



# UNIVERSITÀ DI PISA

**Computer Engineering**

Intelligent Systems Project

**Project Documentation**

Gaia Anastasi

---

**Academic Year 2020/2021**

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>                                       | <b>3</b>  |
| <b>2</b> | <b>Design and develop of an intelligent system</b>        | <b>3</b>  |
| 2.1      | Data Balancing . . . . .                                  | 3         |
| 2.1.1    | Balancing of tha dataset . . . . .                        | 3         |
| 2.1.2    | Feature selection . . . . .                               | 5         |
| 2.1.3    | Normalization of data . . . . .                           | 6         |
| 2.2      | Multi-Layer Perceptron Networks . . . . .                 | 6         |
| 2.2.1    | Arousal MLP Network . . . . .                             | 6         |
| 2.2.2    | Valence MLP Network . . . . .                             | 9         |
| 2.3      | Radial Basis Function Network . . . . .                   | 11        |
| 2.3.1    | Arousal RBF network . . . . .                             | 11        |
| 2.3.2    | Valence RBF network . . . . .                             | 13        |
| 2.4      | Mamdani-type fuzzy inference system . . . . .             | 15        |
| <b>3</b> | <b>Classification of emotions using facial expression</b> | <b>20</b> |
| 3.0.1    | Image selection . . . . .                                 | 20        |
| 3.0.2    | Design Choices . . . . .                                  | 20        |
| 3.0.3    | Final Results . . . . .                                   | 21        |

# 1 Introduction

The project is composed by 2 parts. The aim of the first part is to design and develop an intelligent system which measures a person's affective state using signals recorded by sensors while the aim of the second part is to develop a classifier of emotions using face images as inputs.

The tasks performed in this project are:

- **3.1:** Design and development of two MLP networks and two RBF networks that accurately estimate a person's valence and arousal levels, respectively.
- **3.3:** Design and development of a fuzzy inference system to fix the deficiencies in the arousal dimension.
- **4.2:** Fine-tuning of a pre-trained CNN to accurately classify a person's emotions based on facial expressions.

## 2 Design and develop of an intelligent system

### 2.1 Data Balancing

Before implementing the first task of the project, a pre-processing of the dataset was needed. The steps done are:

- Removal of non-numerical or infinite values using **MATLAB** function *isinf* to verify the presence of such values
- Removal of outliers using **MATLAB** function *rmoutliers* and the default *median* method.
- Balacing of the dataset
- Features selection
- Normalization of data

The last 3 items will be discussed in the following sections.

#### 2.1.1 Balancing of tha dataset

The dataset is composed by 54 signals and for each of them a value for valence and one for arousal are presented. Considering that valence and arousal can take 7 different values, the dataset can be split in 7 classes. The distribution of samples among the classes is the following:

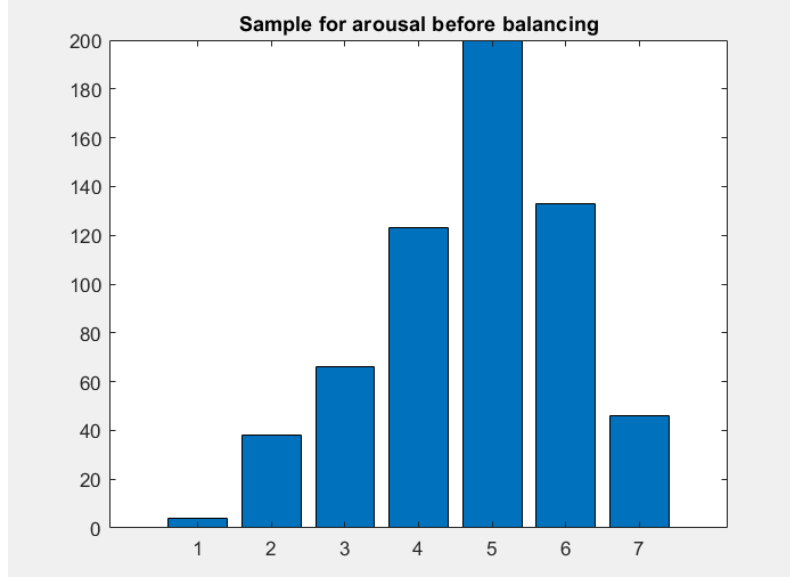


Figure 1: Samples distributions for arousal before balancing

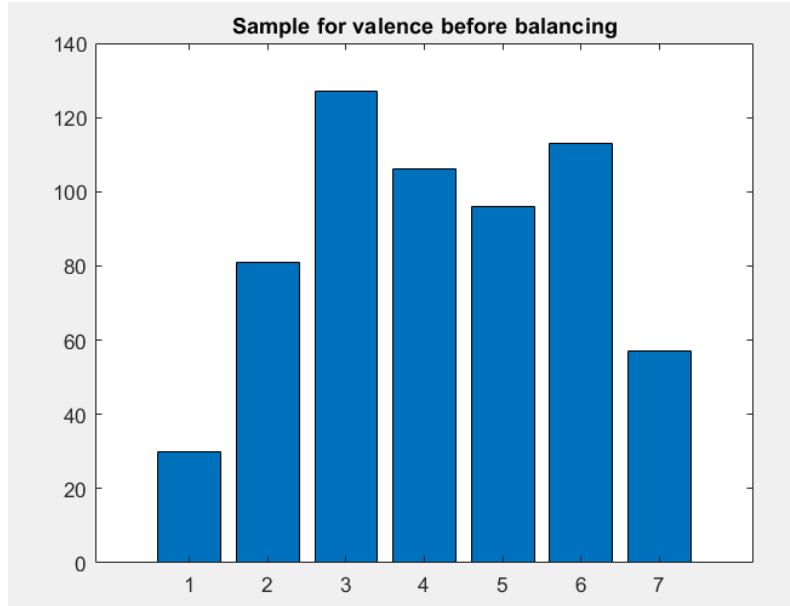


Figure 2: Samples distributions for valence before balancing

The samples are unbalanced so the following algorithm based on undersampling, oversampling and data augmentation was used to balance the dataset. Data augmentation is applied on samples which belong to the least common class for arousal(valence) and not to the most common class for valence(arousal) and then the samples which belong to the most common class for arousal(valence) and don't belong to the least common class for valence(arousal) are removed. These steps were repeated *rep* times.

After performing the balancing, these are the distributions of samples for both

the arousal and the valence:

