Image and Speech Recognition (EIASR)

Final Report

Face mask detection

Mateusz Hryciów, 283365

1. Task description

   The goal of the project is to create the face mask detection system. Based on the photograph, it will be able to determine whether a given person is wearing a face mask or not. For that we have decided to create and train Convolutional Neural Network (CNN). In order to compare results we also used pre-trained ResNet18 network and applied transfer learning.

2. Data selection.

   We have managed to find a suitable data set for our problem. https://github.com/chandrikadeb7/Face-Mask-Detection/tree/master/dataset It contains 1916 images of people wearing a face mask and 1919 images of people without a face mask.

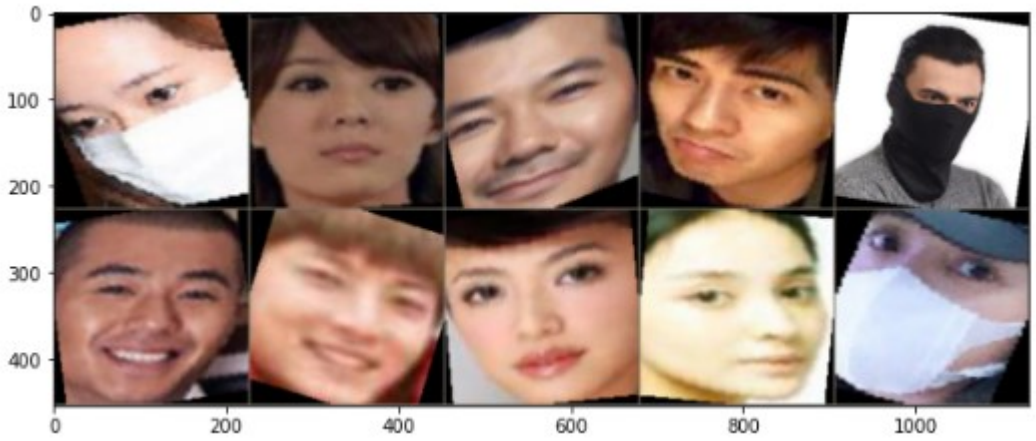
3. Data preprocessing.

   In order to properly train CNN models at first we decided to normalise our input images using the mean and standard deviation of ImageNet, which has been calculated based on millions of images. Additionally the data augmentation process has been performed, which included:

   - horizontal flip - with default probability p equal to 0.5, the images are flipped horizontally,
   - rotation - each image is being rotated by a randomly chosen number of degrees between -20 and 20,
   - normalization.

   Last step was to resize the images, so that they all match the same size of 224x224 pixels.

Sample images after augmentation:



4. CNN network

In order to detect whether the person has a facemask, we have created and trained the convolutional network, which was then compared to the existing, pretrained network.

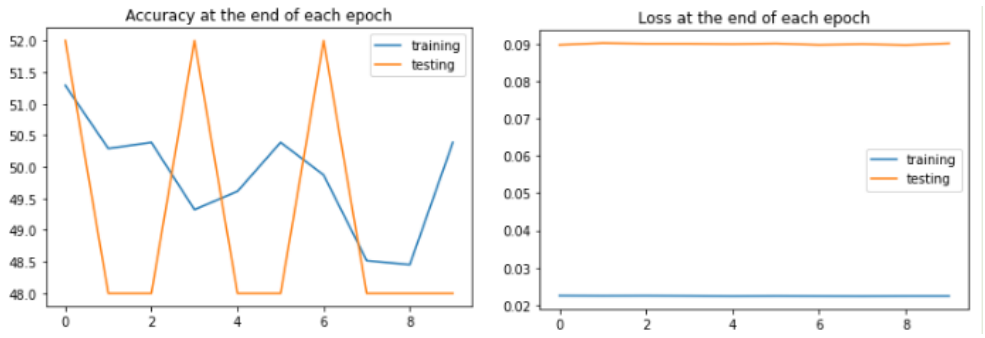
Architecture of the network:

- convolutional layer with filters of size 10x10x16, stride equal to 3 and padding equal to 1
- convolutional layer with filters of size 20x20x32, stride equal to 3 and padding equal to 1
- max pooling layer with 2x2 windows and stride equal to 2
- dropout layer with the probability equal to 0.5
- 3 fully connected layers with 2 output channels at the end of the network

As an activation function we have used the rectified linear unit function. In order to calculate the loss during draining the cross entropy function was used. It was followed by Adam optimization algorithm for updating the weights of the network during training. It is broadly used by the community, as it is a very effective optimizer.

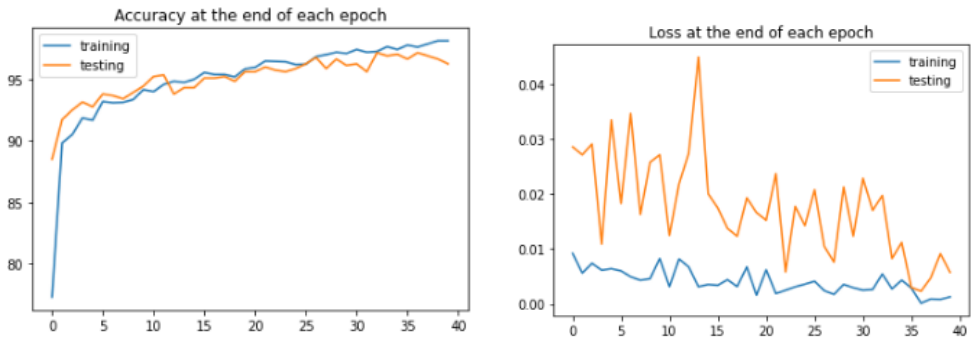
Training of the network - hyperparameters optimization

a) learning rate = 0.01; epochs no. = 10



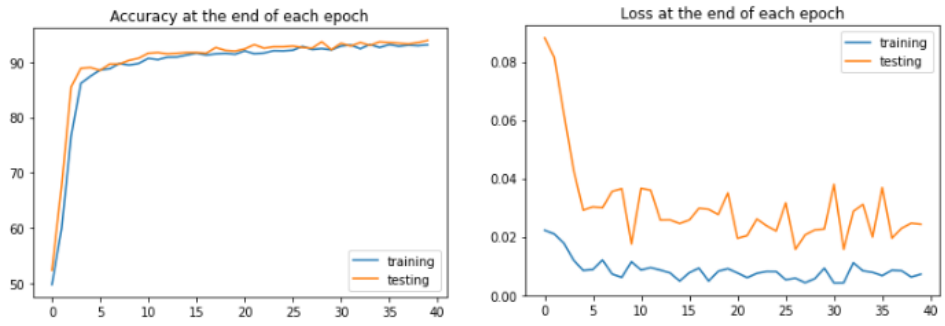
Learning rate equal to 0.01 turned out to be too large, which ended with the very poor model accuracy.

b) learning rate = 0.0001; epochs no. = 40



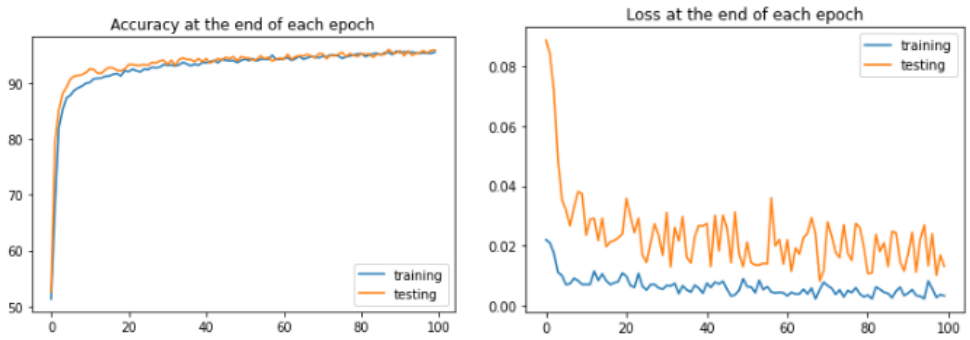
In next step we decided to change the learning rate to 0.0001 and increase number of epochs as well. Model had great accuracy at the level of 98% for training data and around 95% for testing images.

