MEDICAL FACE MASK DETECTION USING CNN

[1]Archana Pushparaj, [1]Raji Rajendran, [1]Sophia BJ, [1]Vishnupriya PB, [2]Al shaina.S
[1]UG Scholar, Department of Computer Science and Engineering,
[2]Asst. Prof, Department of Computer Science and Engineering, Dr. APJ Abdul Kalam Technological University, Kerala ,India

Abstract: ***The Transmission of Corona virus disease is affected world-wide. Around 126 million peoples are infected with the Covid 19 virus and it caused 2.9 million deaths. There is no special treatments against the virus, wearing face mask has become an effective method to prevent the transmission of disease. In this situation face mask wearing is mandatory for one and each. There is not any information about the automatic detection of face mask wearing conditions. In our paper we proposed a face mask detection using CNN and deep learning technologies. Our proposed algorithm contains several steps that are 1. Import tensor flow, 2.Import keras, 3.Create model, 4.Create hidden layers, 5. Create output layers, 6. Compile 7.Fit and Set. Our method proposes a medical face mask wearing detector with CNN. The input***
~~image is tested with our trained ML model and results were the image is masked or not. The door opened for those who are masked. And it~~ will be closed for whom that did not wear mask properly.

Keywords: *CNN(convolutional neural network) MMD (Medical masks dataset) FMD (Face mask dataset) NFW (No face mask wearing) IFW (Incorrect face mask wearing) CFW (Correct face mask wearing*

I. INTRODUCTION

The World Health Organization (WHO) is declared that the coronavirus disease 2019 (COVID-19) could be a pandemic. A world coordinated participation is required to prevent the globe wide spread of the virus. Transmission mechanism and Feco- oral transmission is not isn't much documented yet in literature. When someone sneezes or coughs or speaks, that point the virus can travel a distance about 2 meters and it spreads on the surface. Then when a person touches the infected object along with his hand and keeps it on face or eye, the infection spreads to the person. That meaning the most mode of transmission is through respiratory droplets (macro droplet about 5 microns). Micro droplet containing the virus theoretically can travel quite an distance but that theory is yet to be proved. So, a MASK can prevent the spread of respiratory droplets and subsequently the CORONA virus. Facemask wearing can prevent the spread of virus from one person to others. The right wearing of mask can interrupt these viruses and its particles effectively, such that these pathogens cannot enter the human respiratory system. Since the outbreak of COVID-19, facemasks have became mandatory for peoples to scale back the exposure of these pathogens in air. The patients who are suffering from corona virus is required to wear facemasks for the prevention of virus spreading, all over the world everybody need to wear facemasks for protect themselves from infection. And literally face mask become an integral part of our life.

II. BACKGROUND

Medical face mask detection using CNN is an approach consists of four modules. Those are 1.Hyper parameter learning rate 2. Testing on mobileNetV2 3.Testing on single image 4. Testing on video. The main steps are import tensor flow, import keras, create module, create hidden layer, create output layer, finally fit and set. This section describes the technology behind the CNN and face mask wearing condition identification. Object detection is one of the most relevant topics in the field of image processing. CNN can obtain different types of features from data, it can take global training particles, and is successfully implemented in many types of pattern recognition applications. The deep CNN has efficient robustness on scaling, shifting and transposing. In our work, different types of CNN architecture is used here. Facemask-wearing condition identification is divided in to three categories, that are no facemask wearing (NFW), incorrect facemask-wearing (IFW), and correct facemask-wearing (CFW). Our goal is to find a facemask-wearing condition identification function (FWI), the inputs are images or videos. In processing time it checks masked face and unmasked face with facial characteristics and opens the door for who wear mask and the door did not open for who are un-mask or improperly wear masks. Facemask wearing conditions determined by using facial characteristics such as eyes nose lips etc. It has a space in automatic identification of facial characteristics. Recent technologies are involves deep learning and convolutional neural networks in face mask detection. The main characteristics of deep learning methods are 1.supervised, semi-supervised or un-supervised 2. it has a huge amount of resources 3.and large amount of layers in model etc. Face mask detection refers to detect whether a person wearing a mask or not according to facial characteristics, and opens the door for who ware mask. This section describes the technology behind deep learning network and CNN. Deep learning is the process of composition of non-linear transformation of the data. This steps behind deep learning networks are pre-processing, feature extraction, post processing, classifiers (SVM boosting). A simple single layer neural network have a linear combination of input through a non linear function.