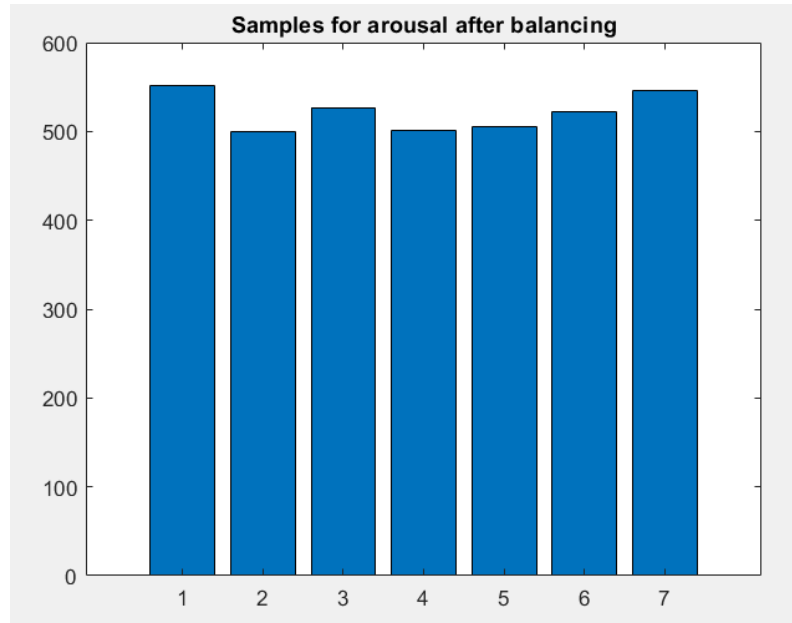


Figure 3: Samples distributions for arousal after balancing

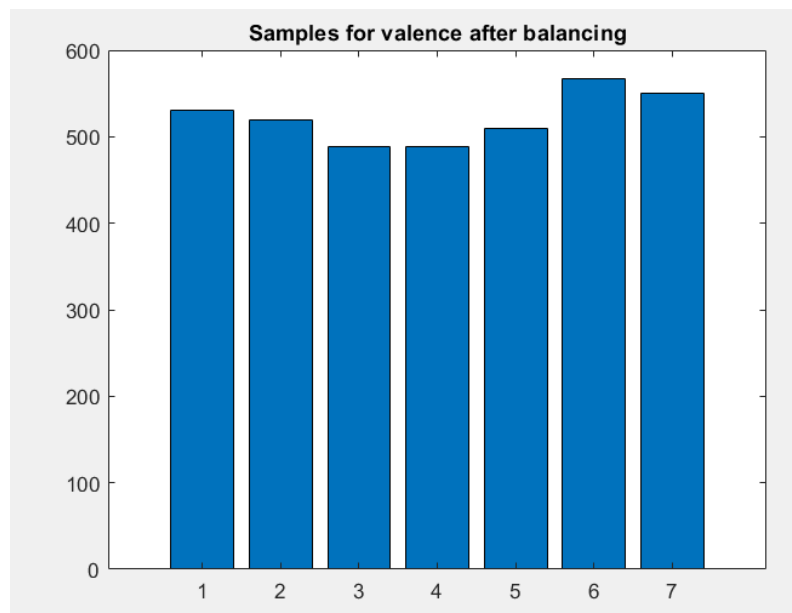


Figure 4: Samples distributions for valence after balancing

### 2.1.2 Feature selection

To select the features **MATLAB** function *sequentialfs* was used. The function should be repeated a statistically relevant number of times to extract the right features, but in order to perform the algorithm in an acceptable amount of time

in this project the *sequentialfs* function was repeated just 5 times. At the end of the algorithm 5 **.mat** files were generated:

- *training-arousal.mat*: which contains the training inputs and corresponding target outputs for arousal training
- *testing-arousal.mat* which contains the testing inputs and corresponding target outputs for arousal testing
- *training-valence.mat* which contains the training inputs and corresponding target outputs for valence training
- *testing-valence.mat* which contains the testing inputs and corresponding target outputs for valence testing
- *best3.mat* which is a structure containing the training data and testing data for arousal of the best 3 features selected. This dataset will be used for the task 3.3

### 2.1.3 Normalization of data

The data were normalized using the **MATLAB** function *normalize* which takes as an argument the data that need to be normalized. For all the other parameters the default values were used.

## 2.2 Multi-Layer Perceptron Networks

In this chapter the design and development of 2 MLP networks to accurately estimate a person's valence and arousal levels, respectively, will be discussed. The experiments reported here were initially performed on data not normalized, but the same experiments were performed as well on normalized data. The results of the latter ones were not reported here because no significant changes were found between the two cases. Some experiments were performed in order to determine which MLP architecture is the best for the project purpose.

### 2.2.1 Arousal MLP Network

Network configuration:

```
hiddenLayerSize_arousal = 30;  
mlp_arousal = fitnet(hiddenLayerSize_arousal);  
mlp_arousal.divideParam.trainRatio = 0.7;  
mlp_arousal.divideParam.testRatio = 0.1;  
mlp_arousal.divideParam.valRatio = 0.2;  
mlp_arousal.trainParam.epochs = 110;
```

