

**Deep**

**Learning**

**Project – Mask or No Mask**

**Spring Semester 2019/2020**

**Group 8**

**Bernardo Mendes Ferreira M20190450**

**Natallia Sysonava – M20190552**

**Pedro Miguel Reis M20180428**

****

**Index**

[1. Introduction 3](#_Toc42714455)

[2. Problem in Hand 3](#_Toc42714456)

[3. Approach 3](#_Toc42714457)

[3.1. Dataset Description 3](#_Toc42714458)

[3.2. Pre-Processing 4](#_Toc42714459)

[3.3. Evaluation Measures 4](#_Toc42714460)

[3.4. First Model 4](#_Toc42714461)

[3.4.1. Description 4](#_Toc42714462)

[3.4.2. Results 4](#_Toc42714463)

[3.5. Model with Pre-Trained Neural Network 4](#_Toc42714464)

[3.5.1. Description 4](#_Toc42714465)

[3.5.2. Results 4](#_Toc42714466)

[3.5.3. Comparison with first model 4](#_Toc42714467)

[3.5.4. Comparison using adam optimizer 4](#_Toc42714468)

[3.6. Model Selection 4](#_Toc42714469)

[3.7. Model Generalization 4](#_Toc42714470)

[4. Conclusions 5](#_Toc42714471)

# Introduction

The following project was carried out within the scope of Deep Learning subject in the Masters in Data Science and Advanced Analytics in Nova-IMS University.

The main idea of the project was to apply some of the topics covered along the course to address a particular problem chosen by the group’s project.

# Problem in Hand

Nowadays due to Covid World Pandemic situation, the use of masks by the population in closed public states is mandatory. For this reason, it was thought that a Deep Learning Model that through images (photos) identified the use of masks by people was relevant to this context. The model can be an input in the evaluation and monitoring of population’s behavior and adaptation to this new reality and can helped in the decision making of the measure’s rigidity of containment and restriction currently imposed.

Basically, the present model will receive as input a series of images of people and will predict whether that person is wearing a mask or not as stated in **Figure 1**.

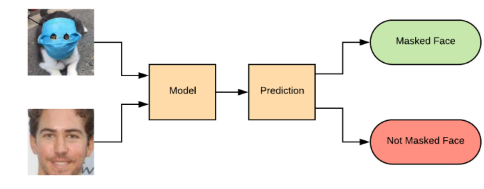
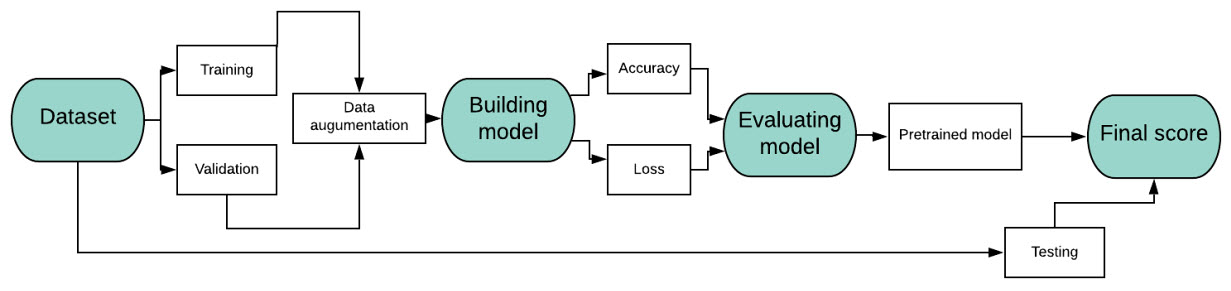


Figure 1 - Model main purpose.

# Approach

### Methodology

Our model approach is present in the diagram below.



### Dataset Description

A kaggle dataset was used in order to get the input images. The dataset consist of 2 folders with images of people faces with mask and without mask as stated in **Figure 2**. There were 5186 images with faces with mask and 4271 images with faces without mask.

