

# **CMP304 Coursework**

## **Machine Learning for Emotion Recognition**

---

**by: Austin A Flynn 1700403**

---

---

## 1. Introduction

---

The aim of the project was to create programs suitable for taking a large pool of pre-categorized images and creating an emotion predicting program using Machine Learning. The deliverables for the project are as follows;

Design a set of pre-processing steps to prepare the data.

Define a set of features to uniquely describe each emotion from the inputs, and give rationale for your choice.

Investigate different AI techniques that can be used for classification. Select one and give rationale for your choice.

Write an application that will act as a classifier for human emotions by implementing and integrating the above functions.

To best deliver on the above statements, I made two distinct programs to accomplish the process. The first part of the automated process is to convert the images into vectors formed from notable facial features. This is done twice, once for the training data and once for the testing data. The training data is then processed by the second program, where the Machine Learning (ML) takes place. The newly trained model is then exposed to the test data to gauge how accurate it is.

---

## 2. Methodology

---

The first step in the pipeline is to select which images are used for training and testing the model. For the database of pre-categorized faces I chose to use the [1]Cohn-Kanade database due to its large number of images and use of real human faces as opposed to the [2]FERG-DB with exaggerated animated faces. There are of course downsides to the C-K database, mostly stemming from most of the images being somewhat grainy, and occasionally over-saturated, potentially skewing the results. However, overall I believe C-K to have been the most well rounded database to use for the application.

The first bit of sorting was done manually, with 14 images from each of the 7 emotions being categorized (anger, disgust, fear, joy, sadness, surprise and neutral) being removed from the training pool and added to a separate testing pool. In retrospect it was perhaps not the best idea to have removed a flat number of images from each emotion, as emotions such as fear had around 1.5 times the number of images as anger, and this could have reduced the accuracy of the completed model. Another potential flaw in the project is the relatively low sample size, with only around 500 images being used for training. The alternative would have been to add images from other databases, however I believe this could have been detrimental to the project as a whole due to the different standards of the data sources. While most notable in the difference between real humans and animated models, inconsistencies in lighting or stage direction would be bad for the accuracy of the final model.

The next step in the process was the extraction and simplification of key facial features on these selected images. To extract the features, I elected to use dlib's 68 landmark model, as recommended by the lab work. I did look into other models for landmarking faces, such as a 5 feature one [3], however these were designed for increased efficiency for use on live video feeds, and lack the accuracy required for in depth facial feature extraction and analysis. The facial features which are extracted and exported to a .csv file are: Left Eyebrow, Right Eyebrow, Left Lip, Right Lip, Lip Height, Lip Width, Left Eye and Right Eye. These are calculated slightly differently depending on the specific feature, with the Lip Width and Lip Height being made of a single normalised distance, while the other features range from 3 normalised distances on

---

---

the Left Lip and Right Lip, all the way up to 4 normalised distances on the Eyes and Eyebrows. Each of these normalised vectors is stored as a double, where it is exported along with the correct emotion label from the dataset. This is applied by having the folders containing the images have the same name as their label, and then using the label itself as the last part of the file address. This process repeats until all images in the 7 emotion folders have been processed.

This process is repeated for the test data, except instead of exporting to a file called `feature_vectors`, it is exported to a file called `test_feature_vectors`. The process takes around 10 minutes for 600 images to be analysed in this manner, with 510 being used as training data, and 98 being used as testing data.

The learning stage of the process is managed mostly by ML.NET. For the actual training of the model I chose to use the `SdcaMaximumEntropyMulticlassTrainer` class as it is according to its Microsoft documentation [4] it is designed for use on normalized data being applied to multiclass classification which is exactly what the task at hand requires. The alternative I looked into briefly was a K means cluster, based on what Microsoft used in their tutorial for ML.NET multiclass ML [5], however overall it wasn't as good of a fit for the project as the final model of choice. After initial testing using only the facial landmarks recommended by the lab sheet I felt that the accuracy could be increased further with more features and so added the Eyes as a vector, which will be covered in more detail in the results section.

The trained model is exported at the end of this process so it can be used on other projects or tested further. In this case I tested the accuracy of the model using several metrics, the details of which will be discussed below.

---

### 3. Results

The results from the application went through a few stages, the first of these being the initial test without adding Eye vector information. In this scenario, the Micro-Accuracy was around 0.49 – 0.51. When considering that random guessing over the 7 categories would result in a 16.6% accuracy rate, this metric is within acceptable parameters, although definitely left room for improvement.

With the addition of the two Eye vectors, the Micro and Macro accuracy increased to around 0.51 – 0.55 depending on the test, an increase of around 4% accuracy.

The final set of results I recorded from the program were as follows; a Micro Accuracy of 0.551, a Macro Accuracy of 0.551, a Log Loss of 1.377 and a Log Loss Reduction of 0.292.

Breaking down these numbers shows that the machine does its job fairly well. The Micro Accuracy is an important indicator considering the previously mentioned imbalance in the number of training cases was not always equal. The accuracy being at 55.1% when a random guess has a 16.6% chance of being correct is a significant improvement, with the model being 3.31 times more accurate than guesswork alone.

The Macro Accuracy is identical to the Micro Accuracy at 0.551. This is likely because of there being an even number of test cases for each emotion being detected, meaning that as all the classes are treated equally it will never deviate from the Micro due to them being treated equally in the test data.

---

---

The Log Loss of 1.377 is well within acceptable parameters for a 7 class system, although despite looking into exactly how the log loss is calculated I am still unsure of its exact importance.

The Log Loss Reduction of 0.292 means that on average the program is 29.2% more accurate than randomly guessing the labels for the images tested on.

---

#### **4. Conclusion**

Overall the project can be considered to be reasonably successful as an introduction to Machine Learning. Although the pipeline I have created could certainly be streamlined in areas, and there are a few further steps I would like to have investigated had physical labs allowed for more direct interaction with lecturers. I am tempted to take the model further in the future with a larger dataset, which would hopefully be able to pair up with a live feed to try and register emotions in real time. The main weakness of the project as it stands is its limited accuracy. To solve this would likely require a much larger dataset, and a few tweaks to the facial vector data, however for now I feel it is a satisfactory first attempt.

---

#### **5. References**

- [1]Cohn-Kanade AU-Coded Facial Expression Database: Kanade, T., Cohn, J.F. and Tian, Y., 2000, March. Comprehensive database for facial expression analysis. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition* (Cat. No. PR00580) (pp. 46-53). IEEE
  - [2] Facial Expression Research Group Database (FERG-DB): Aneja, D., Colburn, A., Faigin, G., Shapiro, L. and Mones, B., (2016). Modeling stylized character expressions via deep learning. In *Asian conference on computer vision* (pp. 136-153). Springer, Cham
  - [3] <https://www.pyimagesearch.com/2018/04/02/faster-facial-landmark-detector-with-dlib/>
  - [4]<https://docs.microsoft.com/enus/dotnet/api/microsoft.ml.trainers.sdcamaximumentropymlclasstrainer?view=ml-dotnet>
  - [5] <https://docs.microsoft.com/en-us/dotnet/machine-learning/tutorials/iris-clustering>
-