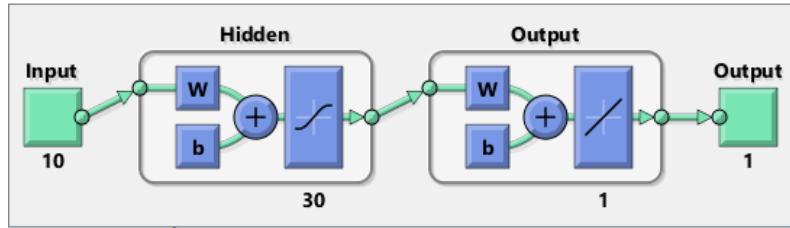


Figure 5: Arousal MLP network architecture

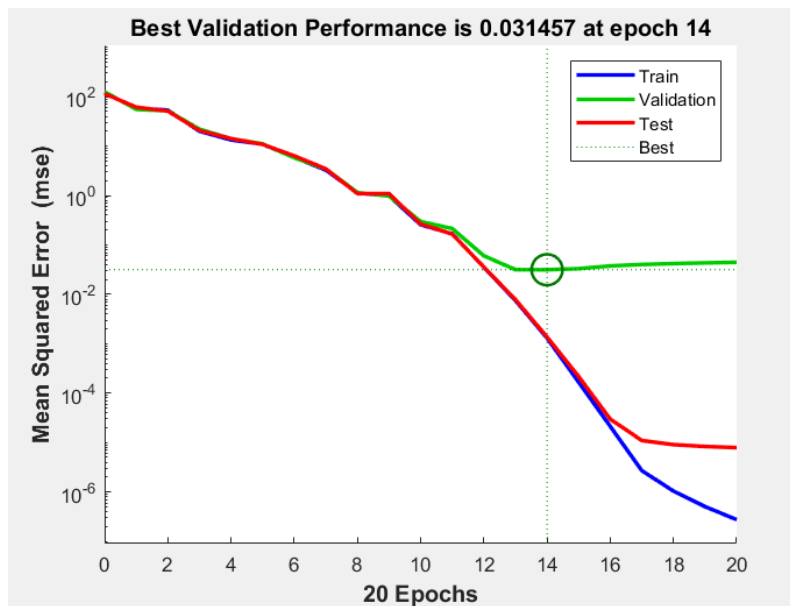


Figure 6: Performance for arousal MLP network

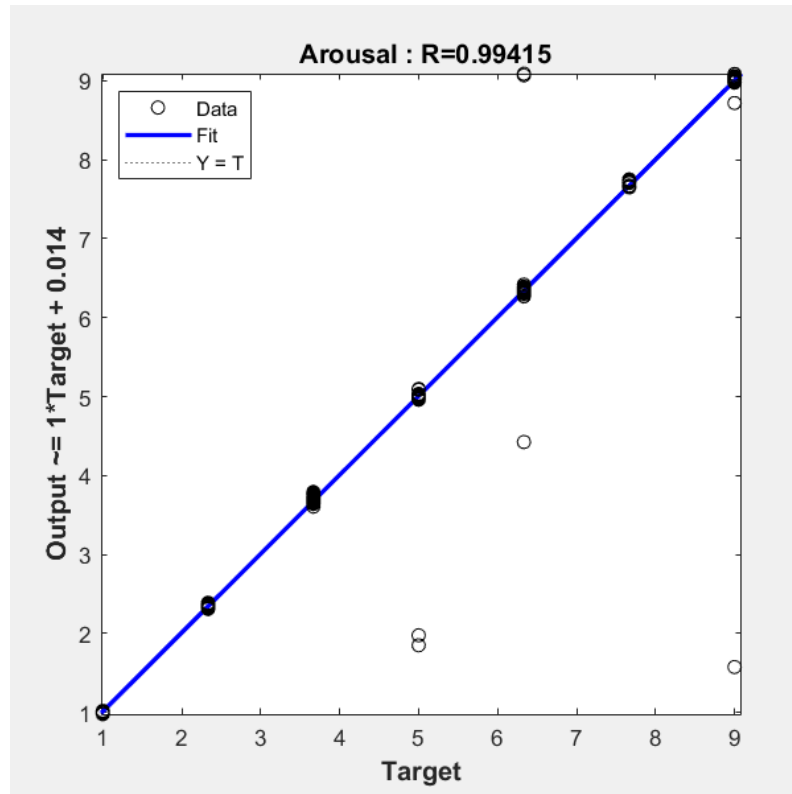


Figure 7: Regression for arousal MLP network



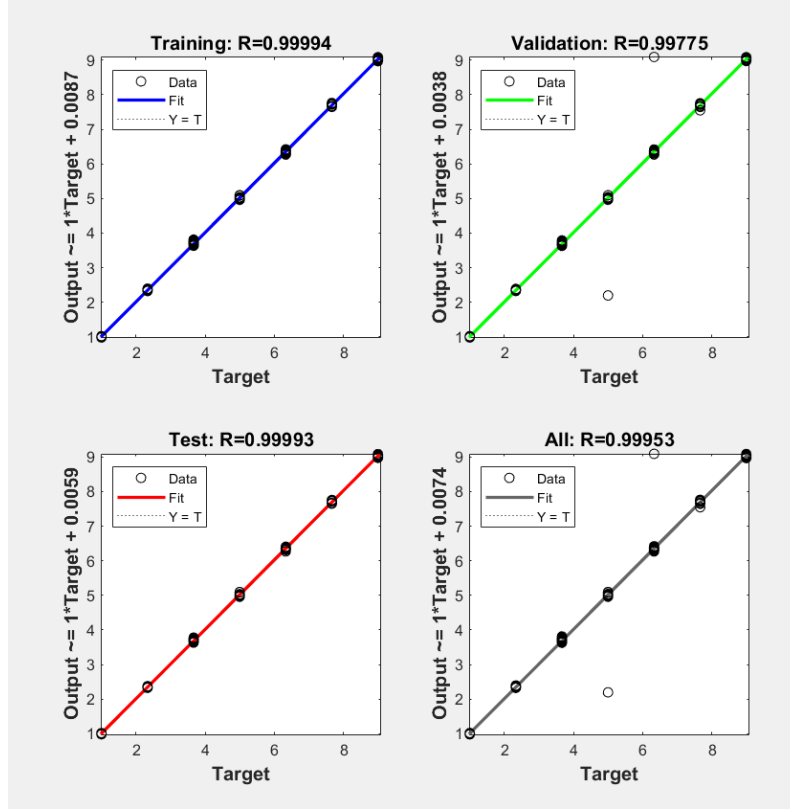


Figure 8: Regression results after training for arousal MLP network

### 2.2.2 Valence MLP Network

Network configuration:

```
hiddenLayerSize_valence = 35;
mlp_valence = fitnet(hiddenLayerSize_valence);
mlp_valence.divideParam.trainRatio = 0.7;
mlp_valence.divideParam.testRatio = 0.1;
mlp_valence.divideParam.valRatio = 0.2;
mlp_valence.trainParam.epochs = 110;
```

