2021 3rd International Conference on Signal Processing and Communication (ICPSC) | 978-1-6654-2864-4/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICSPC51351.2021.9451688

CNN-BASED MASK DETECTION SYSTEM USING OPENCV AND MOBILENETV2

Harriat Christa.G
Electronics and Communication Engineering
Karunya Institute of Technology and Sciences
Coimbatore, India
harriatchrista@gmail.com

Jesica.J
Electronics and Communication Engineering
Karunya Institute of Technology and Sciences
Coimbatore, India
jesjes982@gmail.com

Anisha.K
Electronics and Communication Engineering
Karunya Institute of Technology and Sciences
Coimbatore, India
anishakingslyd@gmail.com

K.Martin Sagayam
Electronics and Communication Engineering
Karunya Institute of Technology and Sciences
Coimbatore, India
martinsagayam.k@gmail.com

***Abstract*—this paper establishes a 'Safety system for mask detection during this COVID-19 pandemic'. Face mask detection has seen an overwhelming growth in the realm of Computer vision and deep learning, since the unprecedented COVID-19 global pandemic that has mandated wearing masks in public places. To tackle the situation, machine learning engineers have come up with several algorithms and techniques to identify unmasked individuals using various mask detection models. The proposed approach in this paper adopts frameworks of deep learning, TensorFlow, Keras, and OpenCV libraries to detect face masks in real time. The trained MobileNet model, presented in this paper, yielded an accuracy score of 0.99 and an F1 score of 0.99 in the training data. This user-friendly model can be incorporated with several existing technologies such as face detection, biometric authentication and facial expression detection for further advancements in the future.**

***Keywords*—Convolution Neural Network, MobileNetV2, OpenCV, Keras.**

I. INTRODUCTION

The rise of the coronavirus globally has necessitated wearing face masks as a regulation to safeguard from contracting the virus. As COVID-19 has been proven to be transmitted predominantly through airdrops, wearing a mask has become a prerequisite to combat the spread of the virus. Previously, people wore masks to protect themselves from air pollution but as of now, wearing a mask is imperative to obstruct the virus particles from entering nostrils. The World Health Organisation (WHO) has declared this COVID-19 as a global pandemic and has authenticated that wearing face masks can help contain the COVID-19 transmission rate. Yet, quite a few people are frivolous and tend to elude themselves from wearing a mask. A constant surveillance to spot the unmasked and improperly masked people is indispensable.

Computer vision is one of the emerging frameworks in the field of object detection and is widely being used in various aspects of research in artificial intelligence. There have been both supervised and unsupervised approaches of machine learning in the past for object detection in an image. An enhanced supervised machine learning technique of object detection has been deployed in this paper. In order to detect masks, initially the model is trained using the Multi-Task Cascade Convolutional Neural Network (MTCNN) algorithm with the database of faces provided. Face is first detected using OpenCV (Open-Source Computer Vision) and those image frames are stored and passed to mask detection classifier for classification. For facial detection, Viola Jones algorithm is used which is also known as Haar Cascade algorithm. We have used a mask classifier model to differentiate faces with masks and without masks. The RGB(Red Green Blue) images are used as data for the classifier. The images are fed to the model after pre-processing and are trained. This model has provided better accuracy than the conventional CNNs. This model can also be used to identify crime suspects who cover their faces partly while performing illegal deeds.

II. LITERATURE SURVEY

As in [1], the authors developed a new facemask identification method that is able to classify three conditions of mask-wearing categories viz masked, improperly face masked, and unmasked. The propounded method has achieved an accuracy of 98.70% in the face detection phase. In this paper, two components are used. Firstly, Deep transferring learning component is used for feature extraction and classical machine learning as the second component. Model performance is improved using three traditional classifiers (Support Vector Machine, decision Tree and Ensemble). The model yields accuracies of 94.54% for decision tree classifier and 99.49% for SVM classifier. The Machine Learning algorithm that creates a collection of classifiers is used in Ensemble.

