CMP304 Face Application Report

Contents

Introduction

Affective computing is a field that tries to make a computer be able to understand emotions of a person and respond to them. The recognised emotions can then be used to aid in fields such as user testing/user sentiment, machine diagnosis, and better interaction with people through digital assistants (Picard, 2003).

Various measures can be used to determine the emotion of the user, however among the least intrusive is facial recognition. Facial recognition relies on images of the user's face, which are then analysed and processed by computer algorithms.

One of the existing solutions to facial recognition is by Munashighne (2018), which uses a library to extract points on a face, which are then used to prepare a feature vector from the Kanade et. al. (2000) image set and pass it to a machine learning algorithm. Sixty-eight points are generated by the facial extraction library Dlib, from which lengths are derived, then normalized to generate six data points for a face. This approach was trained to detect four emotions (anger, happiness, sadness, and surprise), as extracted from the image location, with an average ninety percent accuracy. The machine learning algorithm is a forest classifier, which is a set of binary questions that the computer evaluates to determine a face.

Another solution to facial recognition is by Vangent (2016), which uses largely the same approach (feature extraction, then subsequently using a machine learning algorithm to train on) as one by Munashighne (2018), however the feature vector is calculated differently. First the centre of the face is calculated, then its orientation, and subsequently the angle between the centre of the face (with the angle corrected according to orientation). Subsequently the distances between the centre, along with the raw coordinates are added to the feature vector. This approach yields over two-hundred features per face and the author claims that it gains an accuracy of seventy eight percent on seven emotions (anger, disgust, fear, neutral, happy, sad, surprised). This approach was further tested on a linear and polynomial support vector machine (which rather tries to calculate the line of best fit) as well as a random forest classifier leading to ninety three, eighty three, and eighty eight percent accuracy respectively.

The current solution aims to build on these methods and make a facial emotion identifier using feature vector extraction and a machine learning algorithm. This approach implements both of the solutions to try to derive which one is better, and then builds upon it to try to find the best possible solution.

There were three key approaches used in the calculation of the feature vectors, with one as described by Munasinghe (2018) which uses the Dlib library to extract 'feature vectors' – values that are derived from the various points on the image that are generated by Dlib. Another approach was used, as described by Vangent (2016) which used the same library, however prepared the feature vector differently. The last approach was an attempt to improve the Munasinghe (2018) method, with use of some of the feature vector calculation techniques as defined by Vangent (2016) to improve the accuracy of emotion prediction.

The emotions (happiness, anger, sadness, disgust, fear, surprised, neutral) were selected as they are the basic emotions to be displayed by people, and the accompanying data sets were organised in such an order (Munasinghe, 2018). The original method described by Munashighe (2018) does not take the neutral facial expression into account, and the method described by Vangent (2016) was tested across eight (which included contempt) and five (leaving out contempt, fear and sadness) with both achieving above 60% accuracy.

The extracted features were then processed by a multiclass machine learning algorithm, as provided by the machine learning library ML.NET. This library was used due to its ease of use and the relatively little setup required. Multi-class classification was used, as this was the described method in Vangent (2016) and Munasinghe (2018). While the underlying classification is

This project aims to compare two feature extraction methods as described by Munasinghe (2018) and Vangent (2016) and come up with a separate feature extraction method which would aim to generate better accuracy than either.

Methodology

The application uses ML.NET, which uses the SCDA maximum entropy multi-class classifier. This contrasts with the solutions of Munasinghe (2018) and Vangent (2016) which use linear scalar vector machines and random forest classifiers. The SCDA maximum entropy multi-class classifier was chosen as it determines weights of the different data points regarding the emotions they represent (Vryniotis, 2013). This trainer was chosen as the features are not necessarily separate from each other. For example, a high eyebrow width and low mouth angle may determine a different emotion from having a high eyebrow width and a high mouth angle.

The features are first calculated then given to the machine learning algorithm using the SCDA maximum entropy multi-class trainer. This trainer determines the various parameters with minimal input required from the programmer (How to choose an ML.NET algorithm, n.d.). The trainer works by attempting to estimate the probability distribution from the training data, and then attempting to converge on the global minima of probability – as it assumes that there are no local maxima by using a hill climb function. The initial probability distribution is estimated by analysing the training data without the labels and then the probability distribution is adjusted depending on the values of the parameters (Nigam, Lafferty and McCallum, 1999).

The features were calculated in the same ways as described in the existing solutions, except adapted into ML.NET data structures. This meant using the Dlib library to extract features and using a data set to train and verify the model on. The Kadane et. al. (2000) dataset was chosen as it contains real life subjects with around fifty test images per image, giving six hundred in total. Another contender was the data set by Aifanti et. al. (2010) which had realistic data subjects, however had less images, with a total of four hundred and six and had a progression of emotions rather than the very distinct emotions that the training sets of Munasinghe (2018) and Vangent (2016) used. These two data sets were considered as their images were relatively uniform (front facing with no obscuring features) and did not require any processing to crop etc. (aside from extracting points using Dlib). The reason a realistic data set was chosen, instead of one that was computer generated is that the resulting algorithm can be tested on other data sets.

The Dlib feature extraction library for C# (Takeuchi, n.d.) was chosen as it was the main method described by the other publications, moreover the sixty-eight waypoint option was chosen as the other solutions (by Munasinghe (2018) and Vangent (2016)) used it to reach high accuracies. The points were then extracted and then processed in methods as described in the existing solutions.

Face Data 1 represents the individual parts of the feature vector as described Munasinghe (2018), and are calculated by finding the distance between points 18, 19, 20 and 23, 24, 25 (eyebrows) as normalized by the distance between 21, 39 and 42, 22 (between inner most eyebrow point to tearduct area of eye) respectively for left and right eyes; distances between the tip of the nose and top of lip, width and height of the mouth which are normalized by tip of the nose (33) and top of the lip (51).

Face Data 2 represents the individual parts of the feature vector as described Vangent (2016) which stores the distance, angle and raw coordinates of every point. The points are calculated by first getting the average point of the face by adding all coordinates up and averaging them, then distance and angle between every point and the centre is calculated. The distance is normalized by the point between the tip (30) and end of the nose (27), and the angle is made to be relative to the rotation of the face in the image by calculating the angle of the aforementioned points relative to up.

Face Data 3 represents the self-made feature vector and extraction, based on parts by Munashinghe (2018) and Vangent (2016). It stores the average of the eyebrow distance, the width & height of the eye the inner and outer lip width & height and the angle between the edges of the mouth (48, 54) by calculating a unit vector and converting it to an angle, in radians (as there is less variation in values than storing in degrees). The lip rotation was inspired by the implementation provided by Vangent (2016), their implementation stores the rotation of all of the points relative to the average point of the face. The same approach was taken when calculating the angle of the lip. The inclusion of this feature was added to add more unique data points than feature vector 1, especially as it proved promising when differentiating emotions such as happiness (Figure 0)

Unlike Munashighne (2018) which uses multiple normalization points, the implementation of face data 3 uses the points in the nose as the main normalization point. Otherwise a few more data points are provided such as the inner and outer lip instead of just the outer lip for extra information as the thickness of lips can change depending on how stretched they are. The distance with the eye brows from the centre point was taken from Vangent (2016).

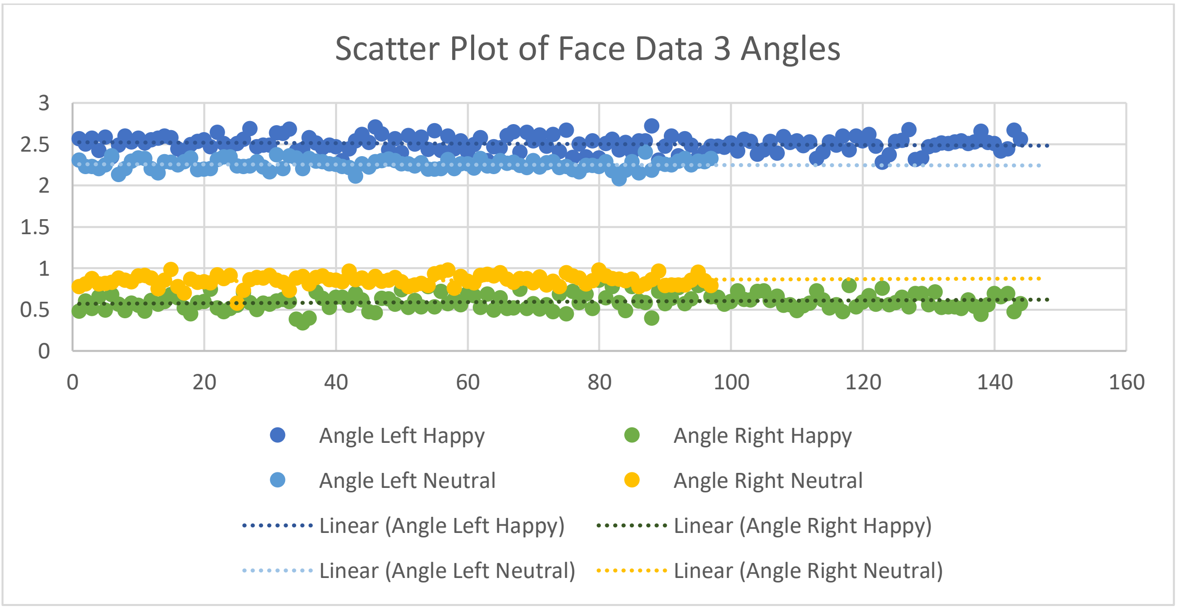

Figure 0 – Scatter graph of face data 3 lip angles.

The evaluation was performed by changing the number of maximum iterations and logging the log loss, log loss reduction, macro accuracy, micro accuracy, time taken, and the confusion matrix five times. The original Kadane et. al. (2000) dataset was split into a test and training set. The test set was twenty percent of the original set, and the evaluation was performed on the test split, and trained on the train split. The maximum number of iterations is the main variable changed as it determines the training time (Figure 1), which is one of the primary variables that affects accuracy of the model (Figures 2 & 3). The main data points displayed on the graph are the median values of the runs, as a normal distribution is not necessarily present. The iterations were increased by a factor of

10 as values indicating performance such as log loss gradually plateau or start getting worse (Figure 4), in addition to time constraints as the time taken to run one million iterations of face data 2 took roughly twenty five minutes, and with five re-runs to ensure data accuracy it takes two hours to run to get the results for a single face data (about six hours in total).

Results and Discussion

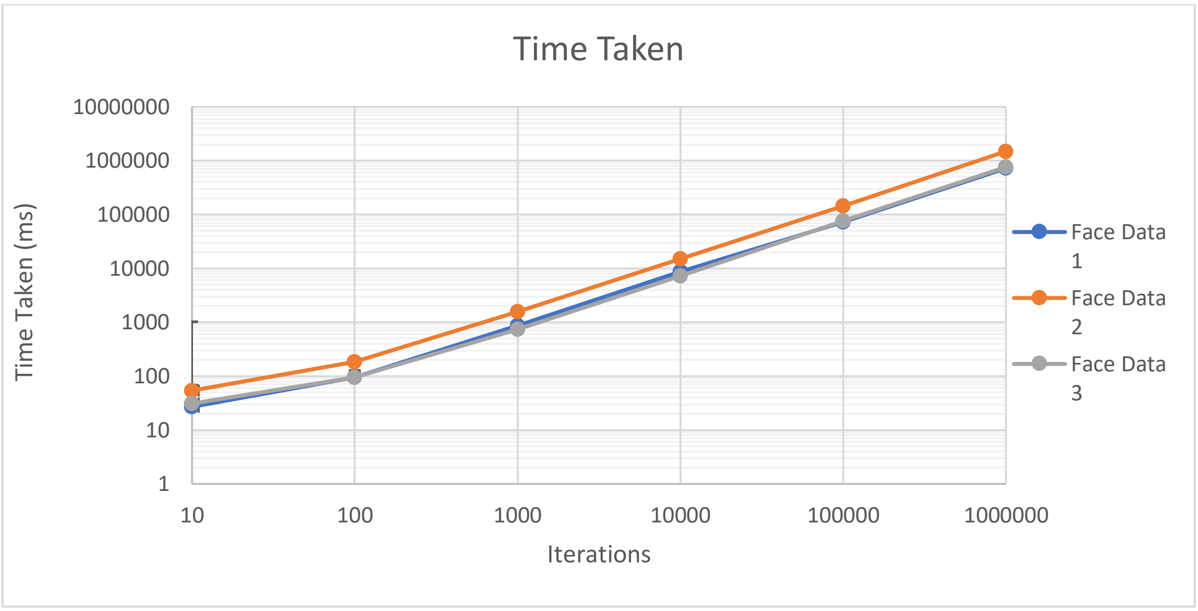

Figure 1 - Time taken vs iterations.

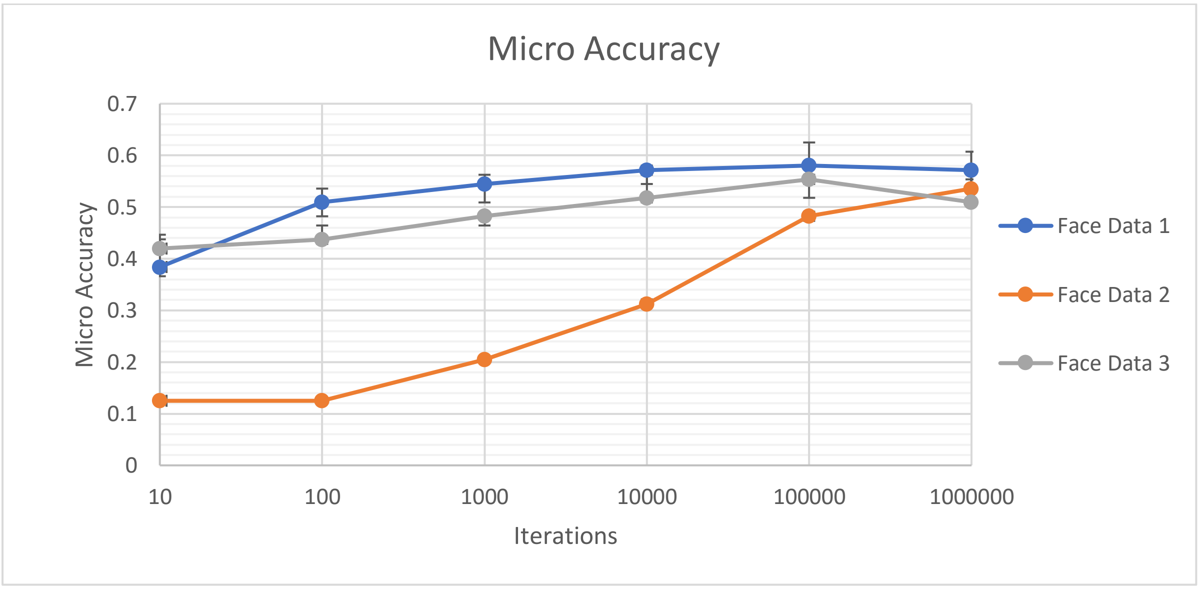

Figure 2 – Micro accuracy.

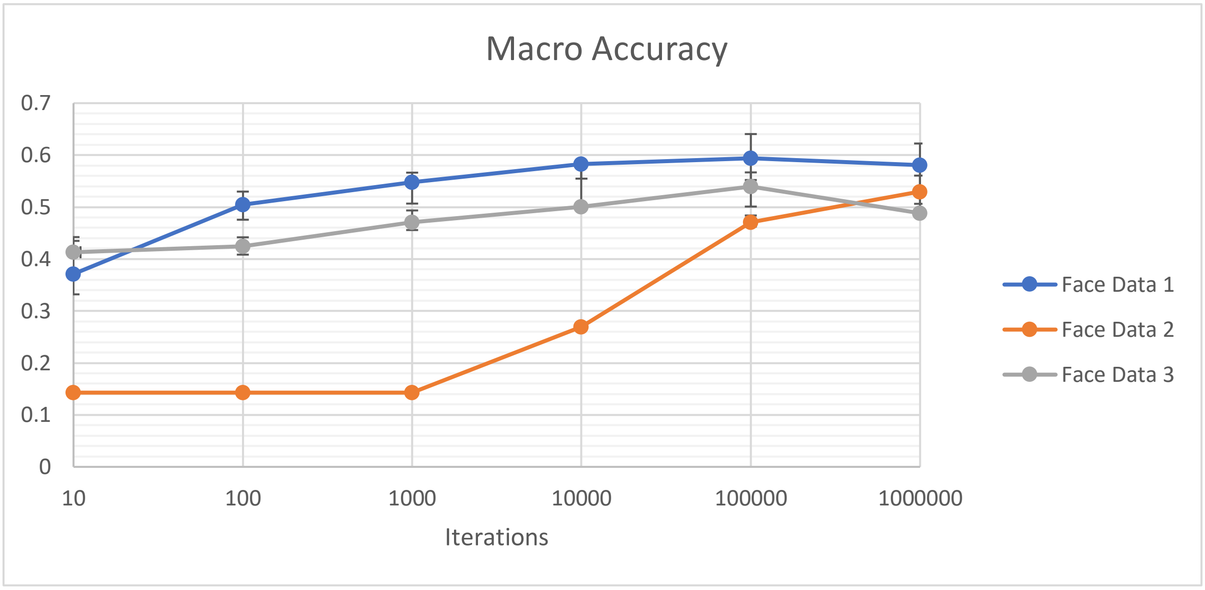

Figure 3 – Macro accuracy.

Face data 1 has the best accuracy overall, with face data 3 close behind. The accuracy of face data 1 and 2 peaks around one hundred thousand iterations and starts decreasing at one million iterations, however face data 3 has not appeared to have gone past its peak accuracy. This could be due to the amount of information points in face data 3 (over 250) and extra iterations it would take to converge on the appropriate probability distribution, as shown by the increasing accuracies and varying log and log loss values (figures 4 and 5). Face data 1 and face data 2 appear to have converged at one hundred thousand iterations and their iterations started decreasing after. This could be due to the model going past the global maxima and starts 'climbing' again, decreasing accuracy or overfitting.

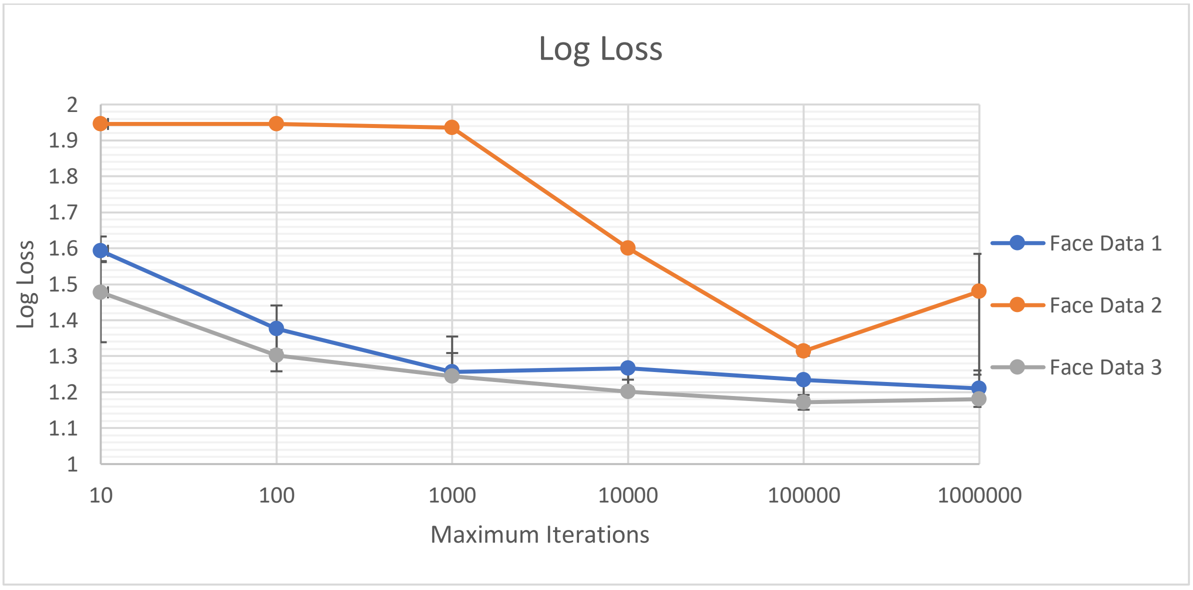

Figure 4 – Log loss.

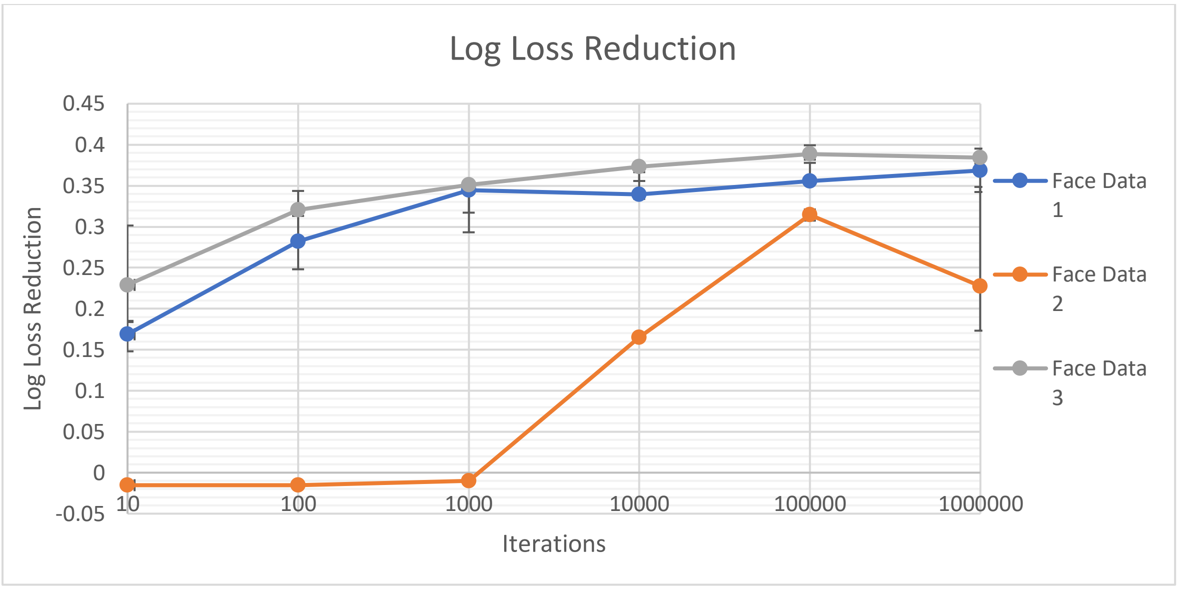

Figure 5 – Log Loss Reduction

The log loss and log loss reduction give further insight into the accuracy of the model. Log loss measures the probability of how much the actual label varies, with the best value being zero. The log loss reduction shows how much better than the predictions are than guessing randomly, with a log loss of one being best.

In this case figures 4 and 5 give further insight to the results of figures 2 and 3 (that the model could be overfitting). In this case it appears that face data 2 has a strong variation between results at one million iterations, with the general log loss going away from its peak at one hundred thousand iterations, unless the outlier at one million iterations is included. This suggests that face data 2 may benefit from the use of cross validation and then picking the best model, as cross validation trains multiple models and allows the programmer to pick the best one (Train a machine learning model using cross validation, n.d.).

Face data 1 appears to still have a decreasing log loss which suggests that it may benefit with more iterations. Face data 3 appears to have values that are better than face data 2 and 3, implying that it is less likely to mislabel and is slightly better at predicting than the other face data. This could be due to the addition of the lip rotation distance.

Face Data 1 CM	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**Recall**
0. neutral	11	0	1	3	9	2	3	0.785714
1. surprise	0	14	0	0	0	0	0	1
2. sad	1	0	12	3	5	2	0	0.923077
3. fear	1	0	0	8	2	2	0	0.470588
4. angry	0	0	0	0	0	0	0	0
5. disgusted	0	0	0	0	3	4	0	0.363636
6. happy	1	0	0	3	1	1	20	0.869565
Precision	0.363636	1	0.5	0.533333	0	0.5	0.769231	

Figure 6 – Median confusion matrix of Face Data 1 at one hundred thousand iterations.

Face data 2 CM	0	1	2	3	4	5	6	Recall
0. neutral	8	0	6	2	2	0	3	0.571429
1. surprise	0	12	0	0	0	0	0	0.857143
2. sad	1	0	6	4	2	0	0	0.461538
3. fear	1	1	1	6	2	0	3	0.352941
4. angry	1	1	0	1	9	6	0	0.45
5. disgusted	1	0	0	1	3	4	0	0.363636
6. happy	1	0	0	3	2	1	16	0.695652
Precision	0.363636	1	0.428571	0.4	0.5	0.5	0.68	

Figure 7 – Median confusion matrix of Face Data 2 at one million iterations.

Face data 3 CM	0	1	2	3	4	5	6	Recall
0. neutral	7	0	4	3	5	1	2	0.5
1. surprise	0	13	0	0	0	1	0	0.928571
2. sad	3	0	6	3	7	1	0	0.461538
3. fear	0	0	1	6	1	0	1	0.352941
4. angry	3	1	1	2	4	4	0	0.2
5. disgusted	0	0	1	0	2	4	0	0.363636
6. happy	0	0	0	3	1	0	20	0.869565
Precision	0.333333	0.928571	0.3	0.625	0.307692	0.571429	0.8	

Figure 8 – Median confusion matrix of Face Data 3 at one hundred thousand iterations.

The confusion matrices were picked from the iterations that yielded the best accuracy as per figures 2 and 3. This data reinforces the data shown by figures 2 and 3, but also gives insight to which emotions are classified the best. In this case it appears that face data 1 classified emotions the best overall. However, face data 1, like wise with face data 2 and 3 had poor results when classifying fear anger and disgust. In the case of face data 1 and 3, which appear to have reached their minimum log loss values, the addition of more hyperparameters would have to be investigated. In the case of face data 2 an increase in the amount of iterations would be required to be able to correctly identify such emotions, similar to the one performed in figure 0.

Face data 1 and 3 perform poorly suggesting that additional hyperparameters may need to be added, or reviewed. The storage of both lip widths may have impacted this as closed mouths have a generally lower inner lip height, which is typical of both emotions as shown by the data set. Another review that may be required is into the storage of the eyebrow heights as that is what appears to differ the most (Figure 9) between the emotions.

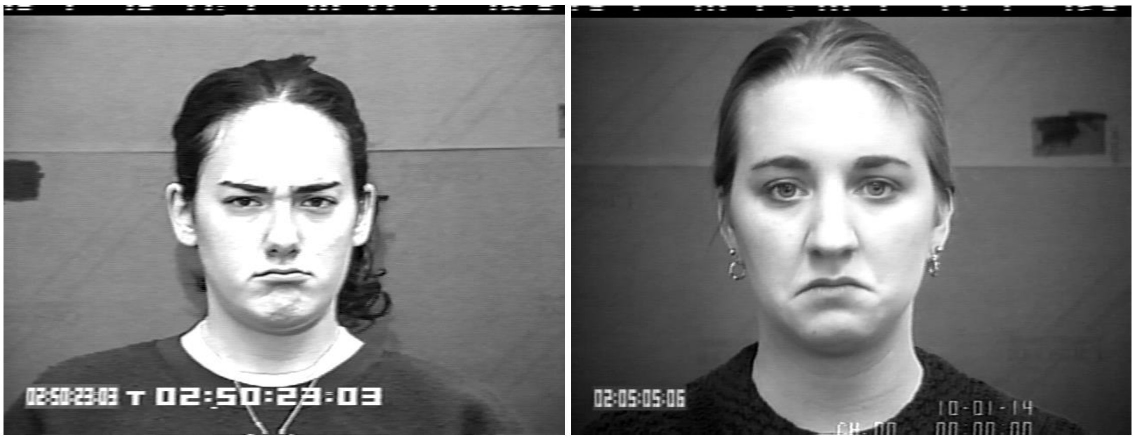

Figure 9 – Angry expression on left, sad expression on right.

Conclusion

The goal of the project was to evaluate and compare two different methods of feature extraction as described by Munasinghe (2018) and Vangent (2016) and attempt to create a better feature extraction method which would ideally improve their accuracy.

The performance was measured by changing the iterations as it appears through figures 1, 2 & 3 that training time (and therefore amount of iterations) influences the accuracy. The extracted facial features trained using an SCDA maximum entropy multi-class classifier as it is able to infer the connections between the features.

The overall results pointed to Munasinghe (2018) method being the best option for a lower training time, with Vangent (2016) method requiring more training and potentially chainging the way of training to cross validation as there was a wide difference in accuracy in the end. The created feature extraction set was slightly worse performing than Munasinghe's (2018) method potentially due to the way eye brows were extracted (which was based on an average distance, unlike Munasinghe (2018) whose method used the total distance between the eye brows), and potentially lacking enough parameters to be able to differentiate between the various facial features.

Another factor which may have linked into the accuracy is the data set, as per Figure 9 the data set is imbalanced, containing a disproportionate number of items inside of the happiness label. This could lead to the over training of the data set on the happiness, affecting the accuracy. In addition, the data set was rather small at six hundred and eight images, which could further affect the accuracy of the classification algorithm. In addition, the Kadane et. al. (2000) data set upon informal inspection seems to use far more exaggerated emotions than the Aifanti, Papachristou, and Delopoulos (2010) data set, although there are a lot more faces in the Kadane et. al. (2000) data set – rather than the progression of emotions found in the former MUG data set.

Emotion	Num. Images	Percentage
Neutral	97	16%
Surprise	84	14%
Sadness	81	13%
Fear	79	13%
Anger	70	12%
Disgust	53	9%

Happiness	144	24%

Figure 9 – data set accuracy.

The testing was also performed on a part of the data set, which further reduced the amount of possible features used for training. Instead, and to make sure that the maximum amount of data is available for the training a separate data set could be used. This would require further research into data sets that meet the criteria of Munasinghe (2018) and Vangent (2016) original data sets.

Some further processing could be done on the data set to create more images such as mirroring as the features are distinctly labelled left or right. This could potentially help with accuracy as there are more data points to train on and allow the algorithm to detect mirrored images.

It is also worth noting that Vangent (2016) solution used a custom-made data set, which would perform differently, and Munasinghe (2018) did not measure for all seven emotions in his tests. However, Munashinghe (2018) approach still does better than face data 3 and Vangent (2016) on Kadane et. al. (2000) data set.

In addition, it would be beneficial to add more benchmarking functionality into the application, such as compiling graphs as seen here to allow easier testing and compilation of results.

Overall the goal of the project has been mostly met. The two methods facial feature extraction methods between Munasinghe (2018) and Vangent (2016) were compared, and it was found that Munasinghe's (2018) method was better in this circumstance. From this 'face data 3' was created which is primarily based on the feature vector provided by Munasinghe (2018), with variations in how normalization and eyebrow height is done, but also the addition on the lip angle. Unfortunately, face data 3 has not generated a better accuracy than Munasinghe's (2018) approach, however it is slightly better one than Vangent's (2016) approach at its best while taking substantially less time to train.

References

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

Picard, R., 2003. Affective computing: challenges. *International Journal of Human-Computer Studies*, 59(1-2), pp.55-64.

Munasinghe, M.I.N.P., 2018, June. Facial expression recognition using facial landmarks and random forest classifier. In 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS) (pp. 423-427). IEEE.

Vangent, P., 2016. *Emotion Recognition Using Facial Landmarks, Python, Dlib And Opencv – Paulvangent.Com*. [online] Paulvangent.com. Available at: <http://www.paulvangent.com/2016/08/05/emotion-recognition-using-facial-landmarks/> [Accessed 9 July 2020].

Vryniotis, V., 2013. *Machine Learning Tutorial: The Max Entropy Text Classifier | Datumbox*. [online] Datumbox. Available at: <https://blog.datumbox.com/machine-learning-tutorial-the-max-entropy-text-classifier/> [Accessed 11 July 2020].

Aifanti, N., Papachristou, C. and Delopoulos, A., 2010, April. The MUG facial expression database. In 11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10 (pp.1-4). IEEE

Microsoft Docs. n.d. *How To Choose An ML.NET Algorithm*. [online] Available at: <https://docs.microsoft.com/en-us/dotnet/machine-learning/how-to-choose-an-ml-net-algorithm> [Accessed 12 July 2020].

Nigam, K., Lafferty, J. and McCallum, A., 1999. Using Maximum Entropy for Text Classification. [online] Available at: <http://www.kamalnigam.com/papers/maxent-ijcaiws99.pdf> [Accessed 15 July 2020].

Takeuchi, T., n.d. *Dlib Dot Net*. [online] GitHub. Available at: <https://github.com/takuya-takeuchi/DlibDotNet> [Accessed 15 July 2020].

Microsoft Docs. n.d. *Train A Machine Learning Model Using Cross Validation*. [online] Available at: <https://docs.microsoft.com/en-us/dotnet/machine-learning/how-to-guides/train-machine-learning-model-cross-validation-ml-net> [Accessed 16 July 2020].