Intelligent Interactive Systems
Project Report by Group 12

Diego Castillo Ankur Shukla Tristan Wright

June 1, 2018

1 Introduction

The Bosphorus Database is a dataset intended for research of 2D and 3D human face processing tasks. The analysis and classification of the project is to assign one of the six basic emotions (*anger*, *disgust*, *fear*, *happy*, *sadness*, and *surprise*) to a set of labeled facial landmark positions using machine learning algorithms.

1.1 Analysis Summary and Main Results

This report provides a detailed explanation of the analysis performed to the dataset and data preprocessing tasks such as data cleansing, normalization, dimensionality reduction, class imbalance analysis, holdout validation, and finally a discussion of the four different machine learning methods implemented; k-Nearest Neighbors (kNN), Support Vector Machine (SVM), Multilayer Perceptron (MLP), and Random Forest. After performing the former analysis and using the reduced features by the principal component analysis (PCA) of the 3D landmark files ('.lm3') with the corresponding labels, both the SVM and MLP models performed best, consistently achieving an accuracy of $70 - 80\%$, in contrast to the kNN and Random Forest models which obtained an average accuracy of $50 - 60\%$.

1.2 What Has Worked, What Has Not

Our first approach consisted of doing a feature transformation of the original measurements by computing the distance from each feature to the 'Nose Tip' as well as removing features which we thought would have little variation regardless of the emotion. The feature transformation was implemented using either the Euclidean or Manhattan distance, but the accuracy achieved by the classifiers was lower than what these were able to obtain by simply using the original features. Other techniques, such as normalizing and dimensionality reduction with PCA, proved to be of great importance for the classifiers to improve their respective accuracies.

1.3 Role Played By Group Members

Diego Castillo was a developer and integrator. Ankur Shukla was also a developer and integrator. Likewise, Tristan Wright was a developer and integrator.

1.4 Group Members Contributions

Diego Castillo worked in the implementation of the '.lm2' parser, PCA and t-SNE dimensionality reduction techniques, the randomized holdout validation, and the implementation of the kNN and SVM models. Ankur Shukla created the emotion synthesis output format, implemented the confusion matrix plot, made experiments with k-fold and leave-one out validation, and implemented the random forest classifier. Finally, Tristan Wright was in charge of creating the computer vision API format as well as parsing and classifying it, implemented the MLP classifier, the '.bnt' and '.lm3' parsers, and recorded the video showcasing the project.

2 Data

Facial landmarks are key points which allow to perform tasks such as emotion classification. These facial landmarks (features) –for example the location of the outer left eye brow– are provided in the dataset in three different types of files. Files with an '.lm2' extension have 2D feature coordinates, '.lm3' have 3D feature coordinates, and finally the ‘.bnt’ files have both 2D coordinates and corresponding 3D image coordinates, each labeled with the corresponding emotion it represents.

There are a total of 453 samples, each with an ‘.lm2’, ‘.lm3’, and ‘.bnt’ file. A single file corresponds to one out of the six possible emotions. Of these 453 samples, 71 correspond to ‘anger’, 69 to ‘disgust’, 70 to ‘fear’, 106 to ‘happy’, 66 to ‘sadness’, and 71 to ‘surprise’. Since some labels have more samples than others, the dataset suffers from class imbalance. Class imbalance is a phenomenon that usually results in models which tend to focus on the classification of the samples which are overrepresented while ignoring –or misclassifying– the underrepresented samples [2]. Section 3 explains how this issue was addressed.

For each possible file extension a parser was created to extract the features for the emotion classification task. The parsers find all files with a particular extension and create a ‘.csv’ file which consists of all the samples with their corresponding label, and all features associated with each sample.

2.1 Data Cleansing

Each sample can have at most 26 features, but in certain occasions it is not possible for some of these to be defined. The parsers address this issue by setting the value of the feature to ‘NaN’ across all the dimensions which the file being parsed defines. In the analysis and classification all features which have at least one undefined (i.e. ‘NaN’) value are removed.

2.2 Data Transformation

As explained in section 1.2, a few data transformation strategies were tested. Since none of these achieved positive results, the original features with the removed undefined values were used during the rest of the analysis.

2.3 Normalization

When distance is used to compute the difference between two or more distinct samples, features that have a broad range of values will dominate the analysis. Normalization, a technique to avoid such a problem, refers to the process of standardizing the values of independent features of a dataset [2]. All features in the dataset were normalized such that each feature contributed approximately proportionately to the computation.