In this paper [2], SSDMNV2 model is designed for face mask detection using OpenCV Deep Neural Network (DNN), TensorFlow and Keras libraries. It employs MobileNetV2 architecture (Nguyen, 2020) for classification. This model plays competently in differentiating images having masked frontal faces from unmasked frontal faces and detecting whether the mask is worn properly or not. With only a few datasets available that are either artificially created or full of noise or wrongly labelled, data cleaning, error identification and correction is crucial in the pre-processing stage. The

dataset is fed as input and all the files are uniformly resized according to the SSDMV2 model requirement. NumPy array is used for faster mathematical operations. Using multibox, multiple objects in the scene are captured in one shot which is analogous to the YOLO technique. This object detection algorithm is faster in speed and has high accuracy.

In this paper [3], a deep learning algorithm for medical face mask detection is proposed. It exploits YOLO-v2 with the ResNet-50 model to achieve high average perception. It employs two stages. Firstly, feature extraction process based on ResNet-50 and secondly, detection of surgical face mask using YOLO-v2. Average precision and log-average miss rates are used as performance metrics. The distance of similarity between bounding boxes of the target is determined using IoU(Intersection over Union). Using Adam optimizer, the YOLO-v2 model obtains Average Precision (AP) of 81%.

To address the problems involved in face mask recognition for face authentication and face matching on a masked face, a deep learning-based feature is proposed in this paper [4]. Firstly, the masked face region is discarded and the best features are extracted from the obtained regions by applying a pre-trained deep convolutional neural network. The first step is to detect whether the person is masked or not and the second step is to identify the face with the mask on the ground of the eyes and forehead region using a deep learning-based model. In this model, a third Feature Map (FM) is used along with the classical CNN and obtains a recognition rate of 91.3%.

With reference to [5], a statistical dimensionality reduction technique called Principal Component Analysis (PCA) has been implemented on face recognition for detecting masked and unmasked faces. The proposed system is trained to detect frontal faces from the image. For the detection, Viola Jones object detection algorithm, that combines Haar-features and Adaboost algorithm has been deployed. Feature extraction is performed using the PCA algorithm on the training images. The resulting PCA features face is also called Eigen face. These Eigen faces have been used for further process of recognition. In this recognition process, the target image is compared with the corresponding template image. The recognition rate obtained is 72% on masked faces and 95% on non-masked faces. The obtained recognition rate is poorer for masked faces and performs satisfactorily on non-masked faces.

III. PROPOSED WORK

1. Hardware Component

A webcam has been employed as the hardware component in this paper.

1.1 Web Camera

A webcam is an input display device that helps to feed a live image or video in real time to the personal computer (PC). Webcams are usually small cameras that are either attached to a PC built-in to the hardware or externally connected via USB cables. Webcam software enables users to capture an image or record a video or stream the video online. In this paper, a web camera with a specification of full high definition (FHD, 1080p) vision and a rectangular aspect ratio of 16:9 has been used to test this proposed model in real time.

2. Software Component

The code for this paper has been executed in Google Collaboratory.

2.1. Google Collaboratory

Google Collab is an editable online notebook version of Integrated Development Environment (IDE) to work with machine learning libraries that executes python codes within the browser and runs entirely within the cloud. Output is obtained below the same executable code cell rather than a separate window as in the software version of IDEs. The Collab notebooks are highly integrated with google drive. Numpy package has been used for generating random data and to visualize it, matplotlib library has been used.

Several other libraries have come preinstalled in the toolkit which makes it ideal for machine learning projects and minimizes man work. Image dataset can either be uploaded or be imported via google drive to the environment for instructing and estimating the model. For the ML community this notebook has included many application frameworks with it like tensorflow, neural networks model, GPUs (Graphics Processing Units) experiments.

3. Methodology

In this paper we describe a new convolutional neural network architecture called MobileNet Convolutional Neural Network, a very effective feature extractor for object detection that works well with mobile models. The neural network is first trained using the available dataset to create a model for classifying masked and unmasked faces in the image. The trained cascade classifier will find out if the person is wearing a mask or not. The accuracy of the mask detection is processed in this paper. OpenCV, a real-time computer vision library has been used for pre-processing purposes. The workflow of the methodology is represented diagrammatically in fig.1.