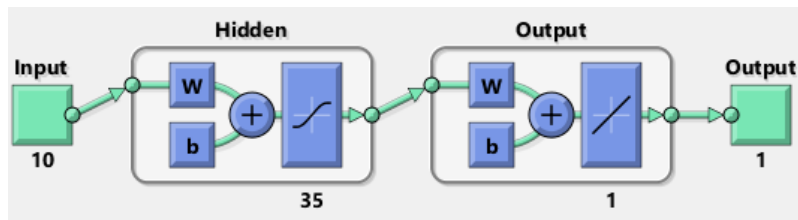


Figure 9: Valence MLP network architecture

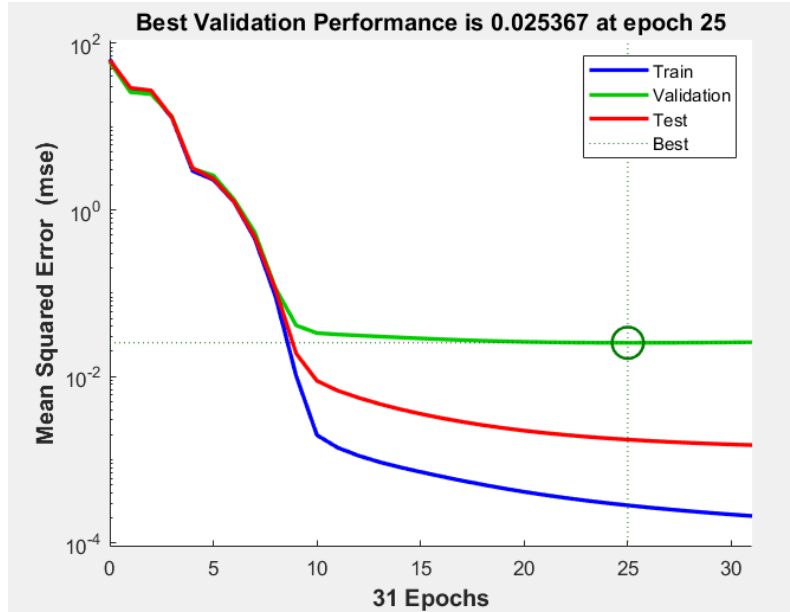


Figure 10: Performance for valence MLP network

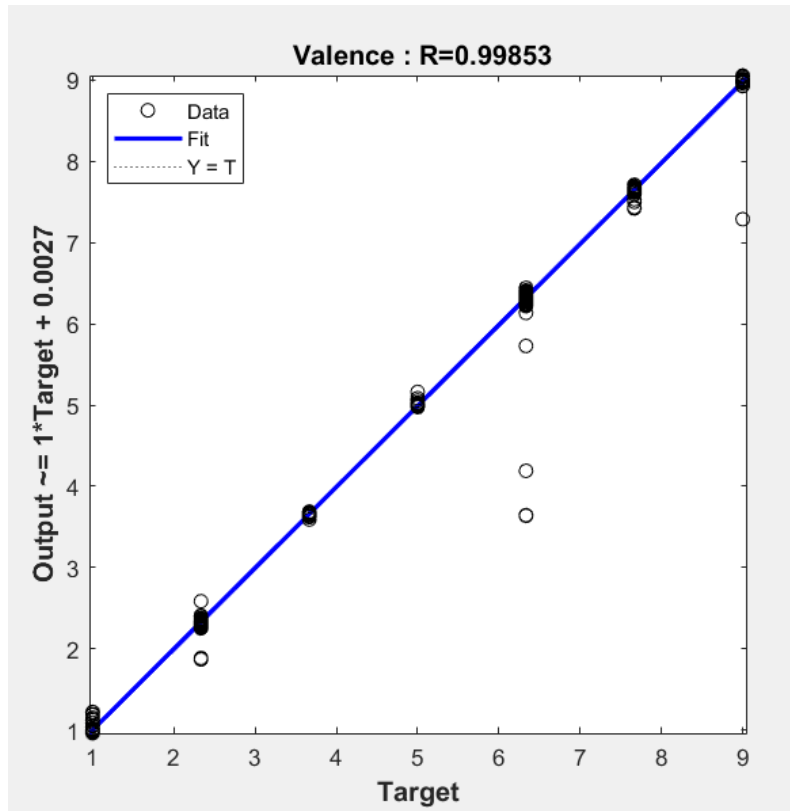


Figure 11: Regression for valence MLP network

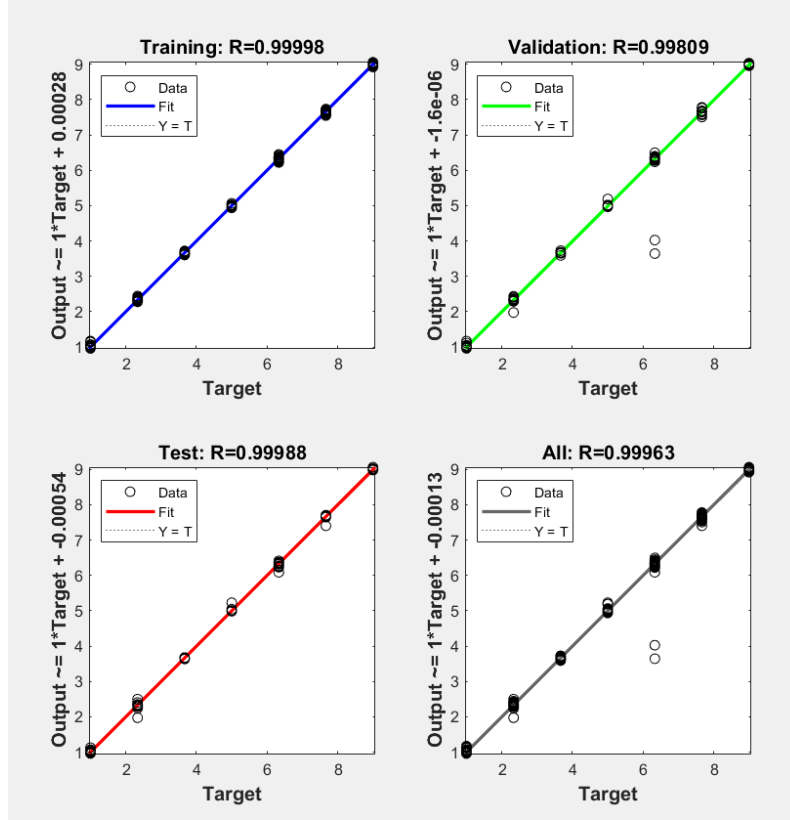


Figure 12: Regression results after training for valence MLP network

In general, valence MLP network requires more hidden neurons to perform as good as the arousal MLP network.

## 2.3 Radial Basis Function Network

In this chapter the design and development of 2 RBF networks will be discussed. Again the experiments were performed on both normalized and not normalized data, but just the results about not normalized ones were reported here. Some experiments were performed in order to determine which RBF architecture is the best for the project purpose.

