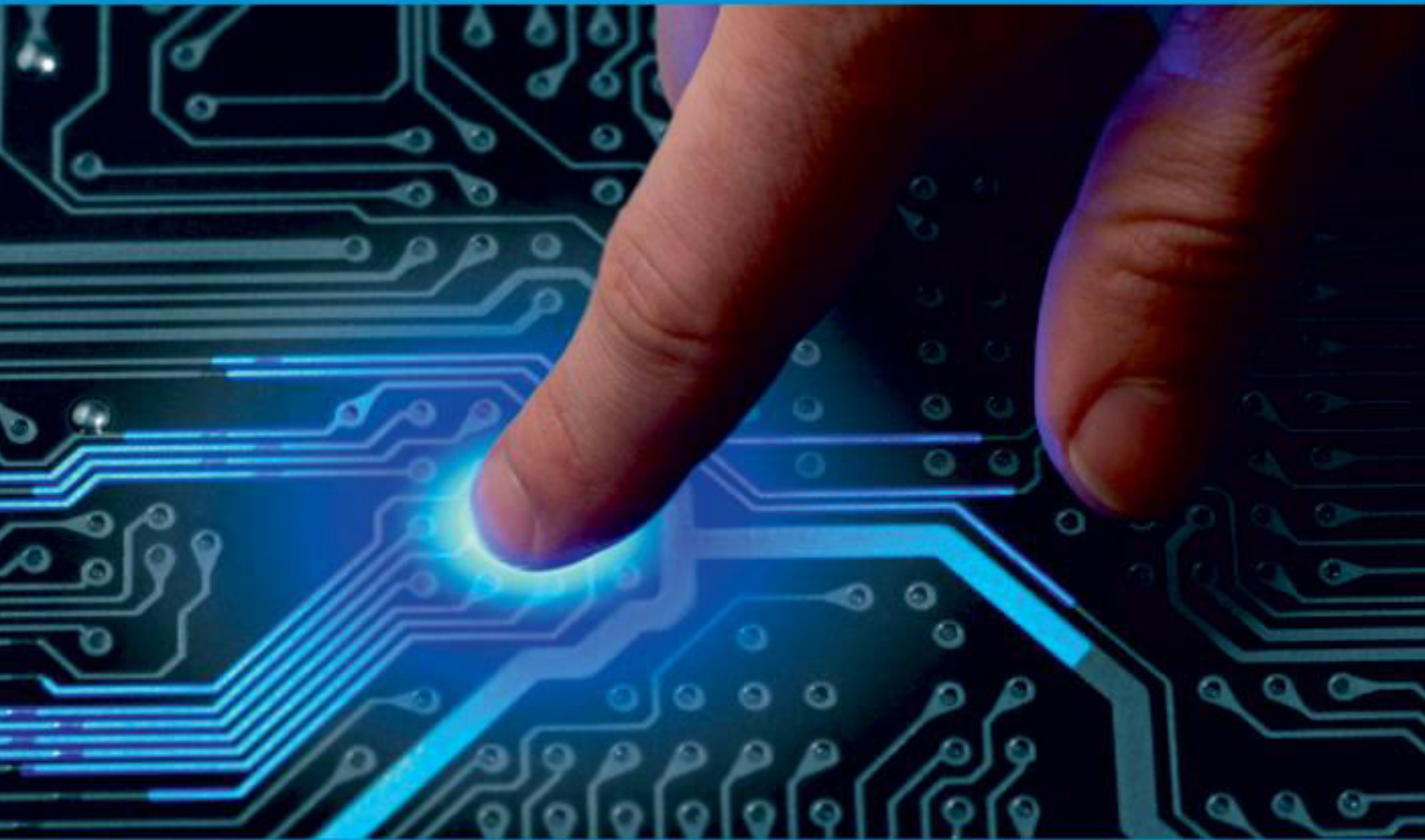
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 5, May 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005244|

Face mask Detection Using Convolution Neural Network (CNN) and MobileNetV2 Architecture

Kola Ratna Sri[*1], Mohammad Khutija Tasneem[*2], Malledi Sambhavi[*3],

Mr. G. Amar Tej[*4]

Student, Department of ECE, Vasireddy Venkatadri Institute of Technology, Nambur, A.P, India[*1,2,3]

Assistant Professor, Department of ECE, Vasireddy Venkatadri Institute of Technology, Nambur, A.P, India[*4]