3.1 Dataset

For rendering the best accuracy from the model, the input datasets play a vital role. Since COVID-19 is a recent pandemic, available datasets were limited. Due to this shortage of data, a mixture of many open-source datasets has been used for training. The first set of datasets was obtained from Kaggle's Medical Mask Dataset by Mikolaj Witkowski which contains images of many people wearing masks and XML files which contain the descriptions. The second set of dataset contains artificially generated mask dataset developed by Prajna Bhandary available at PyImageSearch which

contains standard face images with applied facial landmarks. This is a duplicate way to develop a dataset which includes a mask on a non-masked image of a person. The use of non-face mask samples collected from different sources include the model becoming heavily biased and under fit. So the dataset which contains images of people with and without masks that compensated for the error correction has been used. The dataset has 1,376 images in total divided into two classes viz 690 pictures with mask and 686 pictures without mask. Facial features allowed to locate facial landmark points of a person like eye size, thickness of eyebrows, nose shape, mouth, and jawline. The dataset has been made available for open access in the GitHub repository [8].

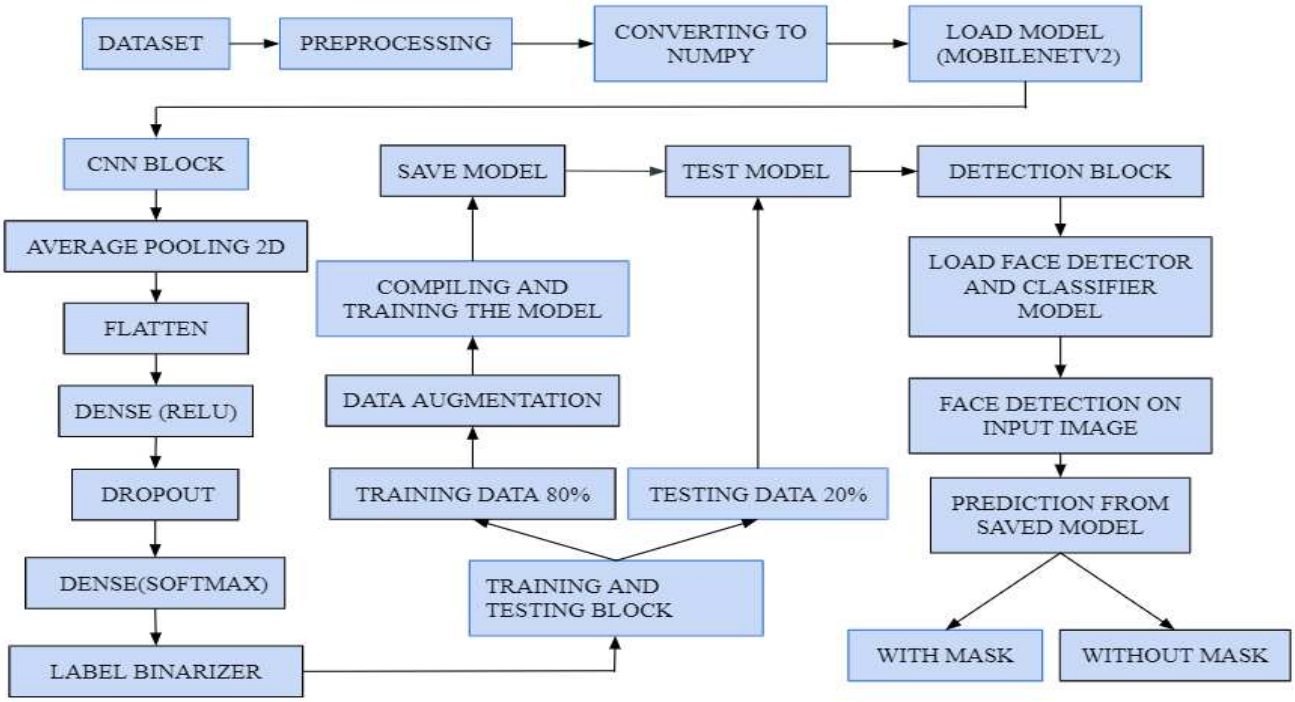

Fig.1. Workflow of the proposed system

3.2. Pre-Processing

The dataset for training contained a lot of noise and duplicates. The accuracy of the model depends on the dataset chosen for training. The dataset hence has to be pre-processed before being fed as input. The images are resized and the pixel representation of the images are converted into list format in accordance with the MobileNetV2 model. This list is then transformed to a NumPy array for quick mathematical operations.