### 2.3.1 Arousal RBF network

Network configuration:

```
goal_ar = 0;
spread_ar = 1;
K_ar = 400;
```

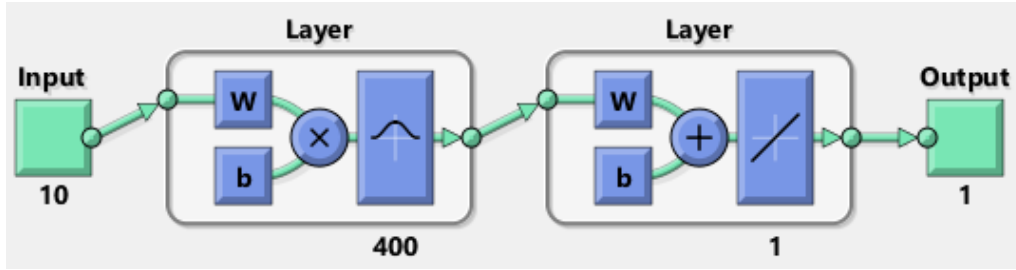


Figure 13: Arousal RBF network architecture

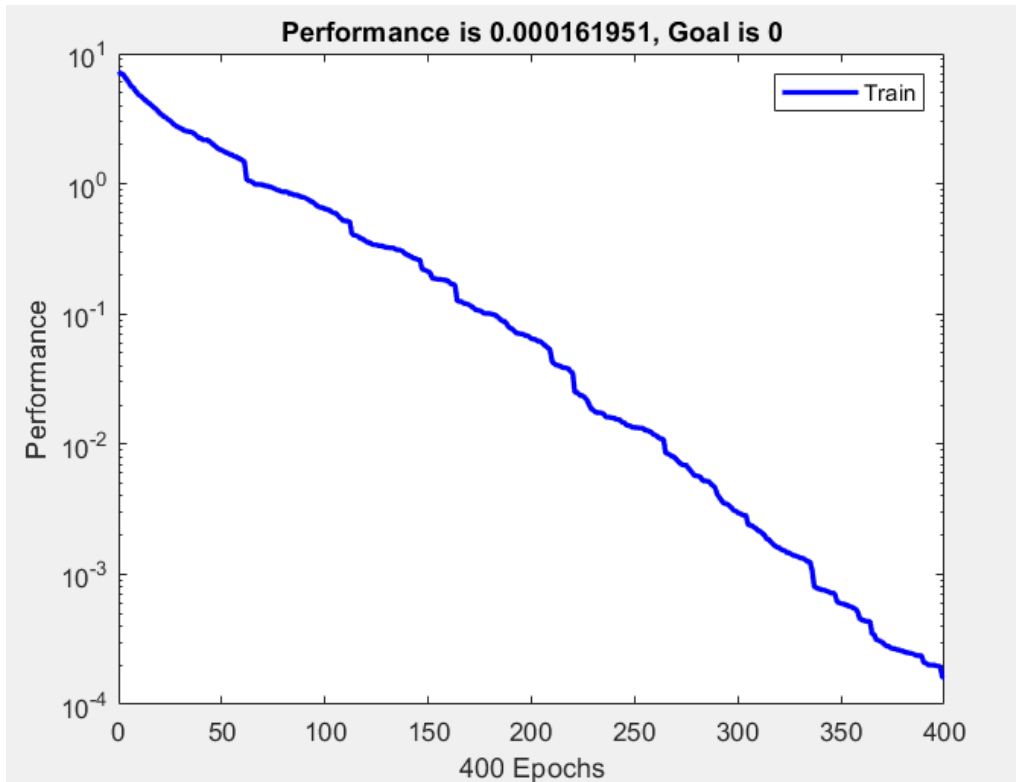


Figure 14: View arousal RBF network

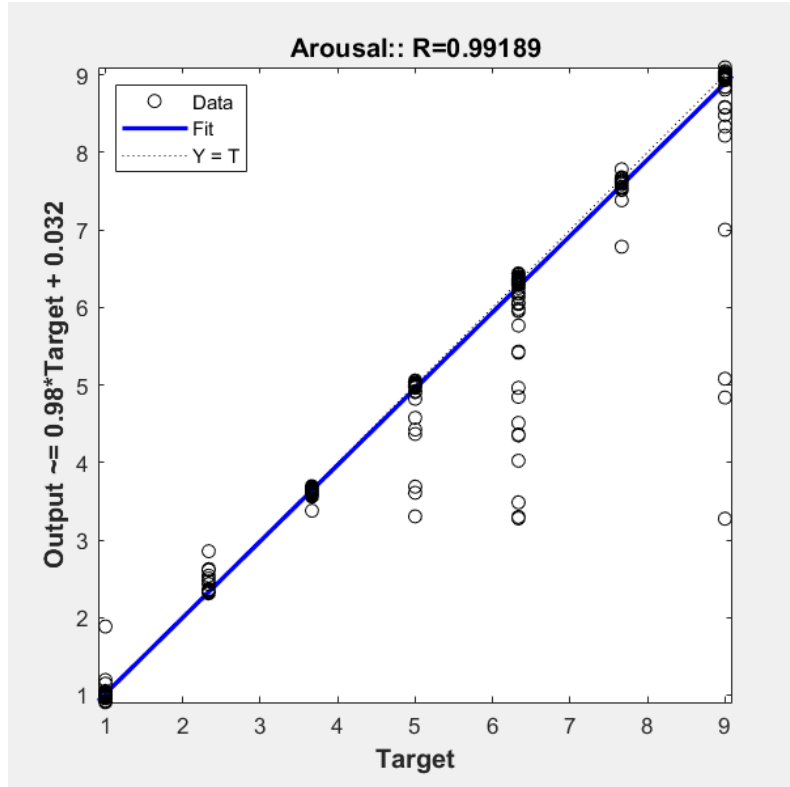


Figure 15: Regression for arousal RBF network

### 2.3.2 Valence RBF network

Network configuration:

```
goal_ar = 0;
spread_ar = 1;
K_ar = 350;
```

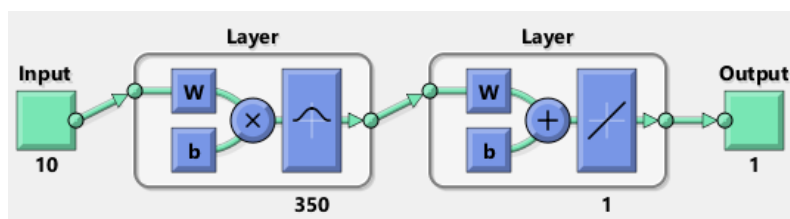


Figure 16: Valence RBF network architecture

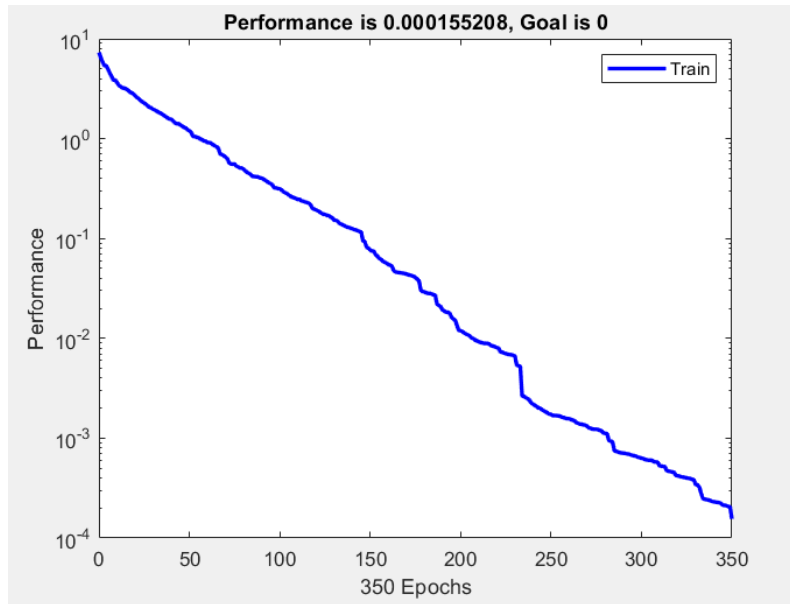


Figure 17: View valence RBF network

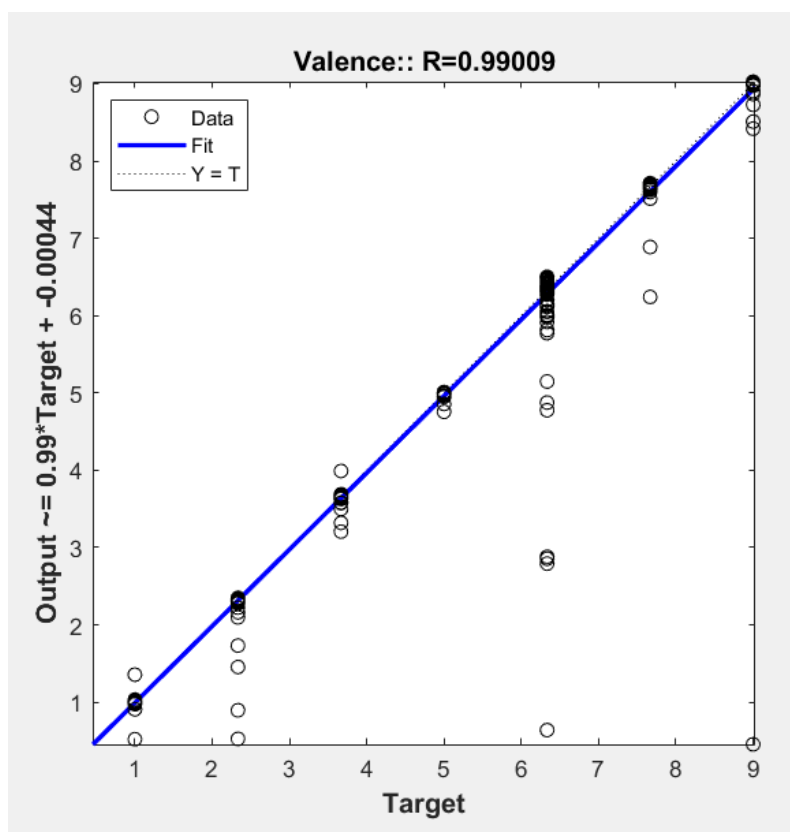


Figure 18: Regression for valence RBF network

In general, arousal RBF network requires more epochs to perform as good as the valence one.

## 2.4 Mamdani-type fuzzy inference system

The aim of this section is to show how a fuzzy inference system which fixes deficiencies in the arousal dimension was designed and developed. At the beginning the **empirical distributions** of the features through some histograms were done.

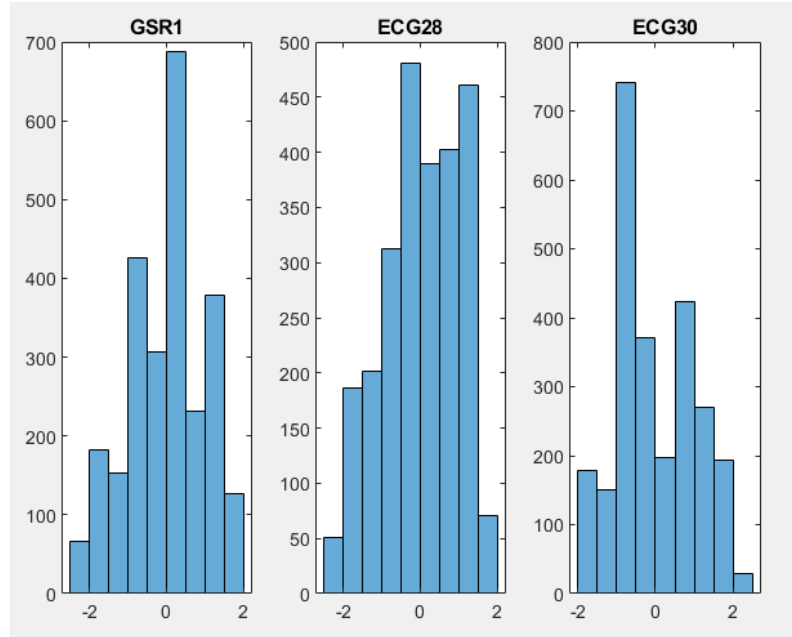


Figure 19: Empirical distribution of the features

These distributions have been used to model the linguistic variables of the three features.