III. CONVOLUTIONAL NEURAL NETWORK

The Convolutional Neural Network (CNN) is a class of neural networks that specialize in processing data. CNN is comprised of various convolutional layers, several pooling layers such as min, max or average, non-linear layer like sigmoid, ReLU and classification layer like Soft max units. Deep CNN networks are typically trained on large labelled datasets like Image Net to pick out general features which are appropriate into several detections and recognition jobs like image classification and verification, object detection, segmentation to texture identification. When CNN architecture combines with multiple detectors which can combine different portion of an object. CNN can obtain essential features from the data, can select global training components, and have been successfully implemented to many disciplines of pattern recognition applications. The deep CNN has sufficient robustness on scaling, shifting and transposing. To fulfil our work, we use different forms of CNN architecture.

3.1. Convolutional Layer

A convolutional layer hold a set of filters whose parameters required to be learned. The height and weight of the filters are slight than those of the input volume. Each filter is complexes with the input volume to calculate an activation map construct of neurons. In further words, the filter is glide across the width and height of the input and the dot products connecting the input and filter are calculated at every spatial location. The output volume of the convolutional layer is getting by stacking the activation maps of all filters besides the depth dimension.

3.2. Pooling Layer

A pooling layer is another building block of a CNN. Its purpose is to gradually reduce the spatial size of the portrayal to scale back the amount of parameters and computation within the network. Pooling layer operates on each feature map separately. The most usual approach utilized in pooling is max pooling.

3.3. Fully Connected Layer

Fully connected layer is merely, feed forward neural networks. Fully connected layers create the last few layers in the network. The input to the fully connected layer is the output via the final pooling or convolutional layer, which is demolished and then fed into the fully connected layer. In prominent machine learning modules , the rearmost few layers are fully connected layers which compiles the data bring out by previous layers to form the final output. It is the second most time consuming layer second to convolutional layer.

3.4. Dropout

When all the features are connected to the fully connected layer, it can cause over fitting within the dataset. Over fitting occurs when a specific model works so well on the training data and this cause a negative impact within the performance of the model. To overcome this problem, a dropout layer is utilised where in some neurons are dropped from the neural network during training process leading to reduce size of the model.

3.5.Activation Function

One of the foremost important parameters of CNN model is that the activation function . AF are wont to learn and approximate any reasonably continuous and complicated relationship between variables of the network. It decides which information of the model should fire within the forward direction and which of them mustn't at the tip of the network. It adds nonlinearity to the network. Commonly used activation functions like the ReLU, Softmax, tanH and also the sigmoid functions. Each of those functions have a special purpose usage.

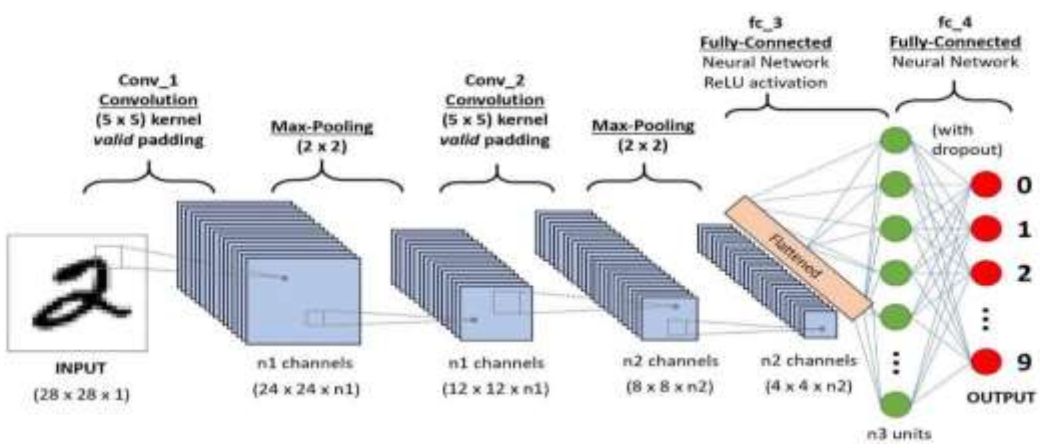

Figure 1 CNN

IV.METHODOLOGY

4.1 Data set

The data set which we have used is RMFD and Kaggle. We are using the real images of people in data set and we couldn't use any morphed or fake images. The data set has 4095 images of two classes, that are 2165 images of with mask and 1930 images of without mask. The image we are collected from RMFD data set, kaggle and Bing search API. Tis model is accurate and we are using Mobile Net V2 architecture. It is also efficient and this making it easier to deploy the model to embedded system. We can divide our dataset into three parts. That are training dataset, test dataset and validation dataset. The advantage of splitting data is to avoid over fitting.