3.3. Data augmentation

To train the MobileNet V2 model, being a deep learning framework, whose performance improves with the amount of dataset, a huge quantity of data is needed to prevent under fitting and for faster convergence of the neural network model. Due to the lack of sufficient amount of data required to train the model, data augmentation technique has been used to compensate for the limited dataset. In this process, methods like rotate, zoom, shift, shear, and flip the picture were done on each image iteratively to create many versions of the same image. For image augmentation, an image data is generated and function is created, and then tests and trains batches of data.

3.4. OPENCV-CNN

OpenCV is an open-source library aimed at performing computer vision tasks. It provides real-time optimized libraries, tools and machine learning framework support. . It performs many techniques such as facial detection and recognition, monitoring human locomotion through videos, tracking camera and object movements, designing three-dimensional models, eye movement tracking, grouping similar faces from the database. It is predominantly used in the CNN architecture and other network-based computer vision architectures. The neural network is trained by OpenCV using the Tensorflow framework.

A deep learning algorithm called Convolutional neural network is commonly used for pattern recognition. It has a multiple layer perceptron model which includes an input layer, many hidden layers and an output layer. It is termed as a fully connected network because every neuron in the first layer is tied up to all the other neurons of the upcoming layer. All mathematical computations and convolutions are performed in the middle layer also known as the hidden layer. The summation of the weighted inputs from each input layer is passed to an activation function, commonly ReLU, followed by a sequence of convolutional layers such as pooling layer, fully connected layer and dense layers.

3.4.1. Convolutional Layer

It is the fundamental block of CNN. Convolution is basically the fusion of two functions to receive another function. Here, the input data, which is a four-dimensional tensor (number of images, height, width and number of channels) is convolved with a convolutional filter called

kernel to obtain the convolved feature. It has been utilized to remove features using back propagation with the removed feature can be used for pattern recognition. The result of the convolutional layer is turned to be a feature map. The attributes which are followed in CNN are convolution filter with width and height, sum of input and output channels, convolution filter depth and convolution operations. The input matrix A with the convolution filter B, gives the output C which can be mathematically written as (1):

$$C(T) = (A * B)(x) = \int_{-\infty}^{\infty} A(T) \times B(T - x)\, dT \qquad (1)$$

3.4.2. Pooling Layer

The obtained feature map is influenced by the area of features in the input. This sensitivity can be overtaken in which the feature map requires down sampling of features. This makes the resulting feature map more speed to the variations in bearings of the aspect in the image. Pooling operations can be applied to make calculations faster, reducing the dimensions of the input matrix except any loss in aspect by summing up the existence of aspects in the feature map. This average presence of an aspect is summed up using two types of pooling methods. One is Average pooling and the other is maximum pooling also called Max pooling. In this paper, average pooling has been implemented. This pooling function sums up the average of all the values in the current kernel region and turns up a single value as the result. Average Pooling Layer is put in 2×2 patches of the feature map with a pace of (2, 2). This process includes summing the average for each patch of the feature map. Which is meant that every 2×2 square of the feature map is sampled down to an average value in the square.

3.4.3. Flatten Layer

Flatten layer combines all available native aspects of the preceding convolutional layers without affecting the batch size. Every feature map channel in the outer of a CNN layer is a result of multiple 2-D kernels which are developed from each channel of the input layer and stacked to form a "flattened" 2-D array. This layer converts a two-dimensional feature of a matrix into a one-dimensional array of vectors which is given into a fully connected neural network classifier. The tf.keras.layers. Flatten function reforms the tensor to a shape which will be equal to the sum of elements present in the tensor.