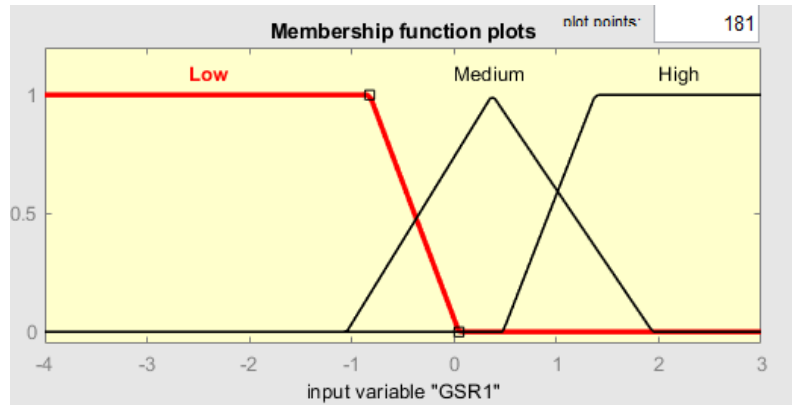


Figure 20: Modeling for linguistic variable of feature 27: GSR1

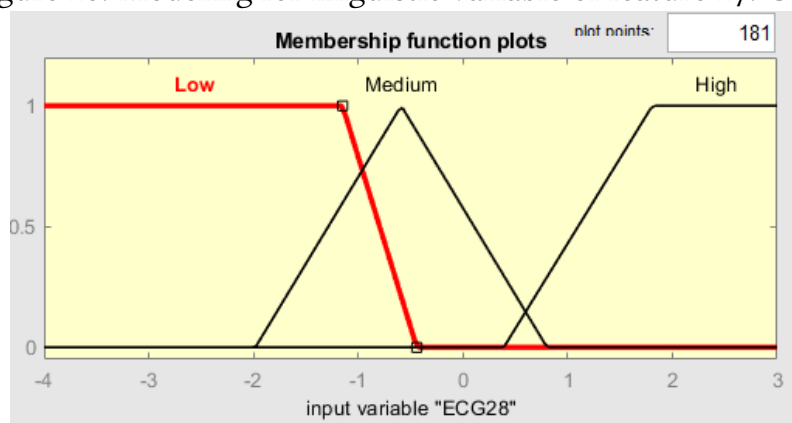


Figure 21: Modeling for linguistic variable of feature 11: ECG28

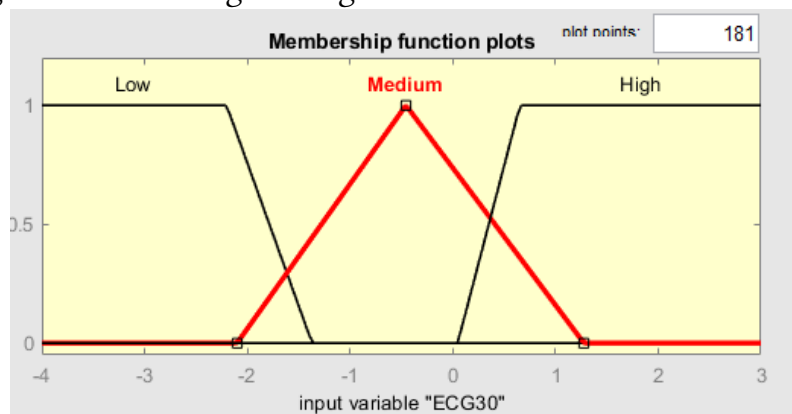


Figure 22: Modeling for linguistic variable of feature 13: ECG30

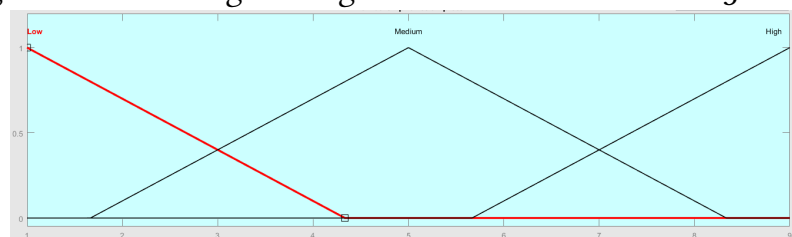


Figure 23: Modeling for linguistic variable of the output



For what concern the rules, other analysis have been performed. In particular a **correlation analysis** has been performed using different combinations of the input features to check possible correlations among the selected features. As we can see from the graphs below there is not correlation among the features. Plus, other histograms have been plotted considering this time just a subset of the samples coinciding to a specific range of arousal. As we have seen during the pre-processing the possible values of arousal can be divided in 7 classes, so the first and the second class were used to implement the low range of arousal, third, fourth and fifth classes to implement the medium range of arousal and final sixth and seventh to implement the high range.

