Table of Contents

[**Data preparation** 4](#_Toc41831858)

[**Data pre-processing** 4](#_Toc41831859)

[**For Machine learning classifier** 5](#_Toc41831860)

[**For Deep learning classifier** 5](#_Toc41831861)

[**Training** 5](#_Toc41831862)

[**Machine learning classifier** 6](#_Toc41831863)

[**Deep learning classifier** 6](#_Toc41831864)

[**Evaluation** 6](#_Toc41831865)

[**For Machine learning classifier** 6](#_Toc41831866)

[**Accuracy** 6](#_Toc41831867)

[**F1 Score** 7](#_Toc41831868)

[**For Deep learning classifier** 8](#_Toc41831869)

[**Accuracy** 8](#_Toc41831870)

[**Training Results** 9](#_Toc41831871)

[**Area Under ROC Curve** 11](#_Toc41831872)

[**Training Results** 12](#_Toc41831873)

[**Conclusion** 13](#_Toc41831874)

List of Figures

[Figure 1. The list of directories where the data is stored. 3](#_Toc41822208)

[Figure 2. Training results on the FEI data set. 8](#_Toc41822209)

[Figure 3. Training results on the JAFFE data set. 9](#_Toc41822210)

[Figure 4. Training results on the mixed (FEI + JAFFE) data set 9](#_Toc41822211)

[Figure 5. Training results on the FEI data set. 11](#_Toc41822212)

[Figure 6. Training results on the JAFFE data set. 11](#_Toc41822213)

[Figure 7. Training results on the Mixed (FEI + JAFFE) data set. 12](#_Toc41822214)

List of Tables

[Table 1. This table indicates test accuracy of the KNN when trained on different data sets 7](#_Toc41822846)

[Table 2. This table indicates f1 score of the KNN on the test data set when trained on different data sets 8](#_Toc41822847)

[Table 3. This table indicates test accuracy of the model when trained on different data sets 8](#_Toc41822848)

[Table 4. This table indicates AUC score of the model on the test data set when trained on different data sets 11](#_Toc41822849)

**Report**

# **Data preparation**

For the data preparation, I had a folder for each corresponding dataset. I had one folder for FEI Face dataset, one for JAFFE dataset, one for mixed (FEI + JAFFE) dataset and one for the test dataset. For FEI, JAFFE and mixed dataset I had two folders in each of them namely “train” and “validation”. And within each of these folders, there were two more folders namely “bad” and “good”. In these folders, the images were randomly distributed with the train folder having 75% of the images and validation folder having 25% of the images of the respective dataset. The advantage of this is I didn’t have to label the images rather I just checked when reading from which folder it was read and that would be the label for that image.

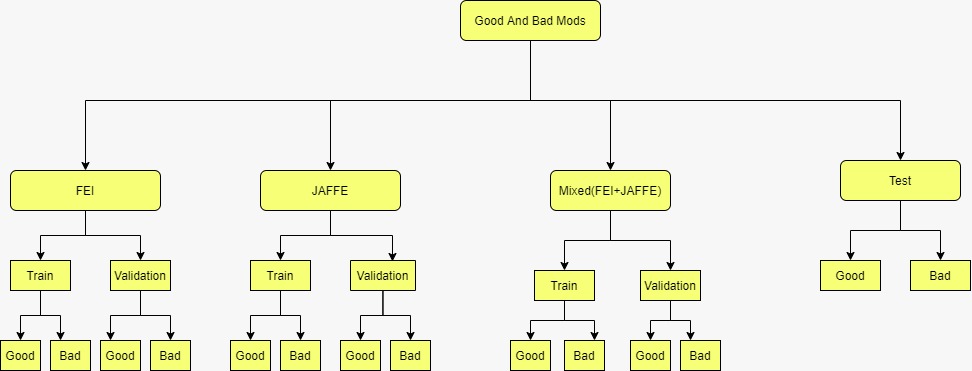


Figure 1. The list of directories where the data is stored.

# **Data pre-processing**

For the data pre-processing step, the technique applied for the machine learning classifier is different from deep learning the classifier so I will discuss them separately.

## **For Machine learning classifier**

For the machine learning classifier, the pre-processing steps were as follows:

1. Read the image through cv2.imread() function
2. Convert the RGB image to grayscale image.
3. Resize the image to 64 x 64.
4. Flattening it to get a single dimension

The resultant image was appended to a list of images.

## **For Deep learning classifier**

For the deep learning classifier, the pre-processing steps were as follows:

1. Set up the data extraction directory.
2. Set up the Image data generator and point it to the data extraction directory.
3. Configuring the data augmentation settings to pre-process the images:
   1. Rotation ranges upto 40 degrees.
   2. Width and height shift range by a factor of 0.2.
   3. Shear and zoom range by a factor of 0.2.
   4. Horizontal flipping.
4. Resize the image to 50 x 50.