ABSTRACT: Due to covid pandemic in the last 2 years, India has faced more than 3 crore cases and nearly 5 lakh deaths. Covid -19 coronavirus pandemic is wreaking havoc on human life. To overcome this people have to follow social distancing norms and wear masks in public places. Many precautionary measures have been taken to prevent the spread of this disease , including the usage of a MASK which is strongly recommended by the World Health Organization (WHO).Many people neglect wearing masks in public places. To monitor this issue, we propose a method to detect whether a person is wearing a mask or not. Here , we propose a face mask detection system using deep learning and Tensorflow. It includes a deep neural network ,more specifically a convolutional neural network, to differentiate between images of people with and without masks. And we propose to integrate a system in various embedded devices with limited computational capacity as it uses MobileNetV2 architecture.

KEYWORDS: Covid-19, deep neural network, convolution neural network, transfer learning algorithm,MobileNetV2 architecture.

I. INTRODUCTION

Covid 19 well known for Coronavirus which is an infectiousairborne disease. It is more likely to get affect people who are already suffering from chronic diseases like Cancer, Diabetes. We can prevent ourselves from Covid 19 by wearing mask which is suggested by scientists. It is observed by the scientists that coronavirus spread through fluid particles, which are usually emitted by people when speak.Till 2001 Face Detection is one of the biggest computer vision problem. Later ViolaJones Face detector made it possible to detect faces. We know that in the last few years deep learning has made a large breakthrough in computer vision areas, such as image classification, object detection, object segmentation. There are many algorithms but Convolution Neural Network (CNN) is preferred because it can automatically learn features from the datasets which are trained . Purpose of the project is to develop a Face Mask Detection system by using OpenCV, Deep Learning which includes detectionof face masks in static images and also in real-time video streams.Here two steps are to be followed, the first step is identify faces on an image/live video by using the trained image detector Open cv and the second step is that mask detection which detects and let us know that whether the face is having mask or not. A dataset is to be feeded for both 'withmask' and 'withoutmask'. The script for face mask detector is in python, it agrees with the input dataset, loads and pre-processes the images, and labels them using TensorFlow, kerasand sklearn. It again filters model with MobileNetV2 classifier using pre-trained ImageNet weights. Perform face mask detection correctlyin static images present in folders. Using webcam, the system developed itself applies face mask detection correctly to every frame in the real-time video stream. Out put which we get explains about probability of "Mask" or "No Mask" in the static images and realtimevideo streams. The system can be easily linked to various embedded devices with limited computational capacity as it uses MobileNetV2 architecture.

II. METHODOLOGY

Existing Methods :

[1] **F.M.Javed Mehedi Shamrat** has propoed a Face Mask Detection using Convolutional Neural Network (CNN) to reduce the spread of Covid- 19 .He Used three deep learning methods for face mask detection,

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |
|| Volume 10, Issue 5, May 2022 ||
| DOI: 10.15680/IJIRCCE.2022.1005244|

including Max pooling, Average pooling, and MobileNetV2 architecture, and showed the methods detection accuracy. A dataset consists of 1845 images from various sources and 120 co-author pictures taken with a webcam and a mobile phone camera is used to train a deep learning architecture. Epochs used here are 13. In this ,training accuracy and validation accuracy is 96.2%.

[2] **Rahman, Mohammad Marufur, Md MotalebHossenManik, Md Milon Islam, Saifuddin Mahmud, and Jong-Hoon Kim** has proposed a "An automated system to limit COVID-19 using facial mask detection in smart city network." In this , a framework is proposed that confine the development of Coronavirus by discovering individuals who are not wearing any facial mask in a smart city network. A deep learning design is prepared on a dataset that comprises of pictures of individuals with and without masks gathered from different sources. The trained architecture accomplished 97% precision on recognizing individuals with and without a facial mask for previously unseen test data. The created framework faces challenges in arranging faces covered by hands since it nearly resembles the individual wearing a mask. While any individual without a face mask is going on any vehicle, the framework can't find that individual accurately. For a thickly populated zone, recognizing the face of every individual is troublesome. For this sort of situation, recognizing individuals without face mask would be very hard for the proposed framework. To get the best result out of this framework, the city should have an enormous number of CCTV cameras to screen the entire city as well as dedicated manpower to uphold appropriate laws on the violators. Since the data about the violator is sent through SMS, the framework fails when there is an issue in the network.