2.4 Dimensionality Reduction

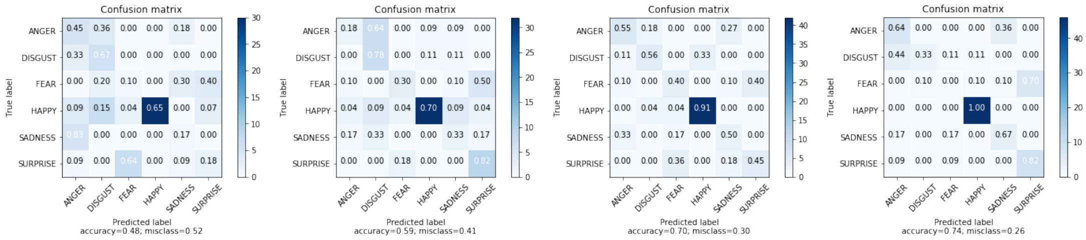
After the features in the dataset were cleansed, a total of 58 features remained (when using the ‘.lm3’ dataset). Machine learning algorithms based on distance perform better if the number of features is low [2]. Hence, both PCA and t-SNE were implemented to reduce the number of features. A divide-and-conquer approach was used to determine the number of features to reduce to. For both dimensionality reduction techniques, a reduction to 15 components achieved the best results. As already stated in section 1.1, the PCA reduction obtained the best results.

3 Implementation

Once the features in the dataset had been cleansed and their dimensionality reduced, the k-fold, leave-one-out, and holdout validation methods were implemented to determine how to train the models. The holdout validation method performed best out of the three.

The holdout validation method consists of splitting the dataset in two independent sets: a train and a test set. The training set is used to train the model, while the test set is used to evaluate its performance. Since the dataset has class imbalance, a total of 60 samples (∼80%) per label were randomly selected to be inserted in the training set, while the remaining samples (∼20%) were inserted in the test set. Hence, even though the dataset has class imbalance, the different models were trained using the same number of samples per label.

What follows is a short description of each of the models implemented and the hyper-parameters used to achieved the best results.



(a) kNN

(b) RF

(c) MLP

(d) SVM

Figure 4.1: Confusion matrices for methods run with PCA dimensionality reduction.

3.1 kNN

A kNN classifier simply classifies samples by a majority vote of its neighbors, with a sample being assigned to the class most common among its `k` nearest neighbors [2]. The hyper-parameter `n_neighbors` determines the number of neighbors to use to choose the label of a sample, a total of 15 neighbors attained the best results.

3.2 SVM

A SVM is a supervised learning algorithm which creates a hyperplane that represents the input samples as points in space, such that samples of different classes are separated from each other. New samples are then mapped into that same space and predicted to belong to a class based on where in that space these are located [2].

3.3 MLP

A MLP is a feed-forward artificial neural network that uses supervised learning with back propagation for training the network (update its weights) to essentially map a weighted input to a label. The MLP uses the `adam` solver, which is a stochastic gradient-based optimizer; a logistic sigmoid function as its activation function, a single hidden layer with a total of 5 neurons, and a maximum of 2000 iterations to reach convergence.

3.4 Random Forests

A random forest (RF) is an ensemble learning method for classification that works by creating multiple decision trees during training and predicting the class that is the mode of the classes of the individual decision trees [2]. A total of 10 trees in the forest, with the `GINI` splitting criterion, and a maximum depth of 6 per tree achieved the best results.

4 Results

Each subsection here is dedicated to a method and is accompanied with two confusion matrices for t-SNE and PCA each. The matrix is from a single run of the model. Results were measured by running the model 100 times on different random splits of data which has been preprocessed with PCA methods discussed in implementation. From these runs we can extract the average accuracy and standard deviation for each model and determine the best one.

4.1 kNN

With kNN methods we can attain an average accuracy of 46.5% with a standard deviation of 0.045; this is a mediocre score. This difference in accuracy compared to other methods can be thought of as a problem with clustering and the nature of the landmark points. Any set of emotions where, for example, an eyebrow point is more or less the same point (e.g. surprised and happy), will be classified by kNN as

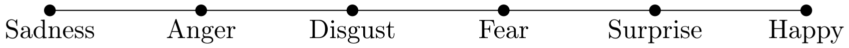


Figure 4.2: A spectrum of emotions inferred from confusion matrices.

the same emotion, no other points will be taken into account. Among our labels are two super-categories of emotions, positive (happy, surprised) and negative (fear, angry, disgust, sad). The reader is invited to make those emotions with his or her face and take note the placement of their eyebrows. They may notice they are in the same place for each emotion, other features may be in the same place too.