4.2 Architecture

The architecture consists of several steps, it has divided into three parts, that are data processing, testing , evaluation. These steps are explained in the architecture. The steps can be explained as four that are pre-processing, model creation, model training, and model prediction. We can explain this steps by data flow diagram.

- Pre-processing

  Pre-processing was applied to all the raw input images to convert them into clean versions, and it could be fed up in a neural network ML model. The input image is resized to 224 x 224 and it is passed to preprocess_input function, which is meant to adequate your image to the format the model requires. And finally we convert data and labels to NumPy arrays for further processing. An encoding is performed on labels to convert categorical data to numerical data.

- Model creation

  Followed by activation function ReLU(to add nonlinearity) and Max Pooling(to reduce the feature map), CNN model includes two convolutional layers. We add dropout for the prevention of Neural Networks from Over-fitting. Then, fully connected layers were added at the end. Finally, we compile our model to the loss function, the optimizer, and the metrics (Line 36). To find error or deviation in the learning process, we use loss function. Keras require loss function during the model compilation process. By optimization process ,the input weights are optimized by comparing the prediction and the loss function. The metrics were used to evaluate the performance of model.

- Model training

  Before start training of model we need to split the data into train dataset and test dataset. In our case, 90% of the data is for training and 10% is for testing data. The models were trained by NumPy arrays using the fit function. The main purpose of this fit function is to evaluate model on training.

Training Set	Validation Set	Accuracy	Sensitivity	Specificity	Precision	U-Kappa
65 %	35 %	77.29 %	0.617	0.513	0.613	0.5910
70 %	30 %	80.02 %	0.581	0.49	0.5122	0.4871
75 %	25 %	84.23 %	0.645	0.57	0.6347	0.5989
80 %	**20 %**	**94.16 %**	**0.676**	**0.61**	**0.6412**	**0.6134**
85 %	15 %	89.22%	0.690	0.546	0.681	0.6115
90 %	10 %	86.54%	0.541	0.412	0.612	0.5813

Table 1. Cross validation

- Model prediction

  This is the final step, we're we evaluate the models performance by predicting the test data labels

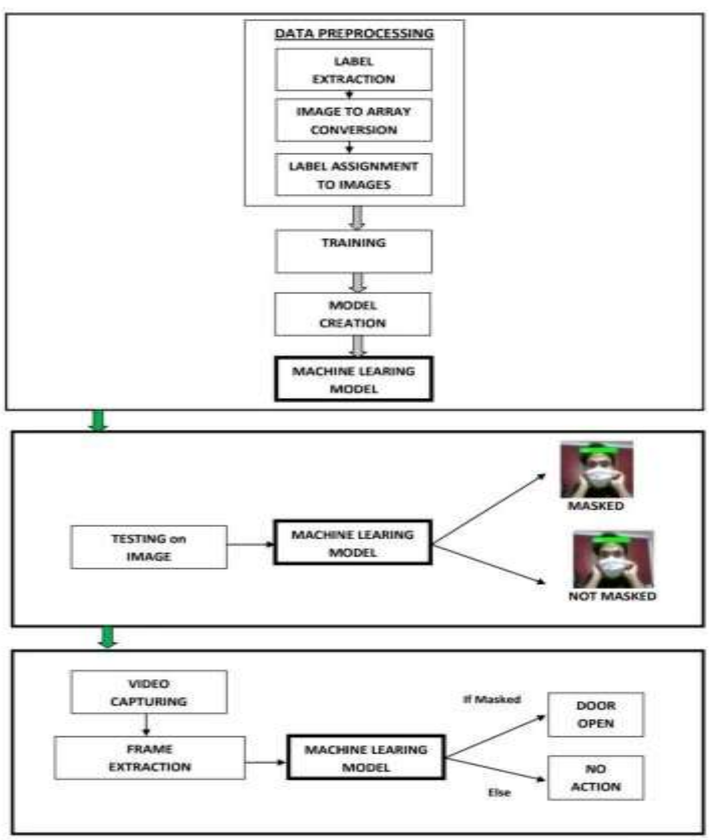

Figure 2.Data flow diagram

4.2.1 Label extraction

Tissue based studies generate imagery and metadata containing biological information, from the histological data. The database stores the extracted label in the structured format, with search capabilities.