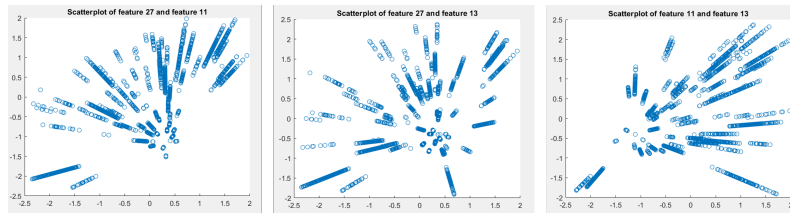


Figure 24: Correlation study

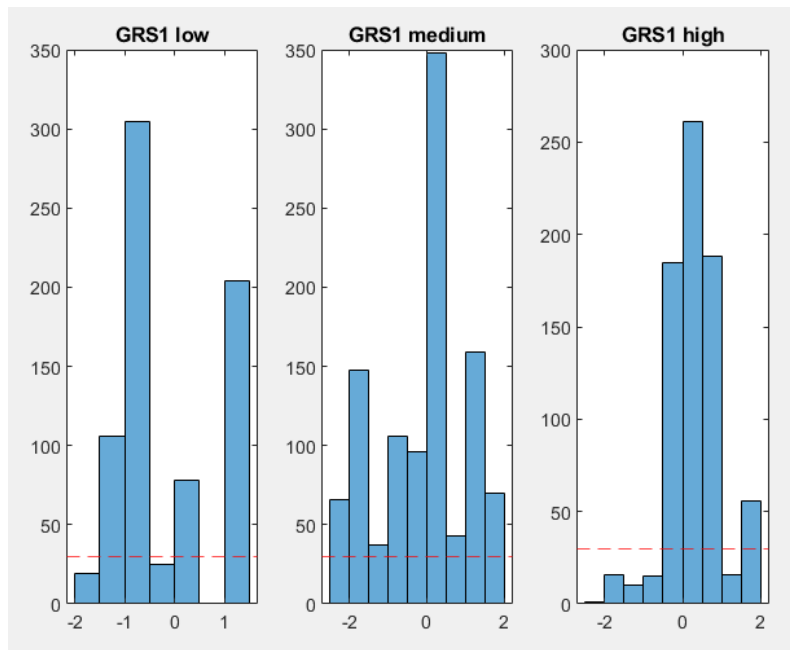


Figure 25: Histogram of feature GRS1(27) with different range of arousal

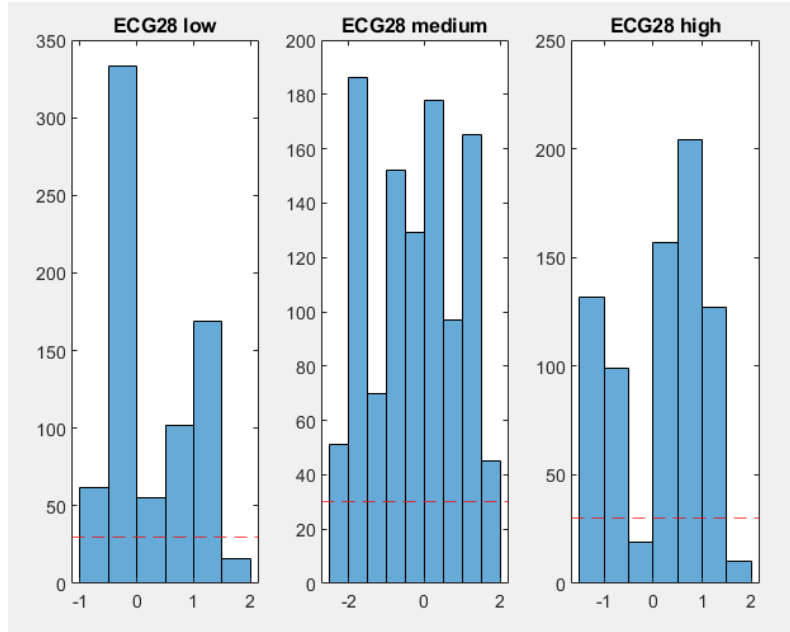


Figure 26: Histogram of feature ECG28(11) with different range of arousal

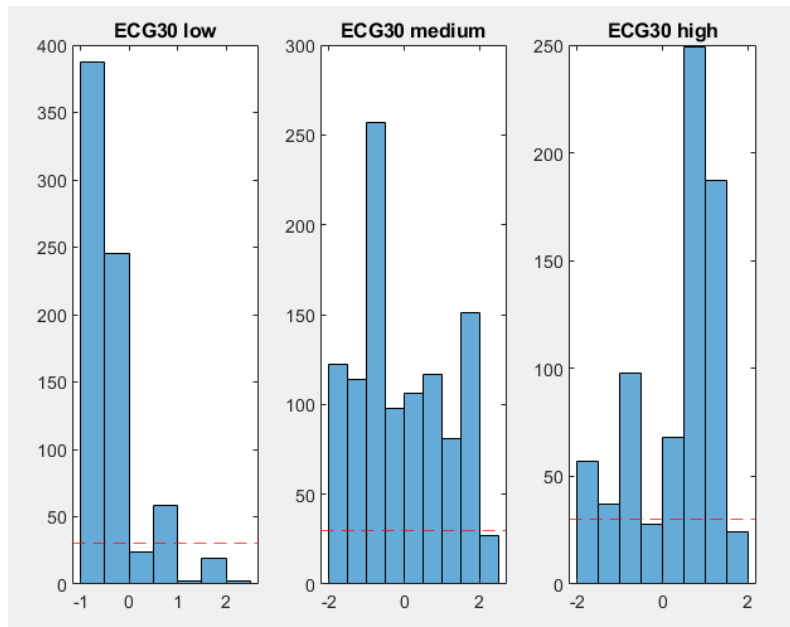


Figure 27: Histogram of feature ECG30(13) with different range of arousal

The rules modelled are the following:

1. If (GSR1 is Low) then (Arousal is Low) (1)
2. If (GSR1 is Medium) then (Arousal is Medium) (1)
3. If (GSR1 is High) then (Arousal is High) (1)
4. If (ECG28 is Low) then (Arousal is Medium) (1)
5. If (ECG28 is Medium) then (Arousal is Low) (1)
6. If (ECG28 is High) then (Arousal is High) (1)
7. If (ECG30 is Low) then (Arousal is Medium) (1)
8. If (ECG30 is Medium) then (Arousal is Low) (1)
9. If (ECG30 is High) then (Arousal is High) (1)

Figure 28: Fuzzy rules

### 3 Classification of emotions using facial expression

In the following chapter the development and the training of a CNN based on the pre-trained AlexNet will be discussed. The network will be used to classify facial expressions. The possible classes are:

- Anger
- Disgust
- Fear
- Happiness

#### 3.0.1 Image selection

Since the dataset provided is unbalanced, an image selection was performed. For each class 500 images were selected randomly at the beginning while in the second experiment the number of images for each class was brought to 1000. Finally, a new experiment was performed using 500 images for each class and each image was accurately selected among the ones provided. For both experiments 70% of these images will be used for training, 20% for validation and 10% for testing.

#### 3.0.2 Design Choices

All the images used were resized in order to fit the input size required by AlexNet. Moreover, a random translation of 30 pixels in horizontally and vertically way was performed as well on the training set in order to improve performance and prevent overfitting.

```
pixelRange = [-30 30];
imageAugmenter = imageDataAugmenter( ...
    'RandXReflection',true , ...
    'RandXTranslation',pixelRange , ...
    'RandYTranslation',pixelRange );

augmented_image_data_train = augmentedImageDatastore(
input_size(1:2), data_train ,
'DataAugmentation', imageAugmenter );
augmented_image_data_validation = augmentedImageDatastore(
input_size(1:2), data_validation );
augmented_image_data_test = augmentedImageDatastore(
input_size(1:2), data_test );
```

The last 3 layers of AlexNet were substituted with other layers. This was made because AlexNet is trained to classify thousands of categories, but in this project there is just 4 classes in which data need to be classified.

```
layers = [  
    original_layers  
    fullyConnectedLayer(numClasses, 'WeightLearnRateFactor', 20,  
        'BiasLearnRateFactor', 20)  
    softmaxLayer  
    classificationLayer];
```

The following hyper-parameters were used to train the network:

```
options = trainingOptions('sgdm', ...  
    'MiniBatchSize', 10, ...  
    'MaxEpochs', 10, ...  
    'InitialLearnRate', 1e-4, ...  
    'Shuffle', 'every-epoch', ...  
    'ValidationData', augmented_image_data_validation, ...  
    'ValidationFrequency', 3, ...  
    'Verbose', false, ...  
    'Plots', 'training-progress');
```

### 3.0.3 Final Results

In this section the final results about the CNN will be reported. In the case of 500 images randomly chosen for each class, the network has to perform less iteration before ending, but as clearly shown in the confusion matrix, performance aren't that great. In particular the network has some difficulty in recognising anger facial expressions and distinguishing them from disgust ones. The only facial expression that is detected with acceptable percentage is actually happiness that is the most different from the other ones and for this reasons also the most easily recognisable.

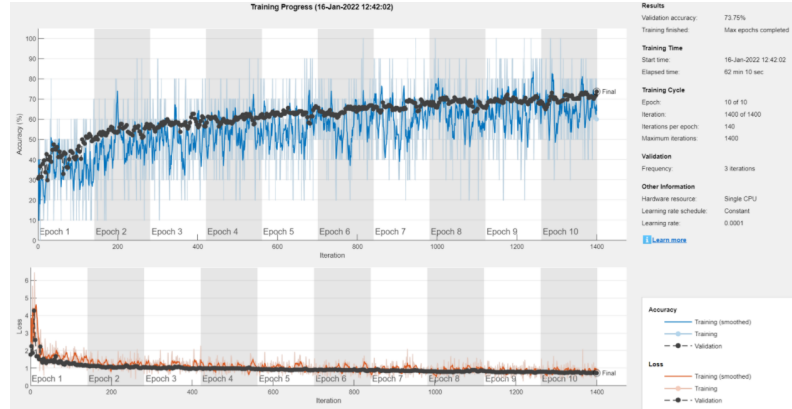


Figure 29: Training process of a CNN with 4 classes and 500 images for each class as inputs

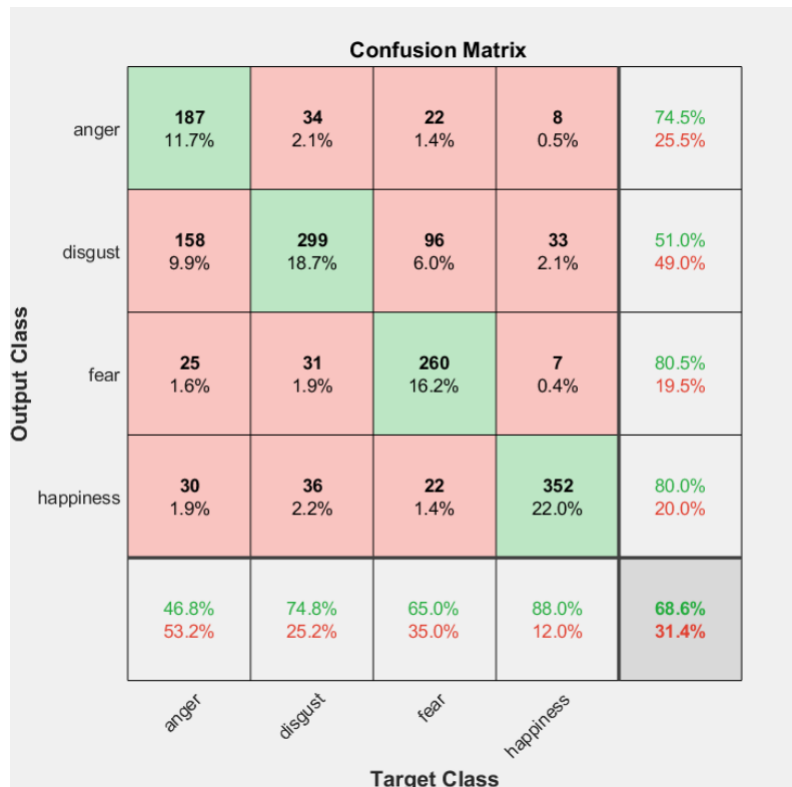


Figure 30: Confusion matrix of a CNN with 4 classes and 500 images for each class as inputs

Increasing the number of images used for each class helps to notice that performance improve. This is due to the fact that there are more data to train our network on. As shown in the following pictures, especially in the confusion matrix, the TPR in detecting correctly anger and fear have increased. Happiness remains the easiest recognisable facial expression among the considered ones.