# **Training**

As I already have the images and the labels so I will be using a supervised learning algorithm.

## **Machine learning classifier**

I chose the K nearest neighbor classifier. I also obtained results from the support vector classifier but the KNN produced better results than the former. So, I went with it in the end. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. I used the built-in KNN from sklearn and I set the neighbors to look at 3 which gave us the most optimal results. KNN was trained separately on FEI, JAFFE and then mixed (JAFFE + FEI) datasets. This could be seen in the jupyter notebook provided with this submission.

## **Deep learning classifier**

I built a custom deep learning model consisting of some convolutional, max pooling and dropt out layers. The convolutional layer passes a filter over an image to extract the corresponding features. The Max Pooling layer then extracts the most prominent features from the processed image. After a few layers, the fully connected layers flatten the image into a (n x 1) tensor which is then processed to output the result after passing through a sigmoid function. The output predicts whether the image is ‘good’ or ‘bad’. The Keras library was used for the development of the model. The model trained well on the FEI FACE dataset and the mixed (JAFFE + FEI) dataset. With respect to the testing accuracy, the model trained on the MIX dataset performed better than the models trained on other datasets.

# **Evaluation**

## **For Machine learning classifier**

There were two evaluation metrics which were used are accuracy and f1 score.

### **Accuracy**

Accuracy in terms of machine learning is defined as:

**Accuracy = Number of items correctly identified / Total number of items**

In the below table, the data set column indicates on which data set the KNN classifier was trained and the accuracy actually indicates the test accuracy (prediction on Joel’s private collection images).

|  |  |
| --- | --- |
| **Data set** | **Accuracy** |
| FEI | 62.5% |
| JAFFE | 40% |
| Mixed (FEI + JAFFE) | 42.5% |

Table 1. This table indicates test accuracy of the KNN when trained on different data sets

I can see from the above table that the KNN provided the best test accuracy when it was trained on the FEI data set.

### **F1 Score**

The F1 score is defined as follows:

**F1 score = 2\*((precision\*recall)/(precision+recall))**

Put another way, the F1 score conveys the balance between the precision and the recall.

In the below table, the data set column indicates on which data set the KNN classifier was trained and the F1 score on Joel’s private collection images.

|  |  |
| --- | --- |
| **Data set** | **F1 score** |
| FEI | 0.68 |
| JAFFE | 0.36 |
| Mixed (FEI + JAFFE) | 0.41 |

Table 2. This table indicates f1 score of the KNN on the test data set when trained on different data sets

I can see from the above table that the KNN provided the best f1 score on the test data set when it was trained on the FEI data set.

## **For Deep learning classifier**

There were two evaluation metrics which were used are accuracy and area under curve score.

### **Accuracy**

In the below table, the data set column indicates on which data set the Convolutional Neural Network was trained and the accuracy actually indicates the test accuracy (prediction on Joel’s private collection images).

|  |  |
| --- | --- |
| **Data set** | **Accuracy** |
| FEI | 52.5% |
| JAFFE | 50.0% |
| Mixed (FEI + JAFFE) | 67.5% |

Table 3. This table indicates test accuracy of the model when trained on different data sets

I can see from the above table that the deep learning model provided the best accuracy on the test data set when it was trained on the Mixed (FEI + JAFFE) data set.

### **Training Results**

#### **FEI Data set**

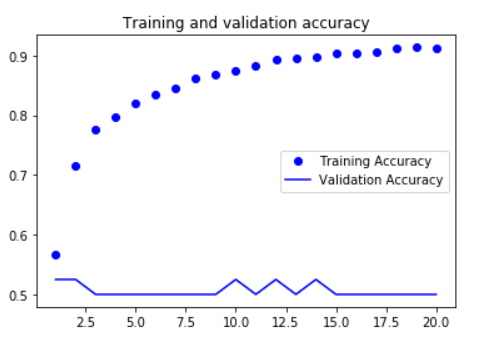


Figure 2. Training results on the FEI data set.

#### **JAFFE Data set**

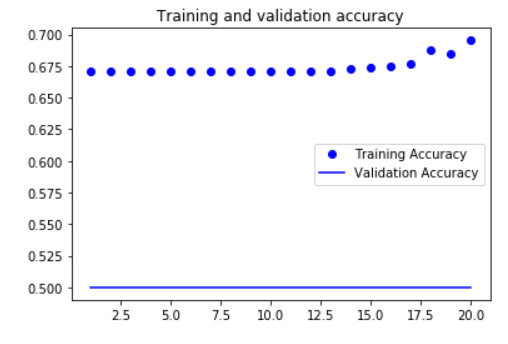


Figure 3. Training results on the JAFFE data set.