3.4.4.ReLU

Rectified Linear Unit (ReLU) is an activation function which is commonly used in deep learning models. It is a supervised learning method. The output of a ReLU function follows the input if the input is positive otherwise the output is zero. It is mathematically defined as: f(x) = max(0, x).

3.4.5. Dropout Layer

The layers which are connected fully makes the network of neuron models prone to over fitting while training. To prevent this, the dropout layer randomly drops a few neurons with respect to the dropout ratio while training. It sets the input of the neuron to 0 with a frequency during the training process.

3.4.6. Dense layer (Soft max)

A softmax layer can run a multi-class function. This function can turn the magnitude and direction of K real values into a magnitude and direction of K real values that sum to 1. Softmax transforms all the input data which can be positive, negative, zero, or greater than one, into values between 0 and 1, so that they can be said as probabilities. Softmax turns input into small probability, if one of the data is small or negative, and if any data is large, then it turns it into a large probability, but the value will always remain in the range between 0 and 1.

3.4.7. Fully-Connected Layer

The fully connected layers are added in the model and they have complete connections to activation layers. In this layer all the inputs are connected to the activation unit of the upcoming layer. The given images are classified as multi-class and the activation function which is used in layers and they give out the result of estimated output classes in terms of probability.

3.5. MobileNetV2

A deep neural network which has been used for the classification problem is MobileNetV2. To thwart the impairment of learning features the base layers are frozen. These layers help to identify the features to classify faces those who wear a mask from faces those who don't wear a mask by training the collected dataset. These pre-trained models don't use unwanted computational costs.

IV. EXPERIMENTAL OUTPUT

The results of this experiment are carried out using a laptop with Intel(R) Pentium(R) CPU N3540 @ 2.16GHz 2.16 GHz and installed RAM is 4.00 GB. Google Collaboratory for the development and implementation process. The datasets are taken from which are separated by labelling with mask and without mask. These folders are loaded along with their pixel values in order to train the model. The images are pre-processed and converted into NumPy arrays for faster calculation. For classification models Keras has been used.

The datasets are split like training and testing batches in the proportion of 4:1. In which training datasets are 80% and the remaining 20% is for testing purposes. After 20 epochs, the average accuracy of the model turns out to be 99%. Calculation of Accuracy and Precision was measured for positive predicted values. The classifier's ability was determined by Recall. The measure of test accuracy plot was given by F1 score. The training accuracy plot is shown in fig 3 and training loss is shown in fig 2.