c) learning rate = 0.00001; epochs no. = 40



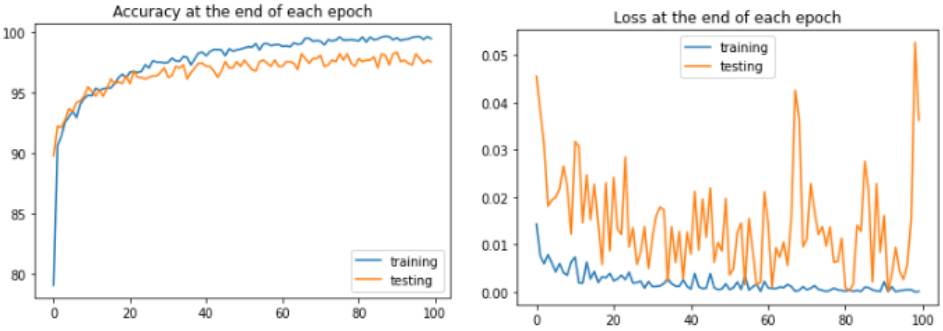
After changing the learning rate by one order of magnitude the results we received are slightly worse in terms of accuracy, but the network is still learning in the final part, so we decided to increase the number of epochs.

d) learning rate = 0.00001; epochs no. = 100



For an increased number of epochs the accuracy of the model increased by 2%. Having in mind that the highest accuracy was received for the bigger learning rate, we decided to test it with a higher number of epochs.

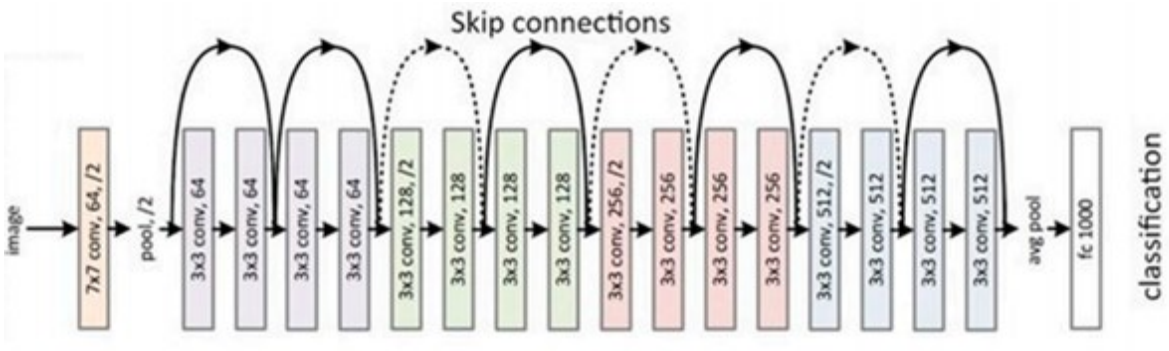
e) learning rate = 0.0001; epochs = 100



The highest accuracy was obtained by learning the network with the rate equal to 0.0001 in 100 epochs. However, as it will be presented in the performance evaluation part, the model is most probably overfitting the initial dataset, as its results on randomly chosen photos were a bit worse, then in the case of networks with lower accuracy.

5. Transfer Learning

Transfer learning was used to compare the results with the created CNN network. This is an approach that uses a pre-trained network. In this case, ResNet18 (Residual Network), which is successfully used for image recognition, was used. However, in order for it to perform its task some changes had to be made to the classifier layer. For this reason, fully connected layer with size 1000 was removed and fully connected layer with a number of outputs equal to 2, which is the number of classes, was inserted. As only transfer learning was performed, weights in all other layers were frozen.

