others [22] model using PCA.Our proposed model using SSD provides 26 % more accuracy as compared to our proposed model using YOLO method.Quality feature values using YOLO and SSD method are within the best desired values. We also found that PSNR value is high, so noise value is low. Thus our system is working with low noise margin for both YOLO and SSD method.The Precision-Recall (PR) curve shows how precise the detector is at different recall levels. The ideal precision value is 1 for various recall levels. Our model using YOLO and SSD method gives Average Precision = 1.00, which is desirable.

5.1 Future Scope

With the help of this scheme implemented under the right conditions, people who are not wearing masks can be detected. This work has great scope for future researchers. The proposed model is not just limited only for the detection of face mask but also it can be integrated with a high resolution video surveillance system.

The proposed model can be further extended for the detection of various facial landmarks for biometric scans. A lot of software is used in many departmental stores today to count the number of customer entering the store. Therefore, the implementation of our proposed model can be extended to display the result on a digital screen.

In addition, our proposed model can be installed in conjunction with an alarm device, so that an audible signal will sound if someone enters a certain area without wearing a mask properly. This can be connected at the entrance gates of hospitals, shopping malls, schools or other public places.

Moreover, the system can be used for different networks and scenarios. It can be used for Industrial Internet-of-Things or smart cities environments for providing security and regulate when an alert should be triggered when someone tries to hide their face intentionally using mask.

Thus we can conclude that this model opens an interesting scope for future research enthusiasts. Restricting the violation of the face mask norms and detection of face identity further increases the scope and utility of this proposed model.

REFERENCES

[1] S. Mattisson, “An overview of 5G requirements and future wireless networks: Accommodating scaling technology,” IEEE Solid-State Circuits Magazine, vol. 10, no. 3, pp. 54–60, 2018.

[2] Q. Xu, D. Gao, T. Li, and H. Zhang, “Low latency security function chain embedding across multiple domains,” IEEE Access, vol. 6, pp. 14474–14484, January 2018.

[3] C. G. Gheorghe, D. A. Stoichescu, and R. Dragomir, “Latency requirement for 5G mobile communications,” in Proc. 10th International Conference on Electronics, Computers and Artificial Intelligence, 2018.

[4] M. Sybis, K. Wesolowski, K. Jayasinghe, V. Venkatasubramanian, and V. Vukadinovic, “Channel coding for ultra-reliable lowlatency communication in 5G systems,” in Proc. IEEE 84th Vehicular Technology Conference, 2016.

[5] J. H. Bae, A. Abotabl, H. P. Lin, K. B. Song, and J. Lee, “An overview of channel coding for 5G NR cellular communications,” APSIPA Trans. Signal Inf. Process., vol. 8, pp. 1–14, 2019.

[6] M. Liyanage, I. Ahmad, A. B. Abro, A. Gurtov, and M. Ylianttila, A Comprehensive Guide to 5G Security, USA: John Wiley & Sons Ltd, 2018.

[7] S. Bhattacharya, “Cryptology and information security-past, present, and future role in society,” Int. J. Cryptogr. Inf. Secur., vol. 9, no. 1, pp. 13–36, 2019.

[8] A. Muhammad Abdullah. (2017). Advanced encryption standard (AES) algorithm to encrypt and decrypt data. Cryptogr. Netw. Secur. pp. 1–13. [Online]. Available: https://www.researchgate.net/publication/317615794.

[9] I. Ahmad, T. Kumar, M. Liyanage, J. Okwuibe, M. Ylianttila, and A. Gurtov, “Overview of 5G security challenges and solutions,” IEEE Communications Standards Magazine, vol. 2, no. 1, pp. 36– 43, April 2018.

[10] J. Pan, P.-W. Tsai, and J. Watada, Advances in Intelligent Information Hiding and Multimedia Signal Processing, Smart Inno. Springer, 2017.

[11] R. author Ahlswede, A. Ahlswede, I. Althöfer, C. Deppe, and U. Tamm, Hiding Data - Selected Topics : Rudolf Ahlswede’s Lectures on Information Theory 3, Germany: Springer Nature, 2016.

[12] FIPS-197: Specification for the Advanced Encryption Standard (AES), Fed. Inf. Process. Stand. Publ. issued by NIST, 2001.

[13] J. Daemen and V. Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard. Berlin: Springer-Verlag Berlin Heidelberg GmbH, 2002.

[14] D. J. C. Mackay and R. M. Neal, “Near shannon limit performance of low density parity check codes-to be published in electronics letters,” Electron. Lett., vol. 33, no. 6, pp. 457–458, 1997.