#### **Mixed (FEI + JAFFE) Data set**

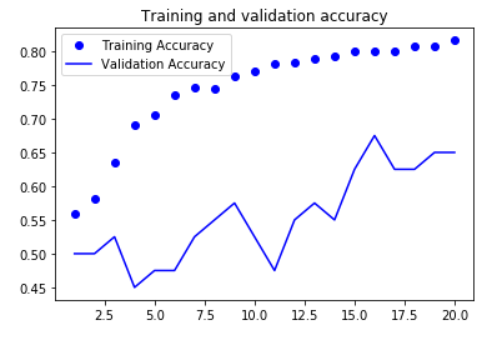


Figure 4. Training results on the mixed (FEI + JAFFE) data set

### **Area Under ROC Curve**

The AUC is defined as follows:

**AUC =** ∫ **TPR(x).FPR(x) dx**

The equation simply computes the 2 dimensional area under the ROC curve. It gives an aggregate measure across the 2 possible classification thresholds.

In the below table, the data set column indicates on which data set the Convolutional Neural Network was trained and the AUC score on Joel’s private collection images.

|  |  |
| --- | --- |
| **Data set** | **AUC score** |
| FEI | 0.605 |
| JAFFE | 0.719 |
| Mixed (FEI + JAFFE) | 0.760 |

Table 4. This table indicates AUC score of the model on the test data set when trained on different data sets

It can be seen from the above table that the deep learning model provided the best AUC score on the test data set when it was trained on the Mixed (FEI + JAFFE) data set.

### **Training Results**

#### **FEI Data set**

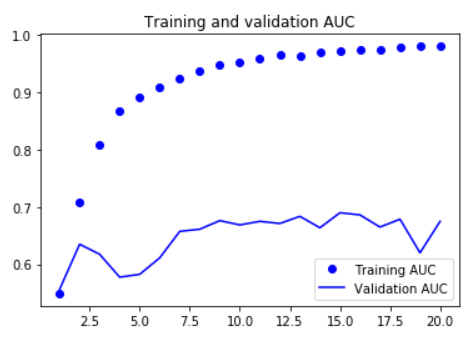


Figure 5. Training results on the FEI data set.

#### **JAFFE Data set**

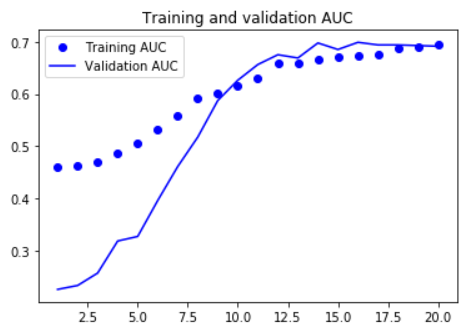


Figure 6. Training results on the JAFFE data set.

#### **Mixed (FEI + JAFFE) Data set**

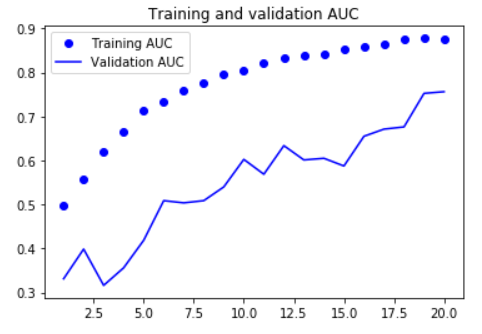


Figure 7. Training results on the Mixed (FEI + JAFFE) data set.

# **Conclusion**

Overall the accuracy reported is relatively low. There are numerous reasons for this. Firstly, with respect to the FEI data set, it had only cropped images of serious and smiling faces. However, if we look at the bad mood images of Joel's collection, it also included frustrated and crying images. Secondly, with respect to the JAFFE data set, there was a huge class imbalance due to there being more bad mood faces than there were cheerful, smiling faces. Also, the images were all of the frontal posture of the people. When I tried to resolve this issue by balancing the classes, the number of images decreased and hence it resulted in decreased accuracy of the models. The second try involved solving the issue by inserting new images after applying data augmentation; however, since the images got distorted, the accuracy decreased as a result.

As far as the machine learning approach is concerned, since there was little feature extraction process, the KNN was able to perform the best for the FEI data set as it assigns labels according to similarity to other images. As far as the JAFFE data set is concerned it had class imbalance in it so it performed poorly on it which could also be seen by the low f1 score.

As far as the deep learning approach is concerned, the model achieved high training and validation accuracy on its own data set (FEI and JAFFE) but showed mediocre performance on the test data set. The reason being the test data set contained images of a person with very different postures and clothes. While the model automatically learned many features on its own, it failed to properly take into account the different postures and clothes.