III. PROPOSED METHOD

Here we proposed a Face mask Detection Using Deep Learning more specifically Convolution Neural Network(CNN) and MobileNetV2 Architecture .The dataset used in this project consists of real images of faces with and without protective face masks. This dataset consists of 4095 images belonging to two classes:

- with_mask: 2165 images
- without_mask: 1930 images

The dataset will be split into 80% training and 20% testing data with the help of sklearnlibrary .Epochs used are 20 in our project.

In this project , we used Open CV ,Deep Learning,OpenCV's face detector based on the Single Shot Multibox Detector (SSD) framework combined with MobileNetV2 architecturefor face mask detection, including Max pooling, Average pooling, and MobileNetV2 architecture to detect the face mask..

Open CV(Computer vision) :

It is a form of Artificial Intelligence that is used by computer to identify things using different algorithms. These algorithms will get trained to collect predefined features helping the computer to pick objects out of the crowd.

Deep Learning:

It is a subset of Artificial Intelligence which is inspired by the structure of the human brain. In terms of deep learning, this structure is called a Neural Network. In Deep Learning , the features required are picked by the Neural Network. The Output is predicted by the neurons in a Neural Network. Neurons have some connections which performs some specific sub tasks for detecting an object are called Neural Networks. It is a trained network. It is called as a dense Neural Network when every Neuron is connected to every other Neuron in other different layers .

OpenCV's "Deep Neural Networks" (DNN) module:

OpenCV comprises of highly efficient DNN module supported by many number of Deep Learning frameworks such as Caffe, TensorFlow, and Torch/PyTorch. This module has a more accurate Caffe-based face detector. In this project, we will be training our Deep Learning model using Caffe.

OpenCV's face detector based on the Single Shot Multibox Detector (SSD) frameworkcombined with MobileNetV2 architecture:

In order to obtain the bounding box *(x, y)*coordinates for an object (mask in this case) in an image we need to apply object detection. SSDs, originally developed by Google, are a balance between R-CNNs and YOLOmethods of object detection. The algorithm is more straightforward than Faster RCNNs.Network architectures such as VGG or Resnet are unsuitable for resource constraineddevices due to their sheer size and resulting number of computations. Instead, weuse MobileNets. They are designed for resource constrained devices such as yoursmartphone.Hence, if we

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005244|

combine both the MobileNet architecture and the Single Shot Detector(SSD) framework, we will have a fast, efficient deep learning-based method forobject detection.

KerasImageDataGenerator:

Generate batches of tensor image data with real-time dataaugmentation. The data will be looped over (in batches).KerasImageDataGenerator class works by:

- Accepting a batch of images used for training.
- Taking this batch and applying a series of random transformations to each image inthe batch (including random rotation, resizing, shearing, etc.)
- Replacing the original batch with the new, randomly transformed batch**.** Training the CNN on this randomly transformed batch (i.e., the originaldata itself is notused for training)

IV. MODELING AND ANALYSIS

Pre-processing:

OpenCV's blobFromImage and blobFromImages to facilitate image pre-processing:

In this Project, cv2.dnn.blobFromImage and cv2.dnn.blobFromImages functions of OpenCV's dnn module facilitates image pre-processing for deep learning classification. These two functions perform:

- **Mean subtraction** - Mean subtraction is used to help tackle illumination changes in the input images in our dataset. For example, the mean values for the ImageNet training set are R=103.93, G=116.77, and B=123.68
- **Scaling** - The scaling factor aids in normalization. A 'blob' is just a collection of image(s) with the same spatial dimensions (width and height), same depth (number of channels), that have to be pre-processed in the same manner. [blobFromImage] creates 4-dimensional blob from image. Optionally resizes and crops image from center, subtracts mean values, scales values by scale-factor, swaps Blue and Red channels.

blob = cv2.dnn.blobFromImage(image, scalefactor=1.0, size, mean, swapRB=True)

Each parameter is described below:

- **Image**: This is the input image we want to pre-process before passing it through our deep neural network for classification.
- **Scalefactor:** After we perform mean subtraction we can optionally scale our images by some factor. This value defaults to `1.0` (i.e., no scaling) but we can supply another value as well. It's also important to note that scalefactor should be as we're actually multiplying the input channels (after mean subtraction) by scalefactor.
- **Size:** Here we supply the spatial size that the Convolutional Neural Network expects. For most current state-of-the-art neural networks this is either 224×224, 227×227, or 299×299. We will be using 224×224.
- **Mean:** These are our mean subtraction values. They can be a 3-tuple of the RGB means or they can be a single value in which case the supplied value is subtracted from every channel of the image.
- **SwapRB:** OpenCV assumes images are in BGR channel order; however, the `mean` value assumes we are using RGB order. To resolve this discrepancy, we can swap the R and B channels in image by setting this value to `True`. By default, OpenCV performs this channel swapping for us. The cv2.dnn.blobFromImage function returns a blob which is our input image after mean subtraction, normalizing, and channel swapping.

Convolutional Neural Networks:

CNN is a feed forward neural network that is generally used to analyse visual the images by processing the data . A CNN is also known as a "ConvNet".

The networks that can be used to enable the machines to visualize the things and also performs certain tasks like image classification, detection and recognition. The process involves taking an input image then process the image and finally classify the image under some categories like In face - nose ,lips,ears ,chin and eyes.For the purpose of image classification and processing of image , these neural networks are used.Mainly,thera are 3 layers in cnn they are:

$$\mathbf{y_i = b_i + \sum x_i \in x W_{ij} * x_i}$$

Where $y_i \in Y$, $i = 1,2,\ldots D$. D is the depth of the convolutional layer. Each filter W_{ij} is a 3D matrix of size [F × F × CX]

Input layer : Input layer accepts the pixels of the image as input in the form of arrays .

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005244|

Hidden layer :Hidden layers carry out feature extraction by performing certain calculation and manipulation.CNN consists of multiple hidden layers like Convolution layer,ReLU layer ,Pooling layer,etc that perform feature extraction from the image.This layer uses a matrix filter which performs the convolution operation to detect the patterns in the image.

Output layer **:** The output layer consists of a fully connected layer which identifies the object in the image .

.

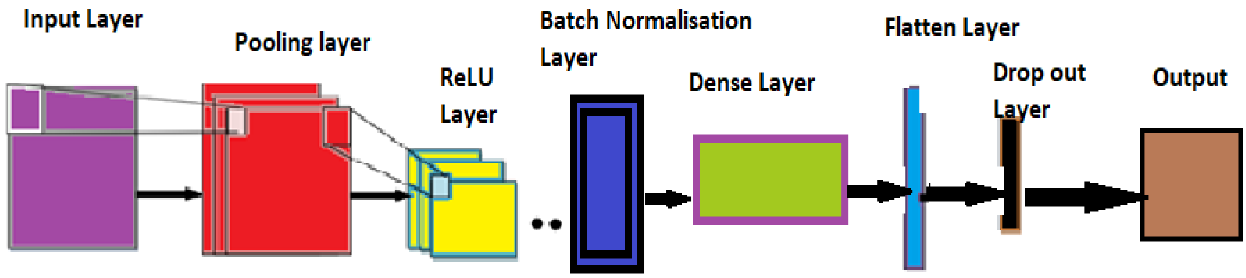


Figure 1:CNN layers

In this project ,we applied deep convolution neural network with 6 layers. They are:

- Pooling Layer
- ReLU Layer
- Batch Normalization
- Dense Layer
- Flatten Layer
- Dropout Layer

Pooling Layer :

Pooling is the operation of down sampling the input that is provided to a layer.So if a feature map of dimension $h \times w \times c$ is presented then the output obtained by the pooling will be.

$$p = \frac{(h - f + 1)}{s \times (w - f + 1) \times c}$$

Where c is the channel presented in the feature map

f is the size of the filter

s is stride length

h and w are height and width of the feature map respectively

There are two different kinds of pooling –

1.Max Pooling

2.Avearge Pooling

Max pooling :

Max pooling identifies the most important feature in a particular vision and neglects the other information which isn't useful.

Average pooling:

The main task of the average pooling is to reduce the size of the input . But in case of Max pooling we are considering the maximum features instead of taking the maximum feature that is present in the region we are taking the average of the all the features that are present in the region

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005244|

ReLUlayer :

Once the feature maps are extracted, immediately the next step is to move them to a ReLUReLU layers helps to prevent the growth to operate the neural network.

$$y = \max(0, x)$$

Where y = output variable

x = input variable

The main advantage of this ReLU layer function over the other activation functions is that it does not activate all the neurons present at the same time .

