LAB ASSIGNMENT-3

FACE MASK DETECTION USING CNN

BY,

MADASU DEEPIKA
19MIA1066

Abstract:

In recent trend, In world-wide lockdowns has been imposed due to COVID19 outbreak and Face Mask became mandatory for everyone while roaming outside.
This paper approaches Deep Learning for Detecting Faces With and Without mask.

While going through the pandemic and the post pandemic situations wearing a mask are compulsory for everyone in order to prevent the transmission of corona virus. This resulted in ineffectiveness of the existing conventional face recognition systems.

Objective:

The main aim is to identify that whether a person's face is covered with mask or not as per the CCTV camera surveillance or a webcam recording. It keeps on checking if a person is wearing mask or not. For classification, feature extraction and detection of the masked faces, Convolutional Neural Network (CNN) are used.

These help in easy detection of masked faces with higher accuracy in a very less time and with high security.

Keywords: Covid-19, Deep Learning, Face mask, Convolution Neural Network.

Introduction:

Face mask detection means to identify whether a person is wearing a mask or not. The first step to recognize the presence of a mask on the face is to detect the face, which makes the strategy divided into two parts: to detect faces and to detect masks on those faces. Face detection is one of the applications of object detection and can be used in many areas like security, biometrics, law enforcement and more.

To monitor that people are following this basic safety principle, a strategy should be developed.

A face mask detector system can be implemented to give accurate results if set up with a CCTV camera to track people without masks to ensure the safety and wellbeing of others, thus help controlling the spread of the virus.

Dataset:

Data set consists of 7553 RGB images in 2 folders as withmask and withoutmask. Images are named as label withmask and withoutmask.

Images of faces with mask are 3725 and images of faces without mask are 3828.

All the images are actual images extracted from Bing SearchAPI, Kaggle datasets, and RMFD datasets. From all three sources, the proportion of the images is equal. The images cover diverse races i.e Asian, Caucasian, etc.

Dataset link: Face Mask Detection Dataset | Kaggle

Fig-1 Some images of with mask dataset

Fig-2 Some images of without mask dataset

Methodology:

We need to split our dataset into three parts: training dataset, test dataset, and validation dataset. The purpose of splitting data is to avoid overfitting which is paying attention to minor details/noise which is not necessary and only optimizes the training dataset accuracy. The training set is the actual subset of the dataset that we use to train the model. The model observes and learns from this data and then optimizes its parameters.

The test set is there a remaining subset of data used to provide an unbiased evaluation of a final model fit on the training dataset. Data is split as per a split ratio which is highly dependent on the type of model we are building and the dataset itself.

In our approach, we have dedicated 80% of the dataset as the training data and the remaining 20%as the testing data, which makes the split ratio as 0.8:0.2 of the train to test set. Out of the training data, we have used 20% as a validation data set.

A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor. By using the sequential method, the architecture of the model is built layer by layer, convolution layer, maxpool layer, flatten layer, drop out, dense layer.

Architecture:

The model is created with the following layers.

□ The input image is grayscale. since there is only one channel.

□ The size of the image is 48 *48

□ Convolution layer with the images as the input, with the ‘relu’ activation Function, padding is valid.

□ Next, the Maxpooling layer is created with the size of 2*2 and the stride of 2*2, with the valid padding.

□ Then the output of the max-pooling layer is normalized with the Batch normalization layer.

□ In the middle of the layers, a Dropout is used to overcome the overfitting of the model.

□ In this way, another 2 layers of convolution, max pooling followed by batch normalization is done.

□ Then the flattened layer is implemented. This layer flattens all the outputs of the Batch Normalization layer.

□ Then the flattened output is given to the next created dense layer.

□ The Dense Layer is created after the flatten this will create a fully connected layer that is used for the classification.

□ Then the drop-out layer is created to reduce the number of outputs from layer to layer.

□ Then a Batch Normalization is created.

□ In this way another 2 layers of dense, dropout followed by Batch normalization is created.

□ Then at last the last layer a dense layer is created with the 2 neurons, with the activation function ‘Softmax’.

□ The last layer gives the output which specifies one of the 2 classes i.e., whether a person is with a mask or without a mask.

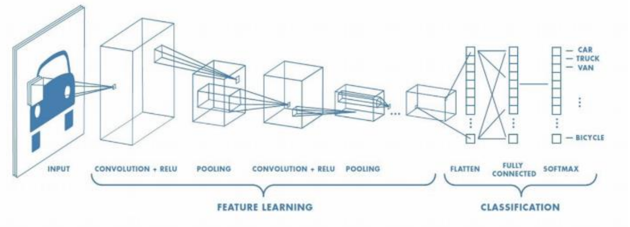
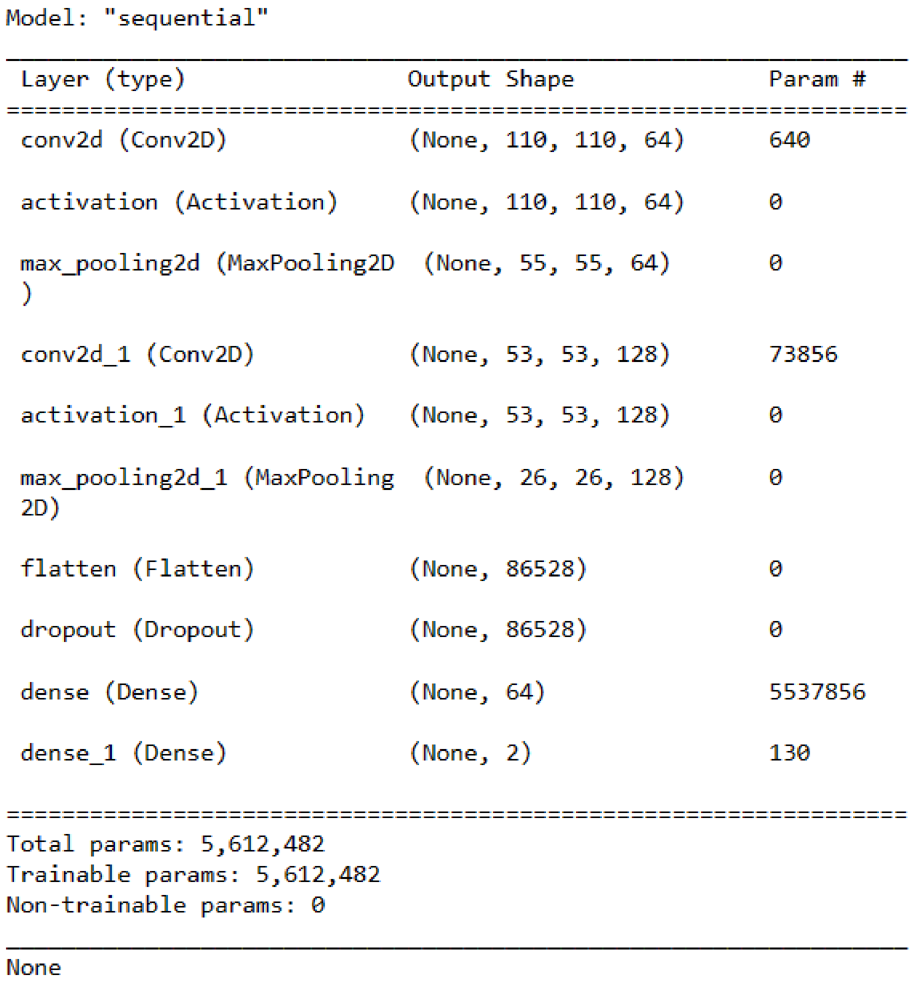


Figure 3

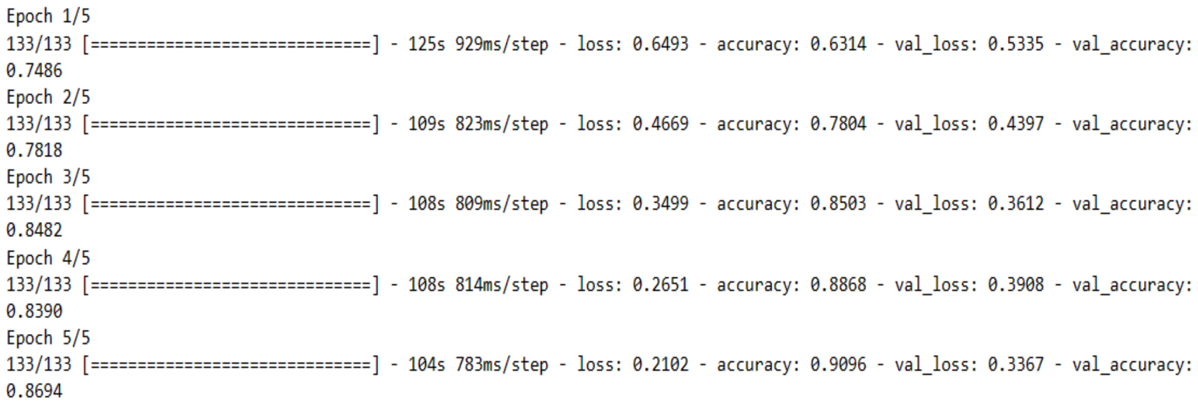
CNN Model:

Pertaining to our CNN model, we trained a CNN with 10 layers and 5 epochs. A summary of the model and training accuracy across each epochs can be found in Figures 4 and 5.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 110, 110, 64)      640       
                                                                 
 activation (Activation)     (None, 110, 110, 64)      0         
                                                                 
 max_pooling2d (MaxPooling2D  (None, 55, 55, 64)       0         
 )                                                               
                                                                 
 conv2d_1 (Conv2D)           (None, 53, 53, 128)       73856     
                                                                 
 activation_1 (Activation)   (None, 53, 53, 128)       0         
                                                                 
 max_pooling2d_1 (MaxPooling  (None, 26, 26, 128)      0         
 2D)                                                             
                                                                 
 flatten (Flatten)           (None, 86528)             0         
                                                                 
 dropout (Dropout)           (None, 86528)             0         
                                                                 
 dense (Dense)               (None, 64)                5537856   
                                                                 
 dense_1 (Dense)             (None, 2)                 130       
                                                                 
=================================================================
Total params: 5,612,482
Trainable params: 5,612,482
Non-trainable params: 0
_________________________________________________________________
None
```

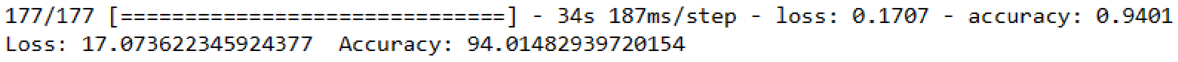
Figure 4



```
Epoch 1/5
133/133 [==============================] - 125s 929ms/step - loss: 0.6493 - accuracy: 0.6314 - val_loss: 0.5335 - val_accuracy:
0.7486
Epoch 2/5
133/133 [==============================] - 109s 823ms/step - loss: 0.4669 - accuracy: 0.7804 - val_loss: 0.4397 - val_accuracy:
0.7818
Epoch 3/5
133/133 [==============================] - 108s 809ms/step - loss: 0.3499 - accuracy: 0.8503 - val_loss: 0.3612 - val_accuracy:
0.8482
Epoch 4/5
133/133 [==============================] - 108s 814ms/step - loss: 0.2651 - accuracy: 0.8868 - val_loss: 0.3908 - val_accuracy:
0.8390
Epoch 5/5
133/133 [==============================] - 104s 783ms/step - loss: 0.2102 - accuracy: 0.9096 - val_loss: 0.3367 - val_accuracy:
0.8694
```

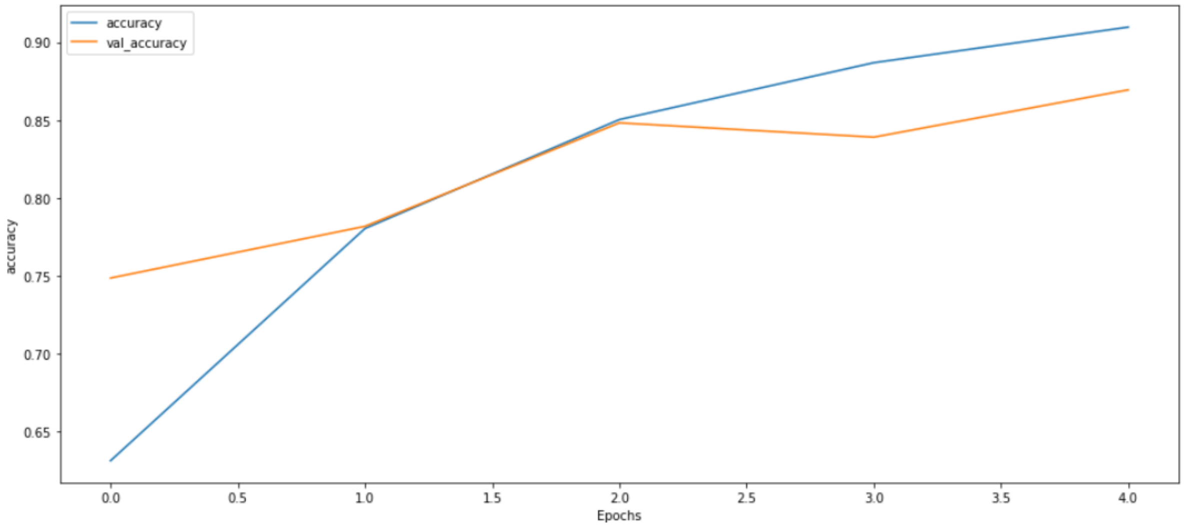
Figure 5

Result:

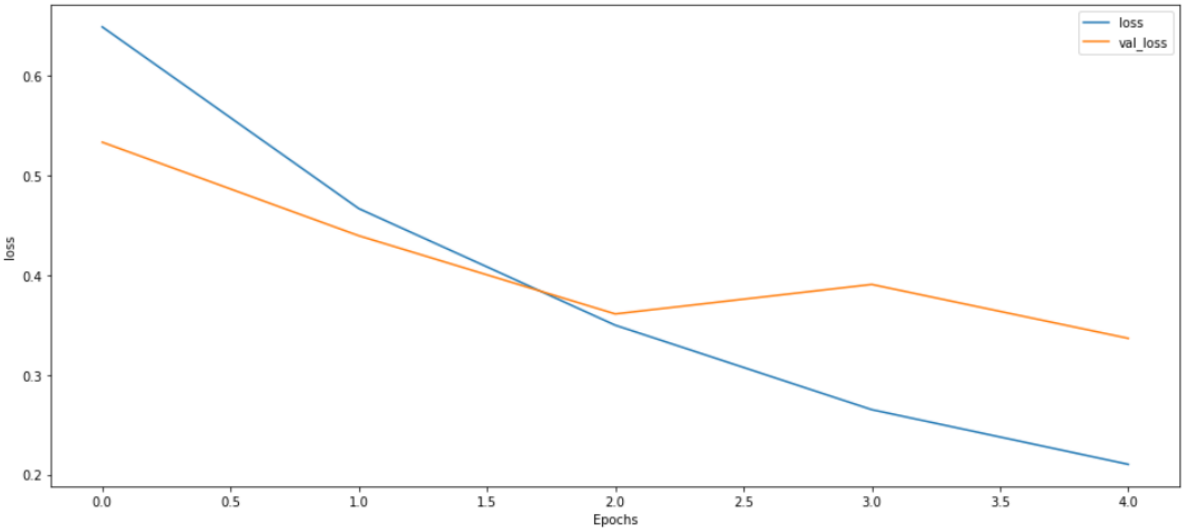


```
177/177 [==============================] - 34s 187ms/step - loss: 0.1707 - accuracy: 0.9401
Loss: 17.073622345924377  Accuracy: 94.01482939720154
```

Training and Validation accuracy:



Training and Validation loss:



As the model trains, the final training loss and accuracy metrics are displayed. This model reaches an accuracy of about 94.01% on the training data and the loss of 17.07%.

Conclusion:

In this paper, we implemented an application to develop face mask detection using the Deep Convolutional Neural Network model. That is to build a CNN-based framework to precisely match both still images or moving real-time video from the web camera and then check whether a person is wearing a mask or not.

The model should hopefully help the concerned authorities in this great pandemic situation which had largely gained roots in most of the world; other researchers can use the dataset provided in this paper for further advanced models such as those of face recognition, facial landmarks, and facial part detection process.

References:

- Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C.…Ghemawat S. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467. [Google Scholar]
- Ahmed I., Ahmad M., Rodrigues J.J., Jeon G., Din S. A deep learning-based social distance monitoring framework for COVID-19. *Sustainable Cities and Society.* 2020 [PMC free article] [PubMed] [Google Scholar]
- Alom M.Z., Taha T.M., Yakopcic C., Westberg S., Sidike P., Nasrin M.S.…Asari V.K. 2018. The history began from alexnet: A comprehensive survey on deep learning approaches. arXiv preprint arXiv:1803.01164. [Google Scholar]
- Anisimov D., Khanova T. Towards lightweight convolutional neural networks for object detection. 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS); IEEE; 2017. pp. 1–8. August. [Google Scholar]
- Chen D., Ren S., Wei Y., Cao X., Sun J. *European conference on computer vision.* Springer; Cham: 2014. Joint cascade face detection and alignment; pp. 109–122. September. [Google Scholar]
- Chen D., Hua G., Wen F., Sun J. *European conference on computer vision.* Springer; Cham: 2016. Supervised transformer network for efficient face detection; pp. 122–138. October. [Google Scholar]
- Ge S., Li J., Ye Q., Luo Z. Detecting masked faces in the wild with lle-cnns. *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017:2682–2690. [Google Scholar]
- Ge X.Y., Pu Y., Liao C.H., Huang W.F., Zeng Q., Zhou H.…Chen H.L. Evaluation of the exposure risk of SARS-CoV-2 in different hospital environment. *Sustainable Cities and Society.* 2020;61 [PMC free article] [PubMed] [Google Scholar]
- Ghiasi G., Fowlkes C.C. Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2014:2385–2392. [Google Scholar]
- Gulcehre C., Moczulski M., Denil M., Bengio Y. *International conference on machine learning.* 2016. Noisy activation functions; pp. 3059–3068. June. [Google Scholar]