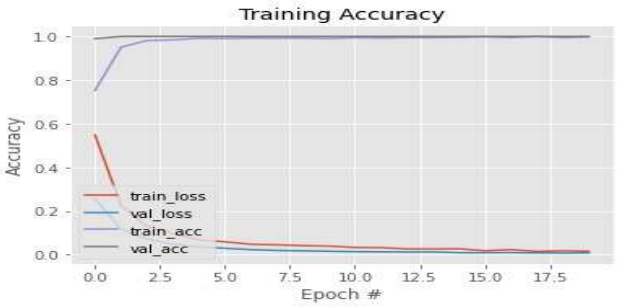

Fig. 2. Training Loss and Accuracy curve

Output with & without mask

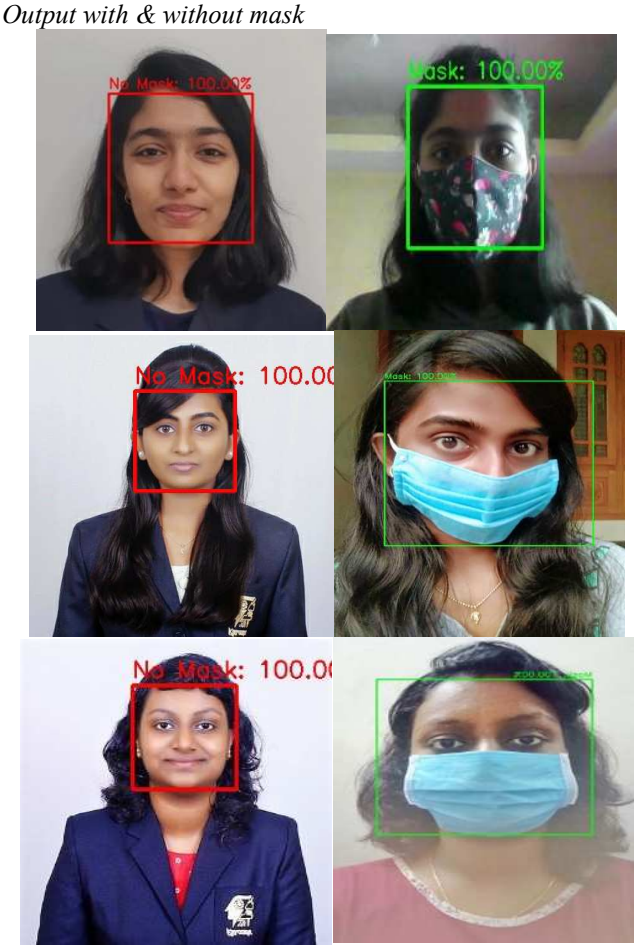

TABLE I

Dataset	*Classification Report*		
	Precision	*Recall*	*F1-score*
Without Mask	1.00	0.99	0.99
With Mask	0.99	1.00	0.99

Table.1.Performance metrics obtained

The output visualization and the prediction of the mask worn or not is finalized by the model created. When the mask is worn correctly a rectangular green box predicts with an accuracy score. On the other hand, when the mask is not worn then a red rectangular bounding box predicts with an accuracy score. The model predicts it from the trained dataset and labels for the detection.

TABLE II

S.No	*Comparison of Accuracy and F1 score for different architectures.*			
	Architecture Used	*Year*	*Accuracy*	*F1-Score*
1	LeNet – 5	1998	84.6	0.85
2	AlexNet	2012	89.2	0.88
3	ResNet – 50	2016	92.7	0.91
4	CNN (Proposed model)	2021	99	0.99

Table.2. Comparision chart of different algorithms

V. CONCLUSION

The proposed model on detection of face mask is successfully done with the model created with CNN architecture using MobileNetV2 which gave a good result with perfect accuracy of detection. The data were trained and tested for the model to gain good accuracy while detection. The other researchers have many problems in output, using the dataset only some were able to get better accuracy. Wrong predictions have been removed successfully from this model since the dataset used was collected from various other sources and images which have been used in the dataset was pre-processed well to get better accuracy of the result.

VI. REFERENCE

[1] Mohamed Loey , Gunasekaran Manogaran, Mohamed Hamed N. Taha , Nour Eldeen M. Khalifa, “hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic,” College of Information and Electrical Engineering, Asia University, Taiwan Measurement 167 (2021) 108288.

[2] Preeti Nagrath , Rachna Jain , Agam Madan , Rohan Arora , Piyush Kataria , Jude Hemanth , “SSDM V2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2 ,” Department of ECE, Karunya Institute of Technology and Sciences, Coimbatore, India Sustainable Cities and Society 66 (2021) 102692.

[3] Mohamed Loey a, Gunasekaran Manogaran , Mohamed Hamed N. Taha , 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,” ,University of California, Davis, USA Sustainable Cities and Society 65 (2021) 102600.

[4] Walid Hariri, “Efficient Maske Face Recognition Method During The COVID-19 Pandemic,” Labged Laboratory Department of Computer Science Badji Mokhtar Annaba University December 12, 2020

[5] Md. Sabbir Ejaz,Md. Rabiul Islam,Md Sifatullah,Ananya Sarker , “Implementation of Principal Component Analysis on Masked and Non-masked Face Recognition,” 1st International Conference on Advances in Science, Engineering and Robotics Technology 2019 (ICASERT 2019)

[6] Qingcang Yu, Harry H Cheng, Wayne W Cheng and Xiaodong Zhou, “Ch OpenCV for interactive open architecture computer vision”, (2004).

[7] Herong Zheng, Yanlin Lu and Xiaofei Feng, “Improved Compression Algorithm Based on Region of Interest of Face”, (2006).

[8] Datasets - https://github.com/TheSSJ2612/Real-Time-Medical-Mask-Detection/releases/download/v0.1/Dataset.zip