We ran another experiment to demonstrate that kNN can detect these distinctions. Simply by running the dataset through the kNN classifier and counting how many positive and negative emotions it accurately detects. The results are telling, we can classify positive emotions with 67% accuracy and negative emotions with 89% accuracy. We can inspect this further by referring to the relevant confusion matrix (figure 4.1a), kNN has quite admonishable accuracy when trying to classify the positive emotion surprise.

4.2 Random Forests

With RF methods we can attain an accuracy of 62.9% with a standard deviation of 0.043. While consistent, compared to the worst of our supervised methods it is still about 10% less accurate. Furthermore, unlike kNN methods, RF's inherit structure determines features to be correlated, it is not just a series of points which some method is blindly clustering, hence the 15% accuracy boost over kNN. An interesting difference to point out in the results is while kNN most misclassifies surprise, RF misclassifies anger as disgust (figure 4.1b). It is also worthwhile noting that RF is the best method with t-SNE, although the accuracy is only barely better than random classification at 28%.

4.3 MLP

With MLP methods we can attain an accuracy of 73.3% with a standard deviation of 0.037. This immediate increase in accuracy can demonstrate the advantages of fully using labels in a dataset. Unlike kNN methods, neural networks can fit correlations and relations in datasets, although perhaps over-fit in some places. After a closer look at the confusion matrix in figure 4.1c we see a faint bottom-left to top-right diagonal from anger to sadness consisting of commonly misclassified emotions.

4.4 SVM

With SVM methods we get an accuracy of 74.00% with a standard deviation of 0.045. To a marginal detriment of consistency against MLP methods, this makes it the most accurate model. It is also the only model to score 100% for a single emotion “happy”. However, along with MLP methods, it misclassifies fear for surprise quite often (figure 4.1d).

By inspecting which emotions are most misclassified as others across all PCA figures for each classifier we can intuitively create an emotional spectrum (figure 4.2).

5 Secondary Groups

Our task in the final pipeline was to read the input given by Computer Vision (CV) group and send the output to Emotion Synthesis (ES) group.

5.1 Computer Vision Group

After training our models we need to predict an emotion by reading input from the CV group. They are creating an array containing 25 landmarks and passing it as input into our models. PCA was used for dimensionality reduction before training our model, so the same PCA model is used on the given input for reducing the number of features.

5.2 Emotion Synthesis Group

We are using an SVM model as it is the most stable and gives the best accuracy as seen in the Results section. We are sending output in a Python dictionary format which contains all six labels with their confidence score. Confidence scores tells us about the probability of the emotion predicted by our model.

5.3 Challenges

Each group worked on a different operating system, this was a huge problem when we met to integrate all of our solutions. Having written all of our code in Python, we were the most platform agnostic group and integrating our code in either direction (toward CV or ES) was trivial. What was quite difficult was getting ES code to work on the CV groups computers and vice-versa. From the CV group, at first, they were not delivering the 25 landmarks in the same order as we were expecting, in less than 20 minutes of collaboration we were able to correctly order the input. Despite this integration, their solution was not very accurate, landmarks –visualized as points overlaid on the live camera image– were rarely on the face, so we could not truly observe our solution working with real world data. With the ES group, they were using Python 2.7, we took about an hour to downgrade our code from Python 3.6. From there, we presume, there were no problems in integrating our code with the ES group's solutions.

6 Conclusion

From a sparse dataset we have created a classifier which uses PCA dimensionality reduction and a SVM to classify landmark points from a depth camera. An admissible accuracy was achieved through trials of trying different preprocessing methods, dimensionality reduction, and classifier models. Through collaboration with the CV group and the ES group, a pipeline was created which uses our best models to interpret a series of facial landmarks from a depth camera and feeds probabilities of emotions to a digital agent. A video of the abstracted solution is available on YouTube.

6.1 Ethical Concerns

We need to look no further than science fiction for reasons for pause about fully developing emotional systems. The 2012 film "Prometheus", features an android named David who's ulterior motives and ability to lie kill the vast majority of the human characters. On the opposite end of this spectrum, the 2013 film "Her" features a fully developed relationship between a human and a non-corporeal, artificial, digital assistant: Samantha. While David can lift boulders and Samantha is trapped in a machine, we should not discount their capability to manipulate and evoke a spectrum of emotions, from anger to love. Feelings are real and we should be careful in developing artificial systems which have the power to manipulate them.

In regards to our project, it is not out of the realm of imagination that, in a zero-sum scenario, our pipeline could be used to exploit human actors involved in the situation. By analyzing the emotion on an actors face, with a simple "scheming" module, our pipeline could deliver an expression that could trick or deceive the human. We must not make any mistake here, the machine does not feel "happy." Last winter, an AI avatar came under fire for joking about destroying humanity. In response to criticism the creator had this to say:

> "Many of the comments would be good fun if they didn't reveal the fact that many people are being deceived into thinking that this (mechanically sophisticated) animatronic puppet is intelligent. It's not. It has no feeling, no opinions, and zero understanding of what it says. It's not hurt. It's a puppet." [3]

Though the Muppets are puppets too [1], their nature is more observable, the puppet master is often nearby and we can see his or her lips moving subtly. They are transparently built and controlled for the purpose of entertainment. Concerning emotional machines, until we can reasonably establish a framework for whether a being has emotions it can reason about, we must keep the above warning in mind.

References

[1] Disney Jim Henson. The muppets. `https://muppets.disney.com/`.

[2] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. ISBN 0321321367.

[3] James Vincent. Facebook's head of ai really hates sophia the robot (and with good reason). *The Verge*, January 2018.

Appendix A Original Proposal

- Monday 23/04/2018, 16:00:
  Milestone: Deadline for submission of project specifications. Requirements: Outline of further project milestones.

- Tuesday 24/04/2018 (13:15-15:00, room 1311):
  Milestone: First feedback session on projects (individual groups). Requirements: Each member of the group has read the paper "Group-level Arousal and Valence Recognition in Static Images: Face, Body and Context."

- Friday 27/04/2018, 16:00:
  Milestone: Deadline for submission of project specifications supplementation. Requirements List milestones in a per week basis. Include details on how we plan to implement the data analysis, feature extraction, and classification tasks.

- Wednesday 02/05/2018, 17:00:
  Milestone: Analysis of dataset. Requirements Study data features and labels. Perform a simple analysis on class imbalance and distribution of values (take notes on findings). Be able to read in facial landmark positions as input from dataset. Implement at least 2 different dimensionality reduction strategies (take notes on how each of them work and what does each hyperparameter do). Explore other methods of dimensionality reduction, optionally implement them.

- Friday 04/05/2018, 17:00:
  Milestone: Implement KNN and SVM models. Requirements Implement the two classifiers demonstrated in assignment 1 part 1 and get an accuracy higher than chance probability (take notes on the implementation, how each of the models work, on how does each hyperparameter affect the result, are there documented hyperparameters which could improve the results?) Compare the two classifiers using the same methods as in assignment 1 part 1 and write notes on the obtained results.

- Tuesday 08/05/2018 (13:15-15:00, room 1311):
  Milestone: Second feedback session on projects (plenary). Requirements: Present progress of the implementation of the previous milestones in a coherent way. Be able to read in facial landmark positions as input from Computer Vision group. Classification results should be formatted such that they can be used by the Emotion Synthesis group.

- Friday 11/05/2018, 17:00:
  Milestone: Implement two more classifiers. Requirements: Implement two more classifiers different from the ones used in assignment 1 part 1 (take notes on the implementation, how each of the models work, on how does each hyperparameter affect the results). Determine which of the four classifiers implemented gives the best result and research why is that the case.

- Wednesday 16/05/2018, 17:00:
  Milestone: Write down experiment and research notes in report format. Requirements: Report: Write down report introduction. State contributions. Extend data analysis notes and write them down in the report. Extend model implementation notes (including hyperparameter experiments) and write them down in the report.

- Friday 18/05/2018, 17:00:
  Milestone: Continue work on report. Requirements: Report: Evaluate performance of each implemented model. Discuss any further improvements that are applicable to our research. Elaborate on potential ethical concerns we have identified with the emotional responsive agent. Write down conclusions.

- Monday 21/05/2018 (10:15-12:00, room 1311):
  Milestone: Third feedback session on projects (individual groups). Requirements: Record a video showing a demo of the system developed during the project.

- Wednesday 30/05/2018 (10:15-12:00, room 1311):
  Milestone: Final project presentations by students. Requirements: Presentation which showcases how we have solved the problem and the obtained results.

- Friday 01/06/2018, 16:00:
  Milestone: Deadline for submission of projects' reports and material. Requirements: Submit report (which should already be done by this point).