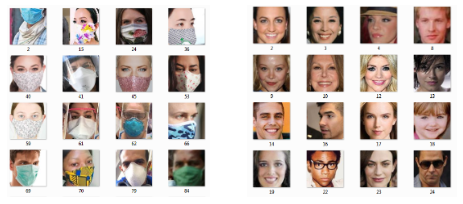


Figure 2 - Dataset description.

### Dataset Pre-Processing

Pre-processing of data was basically divided into two steps. The first one, manually it was deleted all images with less than 96\*96-pixel size. This dimension was used for two reasons: first the dataset had some images too small (which would hurt model’s prediction) and the pre-trained CNN architectures usually required multiples of 32 pixels.

Afterwards the dataset was divided into three categories: training set (70%), validation set (15%) and test set (15%). In order to monitor during training the accuracy of the model on data it has never seen before, it was considered a validation set. In order to achieve this step a specific package was used called **split-folders**. Each split was put in three different directories.

Finally, all images were automatically resized to 96x96 pixels with the help of PIL Image module.

### Evaluation Metrics

Since the current project is a binary classification problem and the output of the network is a probability the **binary\_crossentropy** loss was chosen. For each model the following metrics are collected: accuracy, F1 score, precision and recall. As dataset is pretty balanced, we state accuracy as final metric of model evaluation.

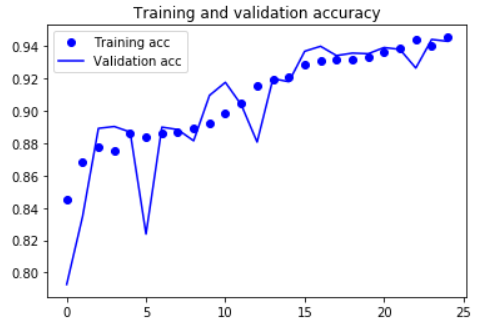
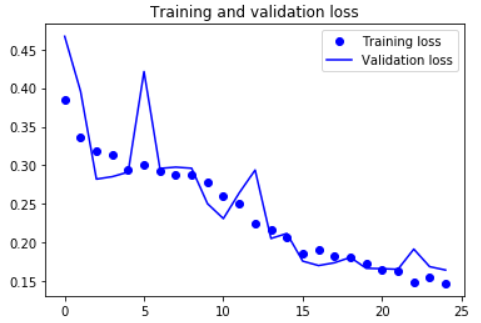
## First Model

### Description

The first CNN model contained a Convolutional layer with an input shape of 96x96x3 (image dimensions, with 3 channels – RGB), 3 hidden layers and a Fully Connected Dense layer with a sigmoid function activation for the binary classification. The model trained for 25 epochs, having the binary\_crossentropy as the loss function without the definition of callbacks.

### Results

As results, we can clearly conclude that the model is converging in every epoch: the loss decreases (from 0.39 until 0.15 points) and the accuracy increases (from 0.84 to 0.94 points) in every epoch. More detailed information is presented in the plots below. The accuracy results of the model are pretty good but it needs some tuning to perform better results. In the following chapter the evaluation steps of the model will be described in more detail.

## First Model Enhancement

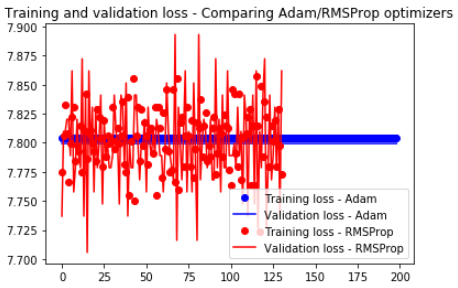
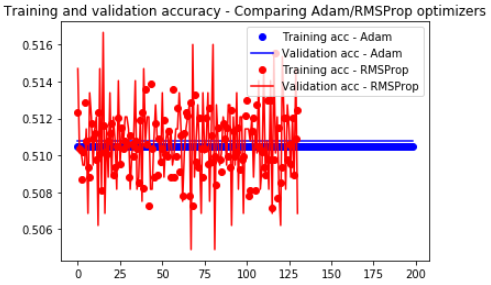
### Description

As a second step, to enhance our model, we trained the first model using two different optimizers, Keras and customized callbacks, and set the number of epochs to 1000. As optimizers, we used both Adam and RMSProp. As callbacks, we used:

* Keras callbacks:
  + EarlyStopping – enables training early stopping;
  + ReduceLROnPlateau – refines Learning Rate when finding a Loss Plateau;
  + ModelCheckpoint – saves the best model;
  + TerminateOnNaN – terminates the model training when unexpected values are obtained.
* Customized callbacks:
  + MakeLRGreatAgain – increases Learning Rate when getting no improvements;
  + BetterCSVLogger – same as Keras’ CSVLogger, adding a date log.

### Results

The results of the model enhancement are displayed in the following plots. Here, we can compare the performance of both Adam and RMSProp optimizers. The RMSProp optimizer is showing a better accuracy and loss result than Adam, as Adam stabilized earlier.



## Model with Pre-Trained Neural Network

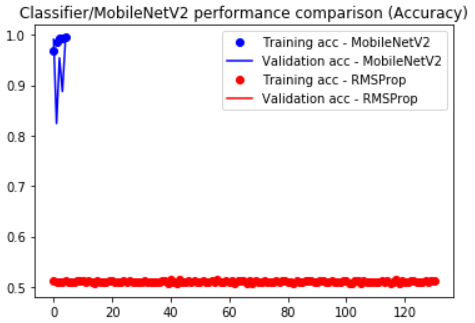
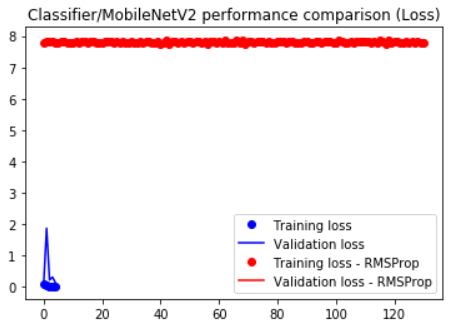
### Description

As a final solution, we used a pre-trained feature extraction CNN named MobileNetV2. This architecture is a lightweight but well-performing network, and it was developed by Google Research. We picked MobileNetV2 because it can run almost in any computer and it trains faster than the most common ones (like VGG-16 and EfficientNets). Plus, we defined our last solution as a trained MobileNetV2 in the ImageNet dataset), without using its fully-connected layers that classify 1000 classes. Instead, we defined our own fully-connected Dense layer for a binary classification, activated by a sigmoid function. Also, based on the previous comparison, we chose the RMSProp optimizer together with the callbacks, due to a better performance.

### Results

We plotted the results of the MobileNetV2 against the previously trained model with the RMSProp optimizer and our callback list.

Increasing the number of epochs brought the model to a clear overfitting to both training and validation datasets (the accuracy in both is equal to 100% and the loss obtained is 0), after 284 epochs. These results are displayed below.

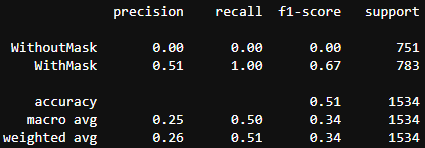
## Final Results

The final results were calculated with the test dataset, a dataset that contains images the model has never seen before.

* Models with Adam and RMSProp optimizers:

Accuracy: 50.91%

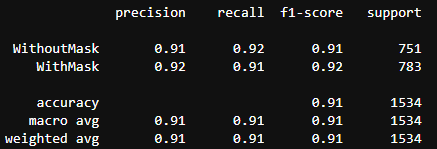
Classification Report:



* Model containing the pre-trained MobileNetV2:

Accuracy: 91.46%

Classification Report:



## Model Generalization

Finally, we’d like to check the performance of our best model when dealing with other dataset, from other source, with new data which our model has never seen. This will be the final test of the model generalization ability.

# Conclusions

In the conclusion, we’d like to state that the problem of the classification whether a person is wearing or not a medical mask can be solved by our best model, with the accuracy of 91%. We consider the performance of our model to be good, making us recognize that using pre-trained models and Transfer Learning is an important approach concerning Deep Learning problems.