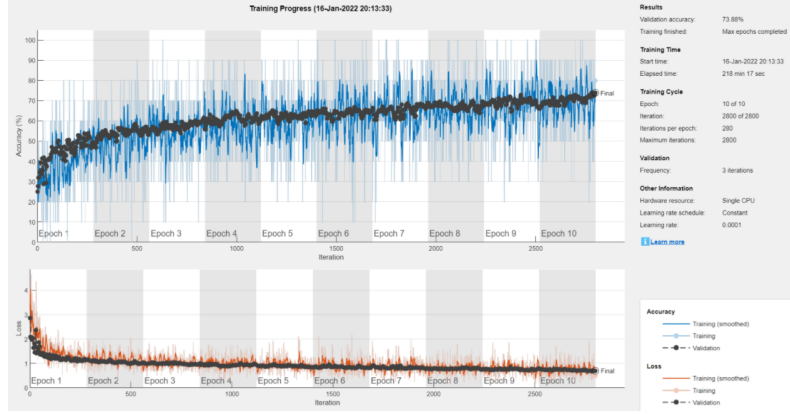


Figure 31: Training process of a CNN with 4 classes and 1000 images for each class as inputs

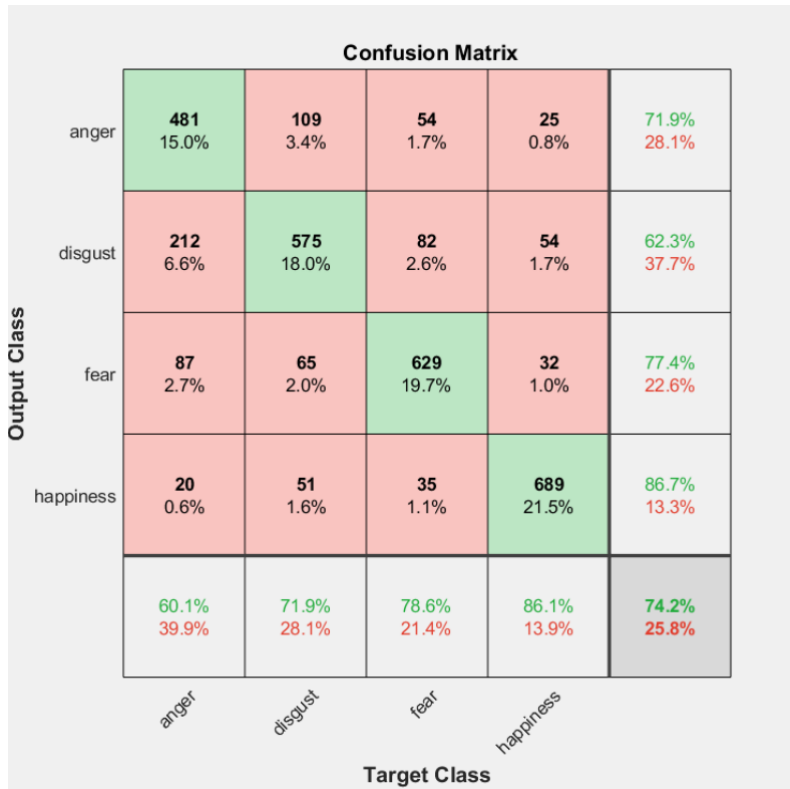


Figure 32: Confusion matrix of a CNN with 4 classes and 1000 images for each class as inputs

The last experiment was performed choosing accurately 500 images as inputs for each class. During the selection the images which have ambiguous facial expressions were removed. Comparing this experiment with the one where 500 images randomly were chosen it is clear that the validation accuracy increases from 73.75% to 87.21% and TPR increases as well while FPR decreases. Performance increases as well if comparing these results with the results obtain in the

case of 1000 images picked randomly.

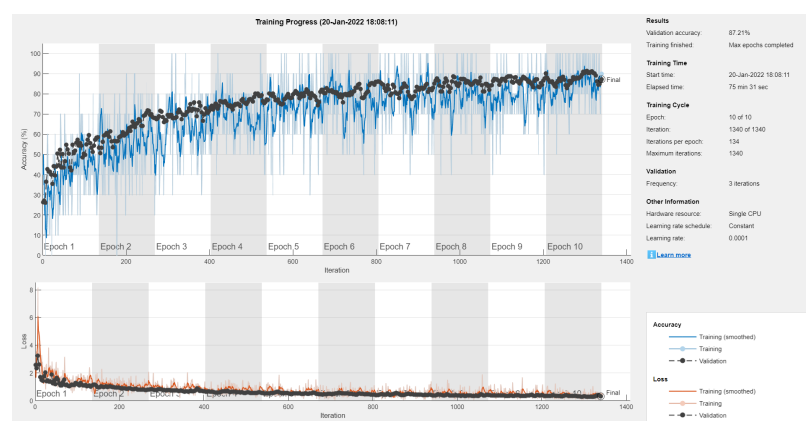


Figure 33: Training process of a CNN with 4 classes and 500 images accurately selected for each class as inputs

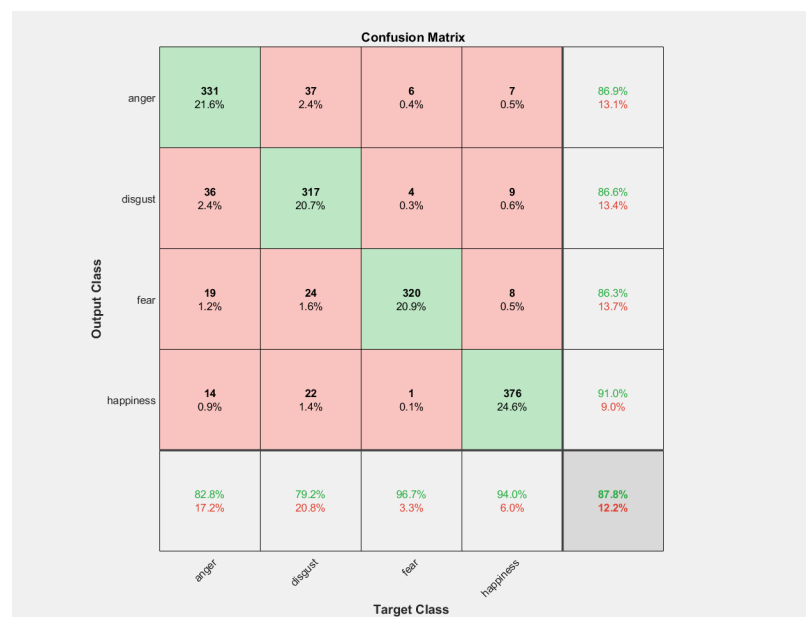


Figure 34: Confusion matrix of a CNN with 4 classes and 500 images accurately selected for each class as inputs

Performance improved a lot especially comparing the case in which 500 images were randomly chosen for each class and the case in which 500 images for each class were selected accurately . This is due to the fact that in the latter case the images provided are less ambiguous and this leads the network to make fewer mistakes.