4.2.2 Image to array conversion program

```
from PIL import Image
import numpy as np  def
load_image( infilename ) :
img = Image.open(
infilename )  img.load()
  data = np.asarray( img,
dtype="int32" )    return data  def
save_image( npdata, outfilename )
:
 img = Image.fromarray( np.asarray( np.clip(npdata,0,255), dtype="uint8"), "L" ) img.save( outfilename )
```

4.2.3 Label assignment to image

Image labelling is the most crucial part of data preprocessing. To get an accurate model, we have to spend more time in labelling. The availability of label data in the form of training set and test set. These images are used in training the ML algorithm, which provide learning basis of future processing.

MobileNetV2

MobileNetV2 may be a convolutional neural spec that seeks to perform well on mobile devices. it's supported an inverted residual structure where the residual connections are between the bottleneck layers. The intermediate expansion layer uses lightweight depth wise convolutions to filter features as a source of nonlinearity. As a whole, the architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers.

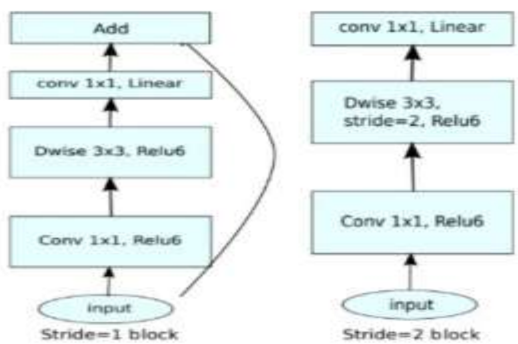

Figure 3. MobileNetV2 V. RESULTS AND DISCUSSION

The implementation and experiments were based on the Tensor Flow deep learning framework. The network was trained and tested on our local machine with an RTX 2070 GPU of 8 GB memory. We have modelled a facemask detector using CNN architecture and transfer learning method in neutral network. For the training validation and testing of the model, we used the dataset that consist of 2165 masked face images and 1930 unmasked faces images. The images of the proposed dataset were collected from multiple public datasets (kaggle and RMFD) as well as images from the Internet. The model was inferred on images and live videos. To select a base model, some metrics such as the precision(p), recall® and accuracy (a) were used. The selected MobileNetV2 architecture with the best performance having 100% precision and 99% recall. It is also computationally efficient using mobile net v2 which makes it easier to install the model to embedded systems.

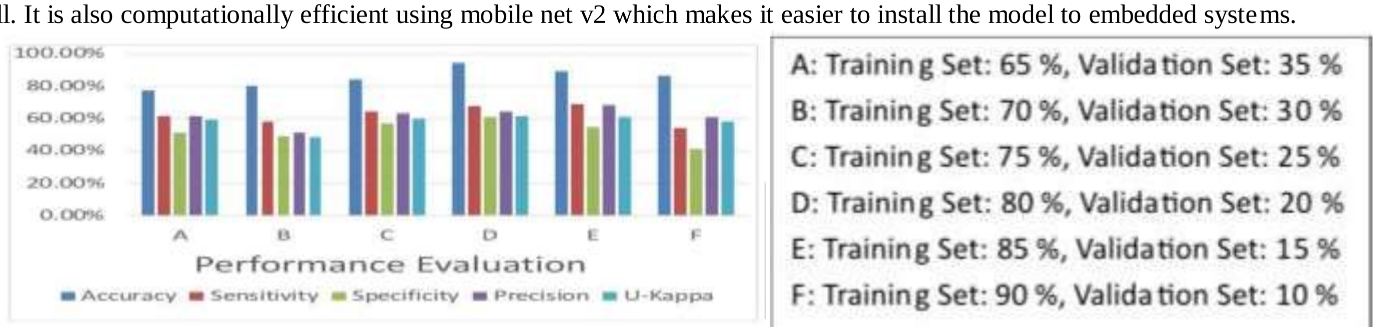

Figure 4.Performance evaluation

Sensitivity =TP / TP+FN

Specificity =TN / TN+FP

Precision = TP / TN+FN

Accuracy = TP+ FN / Total no.of test samples

The input image is tested with our trained ML model and resulted whether the image is masked or not. The door will be opened, for those who are masked. And it will be closed for whom were not wearing mask properly. The steps included in evaluation is that preprocessing, model creation, model training, model prediction. Model hyper parameters are parameters that are more involved in the architecture or structure of the model. They help us to define our model complexity based on the different layers like the input layer, hidden layer, and output layer of a neural network. Initially, we trained with different values of hyper parameters by changing on and keeping the other constant and noted down the results in each case. We selected the hyper parameters that produced better performance through evaluation metrics. We have chosen the hyper parameters as follows: initial learning rate is taken as 0.0001, batch size is taken to be 32 and number of epochs as 20. In our case, the target size is also one of the hyper parameters which we kept (224, 224, 3) as it is default input shape of MobileNetV2.

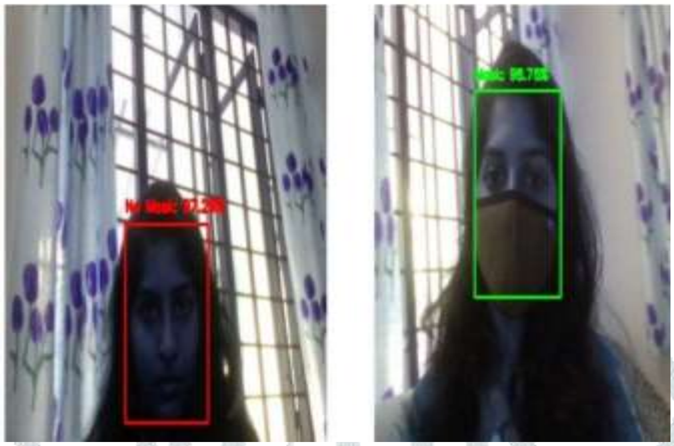

Figure 5.Output screen short

VI. CONCLUSION

To mitigate the spread of covid19 pandemic, measure must be taken. Using CNN architecture we have modelled a facemask detector and transfer learning methods in neural networks. For the train, test and validation of the model, we used the dataset consist of 2165 masked faces images and 1930 unmasked faces images. These images were taken from various datasets like kaggle and RMFD. The model were inferred on images and live video streams. To select a base model, we evaluated the metrics like accuracy precision and recall And selected mobile net v2 architecture with the best performance having 100% precision and 99% recall. It is also computationally efficient using MobileNetV2 which makes it easier to install the model to embedded systems. This facemask detector can be deployed in many areas like shopping malls, airports and other heavy traffic places to monitor and to check who is following basic rules and who is not. When the mask is detected the door is opens otherwise no changes.

VII. ACKNOWLEDGEMENT

We would wish to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to our project guide Ms. Al shaina.S, Assistant Professor in Computer science and Engineering Department who gave valuable suggestion and guidance for our project. We express our deep felt gratitude to beloved HOD Dr. Ramani K, Head of the Department for providing necessary information regarding the project and also her support in completing it. We also thank our project co-ordinator, Mr. Jithin Jacob, assistant professor who gave expert supervision, encouragement and constructive criticism amist his busy schedule throughout the project. We also grateful to all the authors of books and papers which have been referred to publish this paper.

REFERENCES

[1] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, vol. 1. IEEE, 2001, pp. I–I.

[2] R.Girshick,J.Donahue,T.Darrell,andJ.Malik,“Richfea turehierarchiesforaccurateobjectdetec tionand semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[3] R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448. [4] S. Ren, K. He, R. Girshick, and J. Sun, “Faster rcnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, 2015, pp. 91–99.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in European conference on computer vision. Springer, 2016, pp. 21–37.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, realtime object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779– 788.

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Doll´ar, “Focal loss for dense object detection,” 2017. [8] Haddad, J., 2020. How I Built A Face Mask Detector For COVID-19 Using Pytorch Lightning.

[9]Rosebrock,A.,2020. COVID-19: FaceMaskDetectorWithOpencv,Keras/Tensorflow,AndDeepLearning - Pyimagesearch. [online] PyImageSearch. Available at: https://www.pyimagesearch.com/2020/05/04/covid19fa ce-mask-detector-with-opencv-keras-tensorflow-anddeeplearning/ .

[10] W.Liu,D.Anguelov,D.Erhan,C.Szegedy,S.Reed,C.- Y.Fu,andA.C.Berg,“Ssd: Singleshotmultibox detector,” in European conference on computer vision. Springer, 2016, pp.

[11] Francois Chollet “Xception: Deep Learning with Depthwise Separable Convolutions” in Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR), 2017, pp. 1251-1258.

[12] M.Sandler,AndrewHoward,MenglongZhu,AndreyZ hmoginov,andLiangChiehChen,“MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR), 2018.