Batch Normalization:

Batch Normalization is the network layer which is inserted in between a hidden layer and the next hidden layer. The main aim of this layer is to take the outputs from the first hidden layer and then normalize them before passing them on as the input of the next hidden layer .

$$x * = \frac{(x - E[x])}{\sqrt{(var(x))}}$$

where x *= output variable

x = input variable

$E[x]$= mean within a batch

Flatten:

It is the last step performed in a convolution neural network .Flatten Layer involves taking the pooled feature map which is generated in the pooling step and transforming it into a 1 dimensional vector form.The output of flatten layers once done can be given as inputs to the dense layer.

Dense Layer:

Dense layer is used to divide the image based on the output from the convolutional layers.It is a regular layer of neurons in a neural network .Each and every neuron receives the input from all the neurons in the past layer, and thus densely connected to the layers. The dense layer's neuron in a model receives output from every neuron of its preceding layer, where neurons of the dense layer performsmatrix-vector multiplication. It is a procedure where the row vector of the output from the preceding layers is equal to the column vector of the dense layer.This layerreduces the dimension of the vectors. So, basically a dense layer is used for changing the dimension of the vectors by using every neuron.

Dropout Layer:

It is a technique that drops the neurons from the network.Different neurons are removed from the network on a temporary basis.Dropout Layer is placed on the fully connected layers

Mobile net v2:

In convolutional neural network we have MObileNet-V2 which is of 53 layers deep. It performswell on the mobile devices. Compared to the mobile net v1, the mobile net v2 models are to be faster. It is a very effective feature extractor for object detection and segmentation. In the mobile net v2 with larger image sizes offering better performance, so MobileNets support any input size greater than 32 x 32.

The advantages of using mobile net v2 are decreased network size, decreased number of parameter,Quick inperformance and are helpful for mobile applications.Mobile net v2 is a model commonly deployed on low compute devices such as mobile with high accuracy performance.

Transfer Learning Algorithm:

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is the idea of overcoming the isolated learning paradigm and utilizing the knowledge acquired for one task to solve related ones. Traditional learning is isolated and occurs purely based on specific tasks, datasets and training separate isolated models on them. No knowledge is retained which can be transferred from one model to another. In transfer learning, you can leverage knowledge (features, weights etc.) from previously trained models for training newer models and even tackle problems like having less data for the newer task! Learning is not an easy process, not for humans and not for machines either. It is a heavy-duty, resource-consuming and time-consuming process and hence it was important to devise a method that would prevent a model from forgetting

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005244|

the learning curve that it attained from a specific dataset and also lets it learn more from new and different datasets. Transfer learning is simply the process of using a pre-trained model that has been trained on a dataset for training and predicting on a new given dataset.
"A pre-trained model is a saved network that was previously trained on a large dataset, typically on a large-scale image-classification task such as the ImageNet."

Applications of Transfer Learning:

1. Transfer learning for NLP
2. Transfer learning for Audio/Speech
3. Transfer learning for Computer Vision

Layer(Type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
Conv1_pad (ZeroPadding2D)	(None,225,225,3)	0
Conv1 (Conv2D)		864
bn_Conv1(BatchNormalization)	(None,112,112,32)	128
Conv1_relu (ReLU)		0
expanded_conv_depthwise_BN (Bat)	(None,112,112,32)	128
expanded_conv_depthwise_relu (R)		0
expanded_conv_project_BN (Batch)	(None,112,112,32)	64
block_1_expand (Conv2D)		1536
block_16_depthwise_BN (BatchNor)	(None,112,112,32)	3840
block_16_depthwise_relu (ReLU)		0
Conv_1 (Conv2D)	(None,112,112,32)	409600
out_relu (ReLU)		0
average_pooling2d (AveragePooling)	(None,112,112,12)	0
flatten (Flatten)		0
dense (Dense)	(None,112,112,96)	163968
dropout(Dropout)	(None,7,7,960)	0
dense_1(Dense)	(None,7,7,960)	258
	(None, 7, 7, 1280)	
	(None, 7, 7, 1280)	
	(None, 1, 1, 1280)	
	(None,1280)	
	(None, 128)	
	(None, 128)	
	(None, 2)	

Figure 2: Overview of CNN Layers

V. RESULTS AND DISCUSSION

Work Flow:

Work flow is the main building blocks of every object-oriented methods. The class diagram can be used to show the classes, relationships, interface, association, and collaboration. This is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.It shows:

- Static structure of classifiers in a system
- Diagram provides a basic notation for other structure diagrams prescribed .
- Helpful for developers and other team members too

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005244|

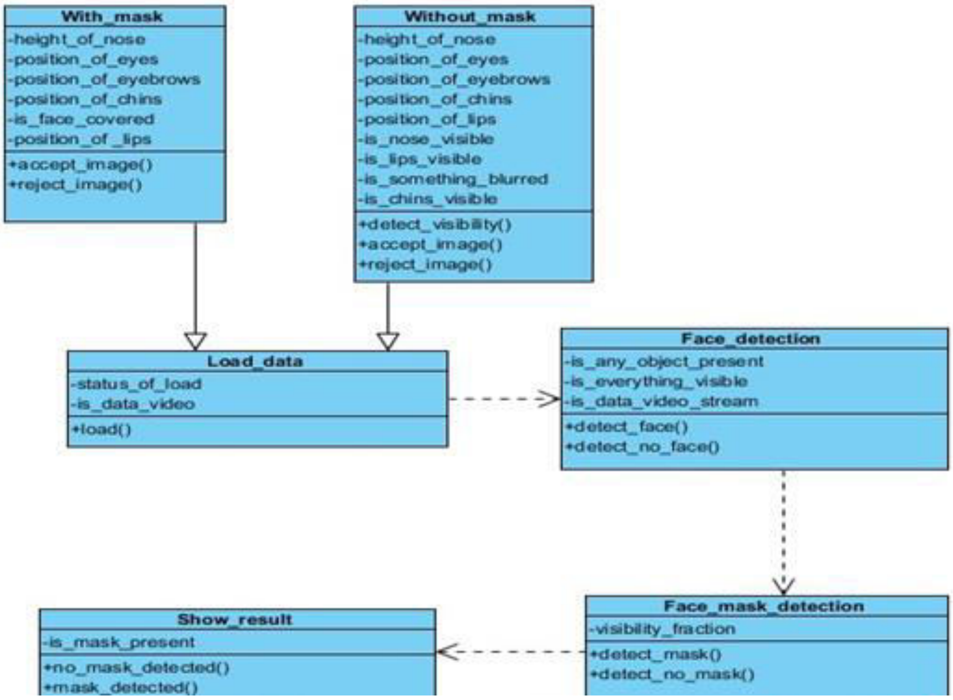


Figure 3: Model Workflow

By using the above materials and methods we got an accuracy of about 97% and also the confusion matrix of the testing data is shown below. Accuracy is calculated by the total number of True predictions to the actual number of the images. Similarly, we also computed Precision, Recall, F1 score, Specificity by using False Positive, False Negative, True Positive, and True Negatives by the below formulas.

- True Positive (TP) = 429
- True Negative (TN) =381
- False Positive (FP) = 5
- False Negative (FN) =4

- Accuracy $=\frac{(TP+TN)}{(TP+TN+FP+FN)} = 0.99$
- Precision $= \frac{TP}{(TP+FP)} = 0.99$
- Recall $= \frac{TP}{(TP+FN)} = 0.99$
- F1-score $= \frac{(2\times Recall\times Precision)}{(Recall\times Precision)} = 0.99$

