Report for Artificial Intelligence

Simone D'Antimo, Francesco Caldivezzi, Harjot Singh

April 11, 2022

1 Introduction

Coronavirus disease (COVID-19) has significantly affected the daily life activities of people globally. To prevent the spread of COVID-19, the World Health Organization has recommended people to wear face masks in public places. Manual inspection of people for wearing face masks in public places is a challenging task. This introduces an urgent need to develop a system capable of detecting the people not wearing the face masks in a simple and autonomous way. For this reason we a system that can recognize if a person is wearing a face mask using Deep learning and OpenCV libraries.

2 Requirements

In the following section, it is described what are the main libraries and tools used to develop the project.
For what concerns the software requirements, this project was pretty much developed by the use of :

- OpenCV
- Tensorflow (GPU version)
- Jupyter Notebook
- Nvdia CUDA Toolkit

Instead, for the hardware specifications :

- GPU : Nvdia GTX 970
- RAM : 16 GB
- CPU : Intel Core i5 4690

3 Dataset

The dataset used for this project (available at : here) contains mainly a total of 3832 images which are divided into :

- 1914 images with masks
- 1918 images without masks

In addition to that, images with masks and without masks represent different persons.

4 Fine Tuning

When dealing with Images and especially when, when solving classification problems, most of the time Convolutional Neural Network is a good choice to start with or even the only one that can reach a reasonable amount of performances. However building from scratch a new CNN model is something that takes very hard effort both in terms of time and also in terms of computational requirements.
Therefore, a solution to the problem explained above, is to use what it is called "Transfer Learning", that means, instead developing a new model, and the train it, we take an already trained model that works well on some training set for a certain task and then, we adapt it to solve our problem.
Even though with Transfer Learning, the issue of creating a new model is no more present, there exists still the difficulty of re-training the entire Neural Network on a new dataset to tune the network chosen to solve another one.
So, what nowadays is common to do is to :

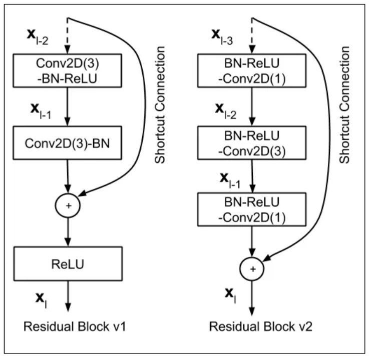
- Select a pre-trained model in the literature that already was trained using a big dataset, and that seems to solve a problem similar to the one that need to be addressed.
- Remove the last part of it and replace it with new fully connected layers.
- Do not allow the not fully connected layers to be trainable.
- Train the Network with the new dataset.

Such process is what it is also known as Fine Tuning and it is the technique that was used in this project.

5 The Model Used For this project : ResNet50V2

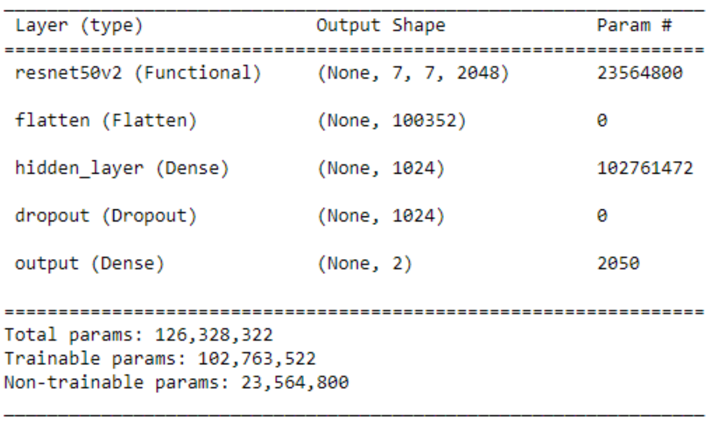
In this project it was used as starting point the ResNet50V2 CNN network (available on Tensorflow library). The choice of the use of such network is due to some mainly two reasons :

- The ResNet50 (V1) Network contrary to what is said in [reference paper 3] was performing poorly in the dataset used for our training if fine tuned
- The ResNet50 (V1) Netwoork was able to learn something on the dataset used only if it was trained from scratch (no fine tuning) on the entire dataset but, due to low hardware performances available the training on the entire dataset was carried out only for one epoch before running out of both RAM and GPU RAM memories.



So, for the reason explained above it was chosen the ResNet50V2 and it was fine tuned by removing its fully connected layers and replaced them with other two. To be precise:

- A layer with 1024 neurons with Relu activation function.
- A layer with 2 neurons with Sigmoid activation function.



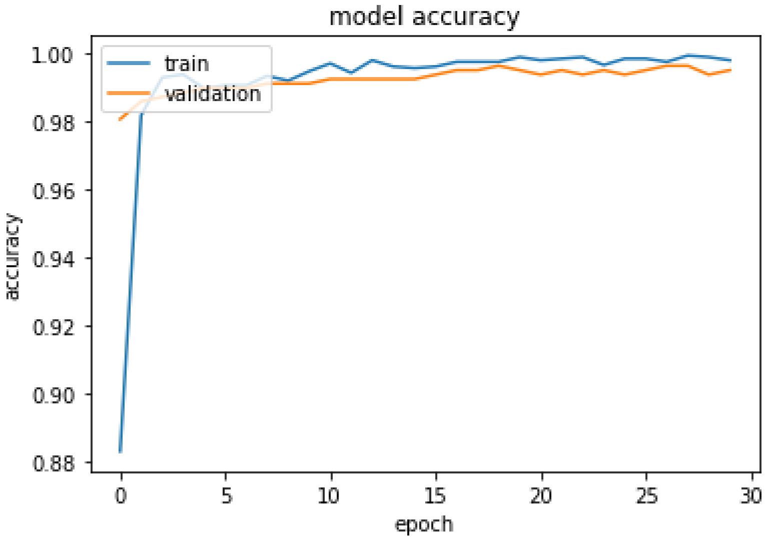
```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50v2 (Functional)     (None, 7, 7, 2048)        23564800

 flatten (Flatten)           (None, 100352)            0

 hidden_layer (Dense)        (None, 1024)              102761472

 dropout (Dropout)           (None, 1024)              0

 output (Dense)              (None, 2)                 2050

=================================================================
Total params: 126,328,322
Trainable params: 102,763,522
Non-trainable params: 23,564,800
_________________________________________________________________
```

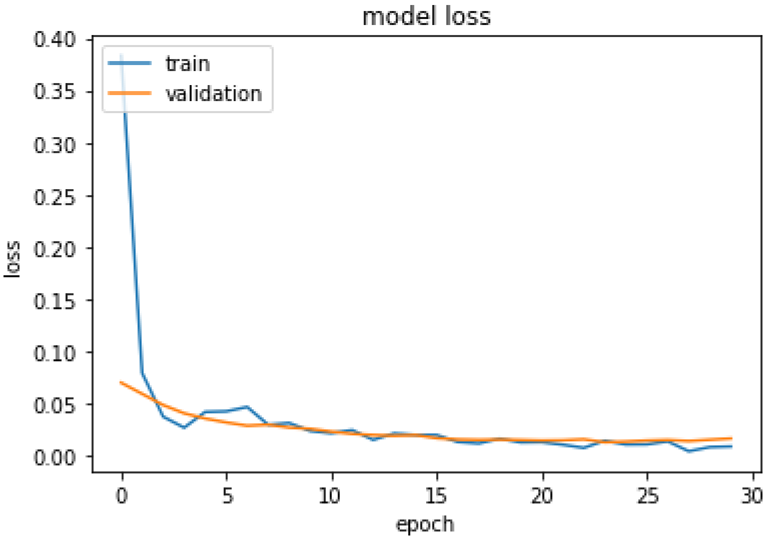
In addition to that, to avoid over fitting issues some regularization techniques were applied. The most important ones are :

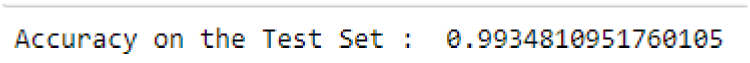
- Early Stopping on the validation loss

- Data Augmentation on the batch items
- Validation on hyper parameters of the network

6 Experimental Results







```
Accuracy on the Test Set :  0.9934810951760105
```

7 Program

The program we have implemented had 2 important parts: The first one is to recognize the face of a person, and to do that we use python with opencv libraries.

The second part of our project regards the model we use to estimate if a person is wearing a mask. We used the RES-Net 50 convolution neural network to create and train our model. Since the model created is very heavy we had to fine-tune it in order to implement it in our program.

The RES-Net 50 model that we created works really well with the dataset that we have used. We split data in 60 - 20 - 20 train-validation-test and the accuracy of the test set is 0.98.

8 Conclusion

This paper has presented a framework for accurate face mask detection and masked facial recognition. This model for face mask detection and masked facial recognition has a good accuracy under diverse conditions i.e., variations in face angles, lightning conditions, gender, skin tone, age, types of masks, occlusions (glasses), etc. The proposed method can be utilized for a variety of purposes and can be integrated with other tools (like instruments for measuring body temperature) in order to maximize the safety of people.

9 Future works

Since our model is heavy, also the fine-tuned one, it is possible that not all the devices are able to compute it in a reasonable amount of time. In order to develop a program that may be used in a more heterogeneous amount of device we have found other models, thinner than Resnet50, that may be implemented to scan faces faster. This model is called MobileN.

10 Bibliography

1. Face mask recognition system using CNN model .

2. A novel DeepMaskNet model for face mask detection and masked facial recognition.

3. COVID-19 Face Mask Recognition with Advanced Face Cut Algorithm for Human Safety Measures.

4. Faster Region-based Convolutional Neural Network for Mask Face Detection .

5. Mask wearing detection method based on SSD-Mask algorithm.

6. Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread.

7. The Face Mask Detection For Preventing the Spread of COVID-19 at Politeknik Negeri Batam.