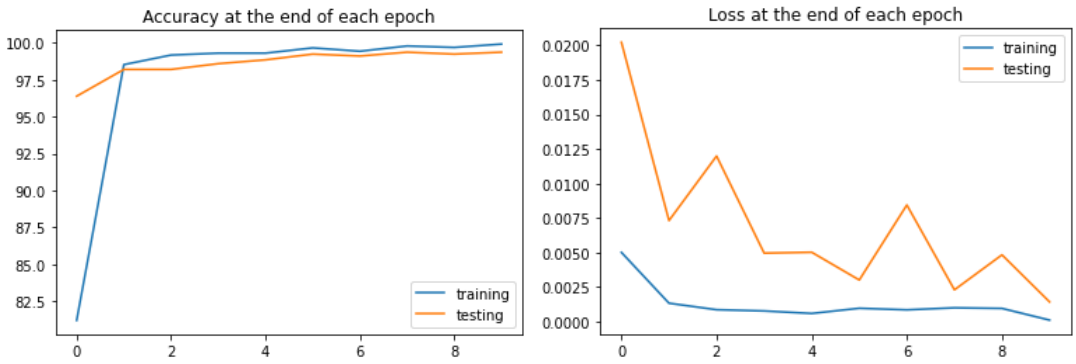


Original ResNet18 model (T. Rahman, M.E.H. Chowdury, "Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection Using Chest X-ray, 2020)

The following parameters were used for learning:

- batch size - 100
- criterion - cross entropy loss
- optimizer - stochastic gradient descent
- learning rate - 0.001
- momentum - 0.9
- number of epochs - 10

The precision and loss changes at the end of each epoch during transfer learning are shown below.

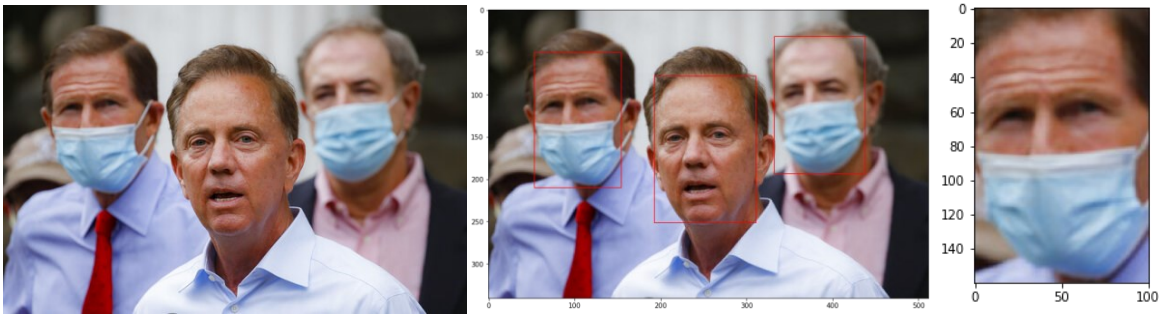


As we can see the network has learned to a satisfying degree. Additionally, minimally better results were obtained on the training set. Further analysis of the network will be done later in the paper.

6. Face extraction

As shown earlier only face images were used for training. So to be able to test the classifier on any photos it was necessary to use face extractor. Due to the fact that it was not the main goal of the project we used a ready-made solution (https://github.com/kb22/Create-Face-Data-from-Images). However, some modifications were made to enable marking frames around each face found.

The performance of face extractor is shown above. As expected, it selects and cuts all faces from the photo.

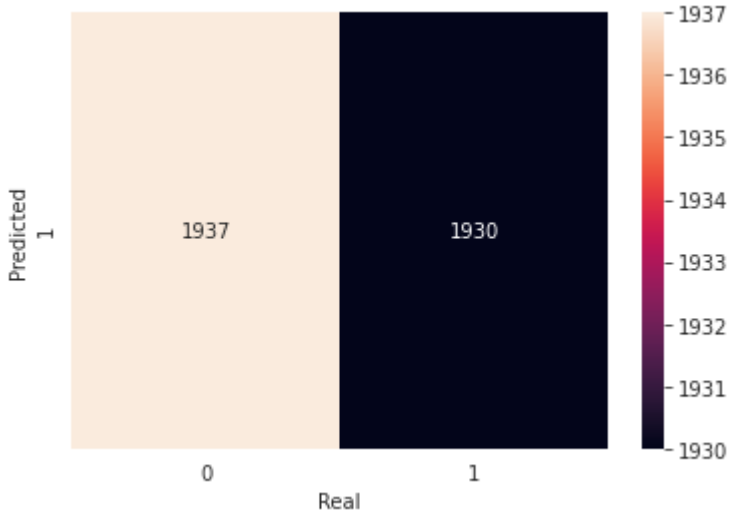


Face extractor is also working if people in the picture are wearing masks.

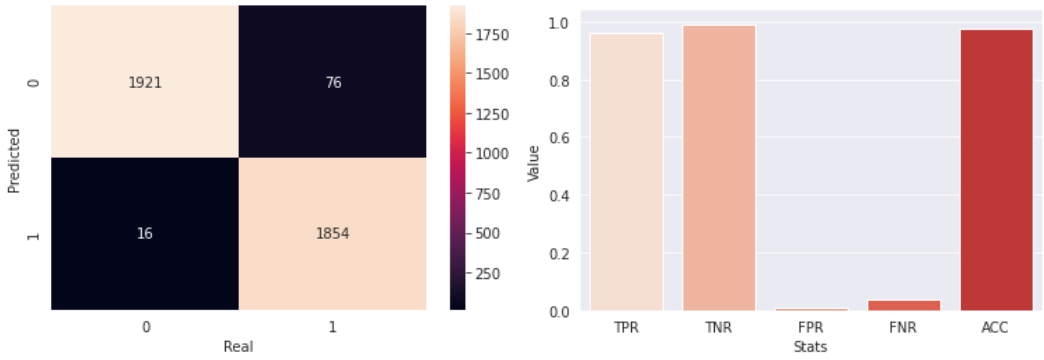
7. Performance evaluation

In order to measure the accuracy of our classification networks we have created a few confusion matrices and derived from them some additional information:

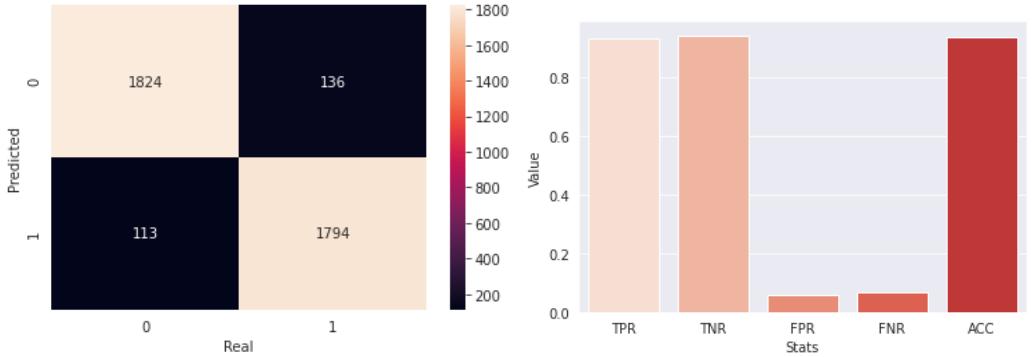
a) learning rate = 0.01; epochs no. = 10