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005244|

Confusion Matrix

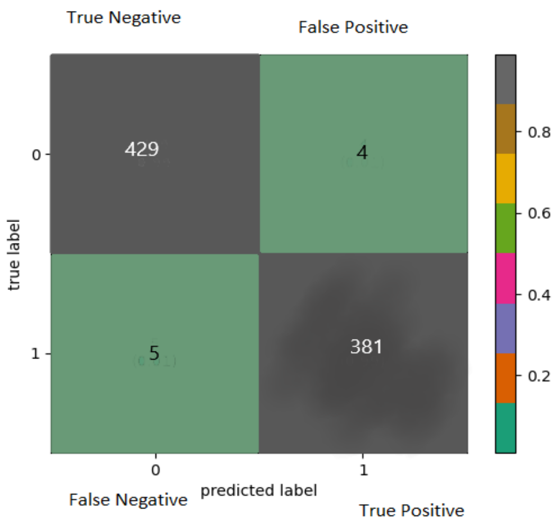


Figure 4: Confusion matrix

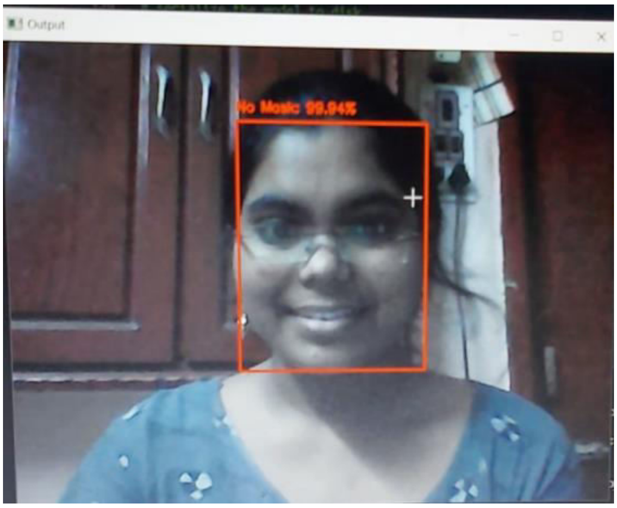


Figure 5: Detecting a face with mask

Figure 6: Detecting a face with no mask

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005244|

Figure 7: Detecion of Face with mask

Figure 8 : Detection of face with no mask

VI. CONCLUSION

An accurate and efficient face mask detection system has been developed which achieves comparable metrics with the existing state-of-the-art system. This project uses recent techniques in the field of computer vision and deep learning. Custom dataset was made from scratch using Bing Search API, Kaggle datasets and RMFD dataset, and the evaluation of the model on test dataset was found consistent. The system correctly detected the presence of face masks on human faces that it detected in static images as well as real-time video streams. To create our face mask detector, we trained a two-class model with images of people wearing masks and not wearing masks. We then fine-tuned our model using MobileNetV2 on our mask/no mask dataset and obtained an image classifier that was 93% accurate. We then took this face mask classifier and applied it to both images and real-time video streams by:

1. Detecting faces in the images/video
2. Extracting each individual face ROI
3. Applying our face mask classifier

Our face mask detector is accurate, and since we used the MobileNetV2 architecture, it’s also computationally efficient and thus making it easier to deploy the model to embedded systems. This system can therefore be used in real-time applications which require face-mask detection for safety purposes due to the outbreak of Covid-19. This project can be integrated with embedded systems for application in airports, railway stations, offices, schools, and public places to ensure that public safety guidelines are followed.

REFERENCES

[1] Loey M, Manogaran G, Taha MHN, Khalifa NEM. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. Measurement:Journal of the International Measurement Confederation. 2021 Jan;167:108288. DOI: 10.1016/j.measurement.2020.108288.

[2] M. Jiang, X. Fan, and H. Yan, “RetinaMask: A Face Mask detector,” 2020. [Online]. Available: http://arxiv.org/abs/2005.03950.

[3] 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), Vancouver, BC, Canada, 2020, pp. 1-5, doi: 10.1109/IEMTRONICS51293.2020.9216386

[4] An automated System to limit covid 19 using facial mask detection in smart city network(2020, IEEE) https://ieeexplore.ieee.org/document/9216386.

[5] Jignesh Chowdary, G., Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal. "Face mask detection using inceptionv3." In International Conference on Big Data Analytics, pp. 81-90. Springer, Cham, 2020.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005244|

[6] Meenpal, Toshanlal, Ashutosh Balakrishnan, and Amit Verma. "Facial mask detection using semantic segmentation." In 2019 4th International Conference on Computing, Communications and Security (ICCCS), pp. 1-5. IEEE, 2019.

[7] Mohamed, Gunasekaran Manogaran, Mohamed Hamed N. Taha, and Nour Eldeen M. Khalifa. "Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection." Sustainable cities and society 65 (2021): 102600.

[8] Aniruddha Srinivas Joshi, Shreyas Srinivas Joshi, Goutham Kanahasabai, Rudraksh Kapil, Savyasachi Gupta," Deep Learning Framework to Detect Face Masks from Video Footage",2020

[9] https://www.pyimagesearch.com/2019/06/03/fine-tuning-with-keras-and-deep-learning/

[10] https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset

[11]https://github.com/AIZOOTech/FaceMaskDetection

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING