Ariel University

Department of Computer Science

Final report | Face Mask Detection

Written and coded by:

Mulugeta Fanta, Ortal Hanoch, and Tomer Maabari

Ariel University, Israel

Course lecturer:
Amos Azaria, Ph.D.

Abstract

After the breakout of the worldwide pandemic COVID-19, there arises a severe need of protection mechanisms, face mask being the primary one. The basic aim of the project is to detect the presence of a face mask on human faces on live streaming video as well as on images. We have used deep learning to develop our face detector model. The algorithm used for the object detection purpose is Haar-Cascade because of its good performance accuracy and high speed. Alongside this, we have used basic concepts of transfer learning in neural networks to finally output presence or absence of a face mask in an image or a video stream. Experimental results show that our model performs well on the test data.

Introduction

The year 2020 has shown mankind some mind-boggling series of events amongst which the COVID19 pandemic is the most life-changing event which has startled the world since the year began. Affecting the health and lives of masses, COVID-19 has called for strict measures to be followed to prevent the spread of disease. From the very basic hygiene standards to the treatments in the hospitals, people are doing all they can for their own and the society's safety; face masks are one of the personal protective equipment. People wear face masks once they step out of their homes and authorities strictly ensure that people are wearing face masks while they are in groups and public places. To monitor that people are following this basic safety principle, a strategy should be developed. A face mask detector system can be implemented to check this. 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. There are many detector systems developed around the world and being implemented. However, all this science needs optimization; a better, more precise detector, because the world cannot afford any more increase in corona cases. In this project, we will be developing a face mask detector that is able to distinguish between faces with masks and faces with no masks. In this report we used neural network to detect presence of a masked faced.

Background related work

The Viola Jones Face Detector which used algorithm Haar- cascaded features is one of the most widely used face detector.

However, with the advent of deep learning, CNN based face detectors are being widely used by the research community and industries alike.

CNN based models learn face representations from the annotated data. These models are then capable of identifying faces and suggesting facial landmarks in the input images.

With the increase in the importance of wearing face masks during the COVID-19 pandemic, there are a plethora of blogs and social media posts that describe approaches to the mask detection problem. However, most of these approaches do not provide access to a reasonably large dataset that works for real world use cases. Most of the work available uses smaller datasets with less variability in terms of types of masks (standard N-95, surgical masks) or only look at specific regional datasets that have been scraped from the web.

Description

After collecting an array of images, the array includes number of images for the training process.

The images rotate and flip, the data is divided into a train and a test is performed to learn about the machine. Loading the dataset from pre-processing data and then the conventional architecture defined for building the neural network in MobileNetV2.

With all the data that had been gathered and configured, the machine is trained to detect masked face, by using ADAM optimizer.

After the model is trained, the data as being saved into the model.

By using OpenCV library to run an infinite loop to use the web camera in which it detect the face using Haar-Cascade classifier algorithm, the model will predict possibilities of each the two classes (masked and unmasked). Based on which the probability is higher, then the square will be chosen and displayed around the face.

Limitations

There were two problems that were time consuming and made the tasks tedious are discussed as follows.

One of them used CNN, the results were inaccurate.

After 50 iterations the accuracy was only 90% and for learning a particularly large image dataset was needed.

Using MoblieNetV2 enabled faster learning using fewer images and achieving better results.

Secondly, at first the pictures were of high quality and the same size. This caused the camera to be unable to detect a mask because it was required to be of such high quality as well. Therefore, it was necessary to lower the quality of the images and create a variety of sizes to achieve good deep learning and better results.

Conclusion

To mitigate the spread of COVID-19 pandemic, measures must be taken. We have modeled a face mask detector using We have used deep learning to develop our face detector model. To train, validate and test the model, we used the dataset that consisted of 702 masked faces images and 695 unmasked faces images. Total of 1397.

These images were taken from various resources like Kaggle. The model was inferred on images and live video streams.

After dana augmentation (rotate and flip images) we have total of 2794 images.

1390 unmasked faces images and 1404 masked faces images.

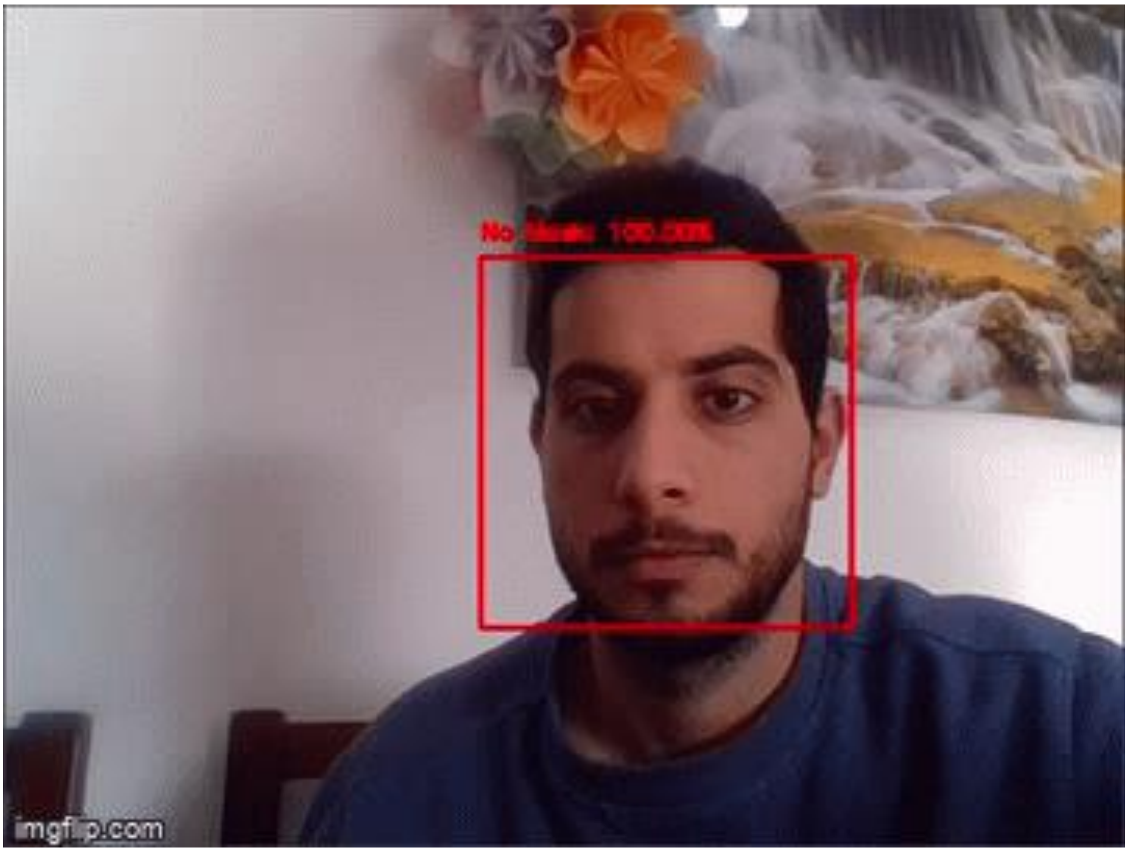
To select a base model, we evaluated the metrics like accuracy, precision and recall and selected MobileNetV2 architecture with the best performance having 100% precision and 99% recall. It is also computationally efficient using MobileNetV2 which makes it easier to install the model to embedded systems.

This face masks detector can be deployed in many areas like shopping malls, airports, and other heavy traffic places to monitor the public and to avoid the spread of the disease by checking who is following basic rules and who is not.

Experiments and Simulation Results

The experimental result of system performance is evaluated with the MobileNetV2 classifier ADAM optimizes. (ADAM is a gradient descent with some enhancements. It uses the moving average of the previous gradients + normalizing by root mean squared error).

As the technology is blooming with the emerging of new trends, we have innovative face mask detector which can possibly contribute to public health care department. The architecture consists of MobileNetV2 classifier and ADAM optimizer as the backbone it can be used for high and low computation scenarios. Our face mask detection is trained on CNN model and we used Open CV, TensorFlow and Keras to detect whether person is wearing a mask or not. The model was tested with image and real time video stream. The accuracy of model is achieved, and the optimization of the model is in continuous process. This specific model could be used as a use case of edge analytics.



Compilation screen for training script of face mask detection:

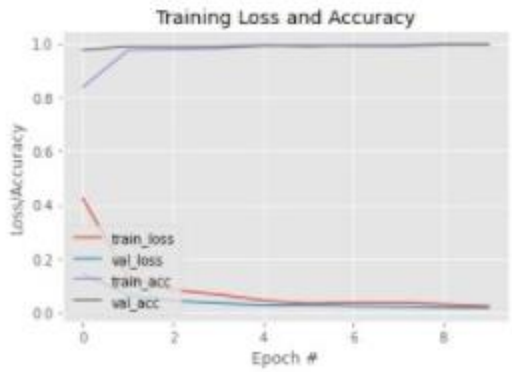
```
[INFO] training head...
Epoch 1/10
55/55 [==============================] - 90s 2s/step - loss: 0.5967 - accuracy: 0.7018 - val_loss: 0.1432 - val_accuracy: 0.975
0
Epoch 2/10
55/55 [==============================] - 82s 1s/step - loss: 0.1493 - accuracy: 0.9762 - val_loss: 0.0644 - val_accuracy: 0.989
3
Epoch 3/10
55/55 [==============================] - 83s 2s/step - loss: 0.0982 - accuracy: 0.9764 - val_loss: 0.0440 - val_accuracy: 0.989
3
Epoch 4/10
55/55 [==============================] - 81s 1s/step - loss: 0.0652 - accuracy: 0.9874 - val_loss: 0.0343 - val_accuracy: 0.989
3
Epoch 5/10
55/55 [==============================] - 81s 1s/step - loss: 0.0450 - accuracy: 0.9897 - val_loss: 0.0272 - val_accuracy: 0.992
9
Epoch 6/10
55/55 [==============================] - 81s 1s/step - loss: 0.0402 - accuracy: 0.9913 - val_loss: 0.0275 - val_accuracy: 0.989
3
Epoch 7/10
55/55 [==============================] - 82s 1s/step - loss: 0.0335 - accuracy: 0.9922 - val_loss: 0.0229 - val_accuracy: 0.992
9
Epoch 8/10
55/55 [==============================] - 83s 2s/step - loss: 0.0322 - accuracy: 0.9906 - val_loss: 0.0207 - val_accuracy: 0.992
9
Epoch 9/10
55/55 [==============================] - 82s 1s/step - loss: 0.0337 - accuracy: 0.9946 - val_loss: 0.0181 - val_accuracy: 0.996
4
Epoch 10/10
55/55 [==============================] - 82s 1s/step - loss: 0.0182 - accuracy: 0.9963 - val_loss: 0.0167 - val_accuracy: 0.996
4
```

Classification report:

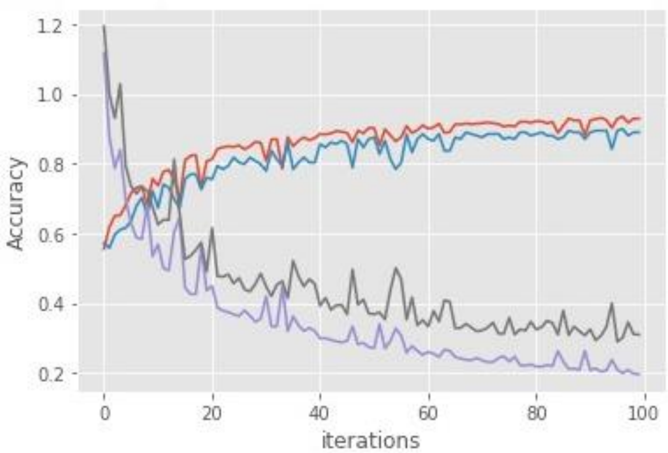
```
[INFO] evaluating network...
              precision    recall  f1-score   support

   with_mask       1.00      0.99      1.00       141
without_mask       0.99      1.00      1.00       139

    accuracy                           1.00       280
   macro avg       1.00      1.00      1.00       280
weighted avg       1.00      1.00      1.00       280
```

Training Loss/Accuracy:

```
Out[25]: <matplotlib.legend.Legend at 0x1debdf0ef10>
```

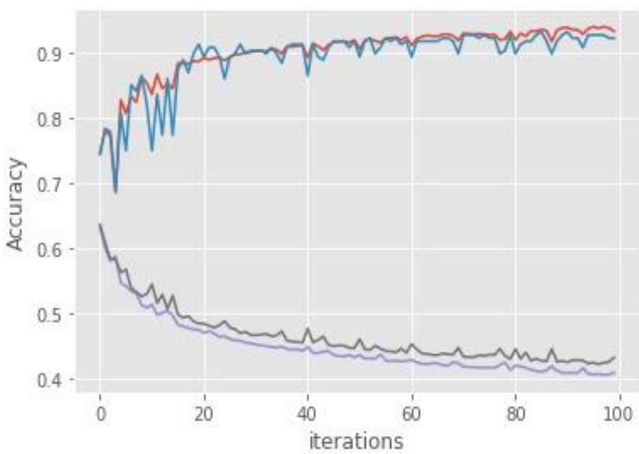


Logistic Accuracy




```
Train Accuracy: 0.93
Test Accuracy:0.89
```

SoftMax Accuracy




```
Train Accuracy: 0.93
Test Accuracy:0.92
```