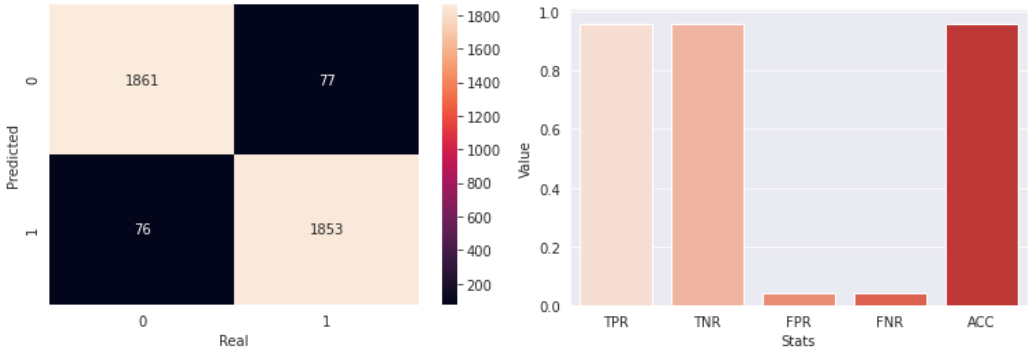
b) learning rate = 0.0001; epochs no. = 40



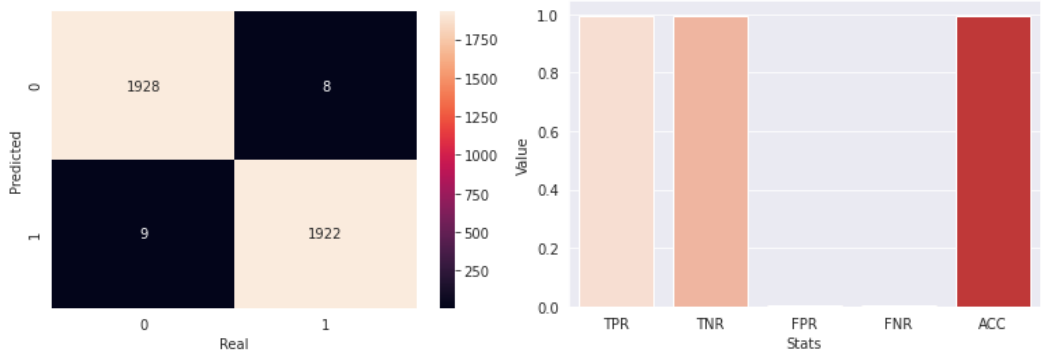
c) learning rate = 0.00001; epochs no. = 40



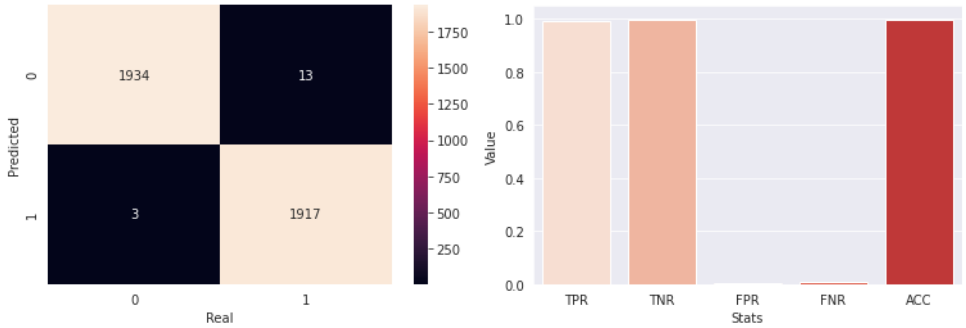
d) learning rate = 0.00001; epochs no. = 100



e) learning rate = 0.0001; epochs = 100



f) ResNet18



As we can see, apart from the first network, the accuracy overall is quite high, especially in network with learning rate = 0.0001 and number of epochs equal to 100 as well as for network created with the transfer learning method. First network with too high learning rate have classified every picture as without a mask, which is obviously a wrong prediction rate, but this particular network was rather created for comparison purposes.

8. Difficult cases and additional tests

Tests on the sample photos from outside of the original database used to train and evaluate the performance:

As we initially assumed, the model has problems with masks that are trying to mimic the real human face.