Chad Monteith II
Mind Reading Machines
Spring 2024

Technical Report on Supervised Learning Model for Emotion Detection

Emotion detection is one of human being's most valuable skills, as we are able to clearly and effectively determine the emotional state of another being, allowing us to react accordingly, whether we be empathetic, excited, hostile, and nearly any other niche emotions. Yet there exists difficulty in having machines understand emotions. Machines lack the ability to empathize emotion, relying on models that can distinguish between emotions in order to adjust the machine's interaction with a human. This project aims to train a model to effectively detect human emotions. With human-centered artificial intelligence becoming more prevalent, machines that can distinguish human emotion is a necessity to encourage positive interactions, rather than blank, emotionless interactions with machines. Through a model that can distinguish emotions, we can create artificial intelligence systems that can reactively adjust its speech and behavior to better serve the human it is working with, essentially bridging the gap between a computational, emotionless machine, and an emotional, organic human.

Design

The architecture of my system includes a simple Jupyter notebook, delineated between each step of the methodology, initially to process data, then train a model, and finally analyze the results. The design of this program is simple, easy to follow, and well documented in order to ensure simplicity and understanding of the code and functions.

Data

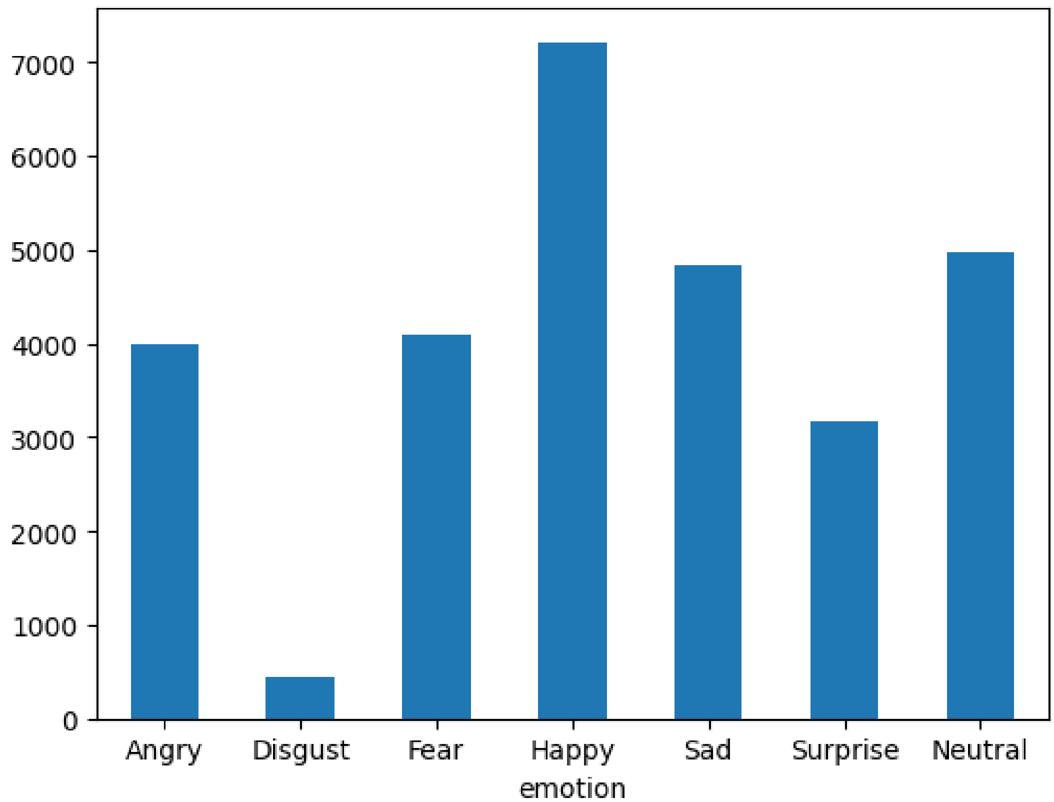
The dataset chosen is from a competition initially started in 2013, called "Challenges in Representation Learning: Facial Expression Recognition Challenge," which is linked here. This data was chosen as it provides a large dataset, of 35,887 images, to be used for this competition. This large number allows a model to be effectively trained. Within this large number, there are 28,709 labeled as testing data, and 3,589 for training and validation.

An aspect of this data that showcases proper representation of humans is the fact that this data includes facial data from a diverse set of individuals. By observing this data, we can see individuals of many different races and ethnic groups, as well as individuals with disabilities in these images. Through this, we can train the model to understand emotions of a varying range of individuals, proving its success as an equitable method of facial emotion detection.

This dataset is given in three columns, the first being the *true label*, with numbers either being $0, 1, 2, 3, 4, 5, 6$. These numbers can be attributed to emotions in the following way:

$$0 = 'Angry'$$
$$1 = 'Disgust'$$
$$2 = 'Fear'$$
$$3 = 'Happy'$$
$$4 = 'Sad'$$
$$5 = 'Surprise'$$
$$6 = 'Neutral'$$

The distribution of these emotions within the training data is exemplified in this graph:



The second column in the dataset outlines whether the particular row of data should be used for $Training$, $PublicTest$, or $PrivateTest$. I adjusted these names to be $training$, $testing$, and $validation$, respectively.

The third column is a 48×48 set of pixels which outline the image and is easily processing and representing in an image format.

I chose to preprocess my data in a method to standardize and scale my data, which saved time in algorithm efficiency and accuracy of my results. First, I standardized the pixel data by dividing each value by 255, giving smaller, more reasonable numbers to work with. Second, I scaled the data using SciKit's *StandardScaler()* function to ensure speed and efficiency when training my model.

Implementation

Python was used inside a Jupyter notebook in order to create this project, and tools and libraries such as $numpy$, $scikit - learn$, $pandas$, and $matplotlib$ were used to effectively process, train, analyze, and represent the data.

A significant piece of my implementation is the method of creating binary labels and training a model on each. With 7 different emotions, it can be difficult to create a singular model to train, and as such, I created seven different models for each emotion, predicted a binary label of *yes* or *no*. By doing this, we can effectively train a model with higher accuracy, given our set of emotions.

The implementation of my model was using a *Random Forest Classifier* for my data, as it works well to prevent overfitting of data. Because of the challenge of binary labels, overfitting is awfully easy, and by using a forest classifier, decision trees are made and is better able to prevent overfitting for the given data.

Initially, I tested a few different classifiers, those being the *Random Forest Classifier*, a *Support Vector Machine*, and a *Gradient Boosted Classifier*. Out of these, the *Random Forest Classifier* ran with the highest rate of success, and in the most reasonable time. I chose to stick with this classifier for the final product, not including other classifiers and their results in the final report.

The difficult part of this project was more the interpretation of this data, as I sought to create a set of predictions from the models and establishing the highest prediction as the predicted label of each image. By creating a list of confidence levels of each model on an image, we can determine which model found an image to be a particular emotion. From this point, I was able to determine the highest two predictions, and through this data, work to find the true emotion of the testing data.

Further implementation included plotting these results using *matplotlib* and analyzing the correctness of each model using a confusion matrix.

Lastly, I sought to better understand where my model was getting confused between emotions, and using the two highest predictions of my models, was able to find a linkage between which emotions are easily confused. Through this, we can find the most "similar" emotions, those where the model had difficulty differentiating.

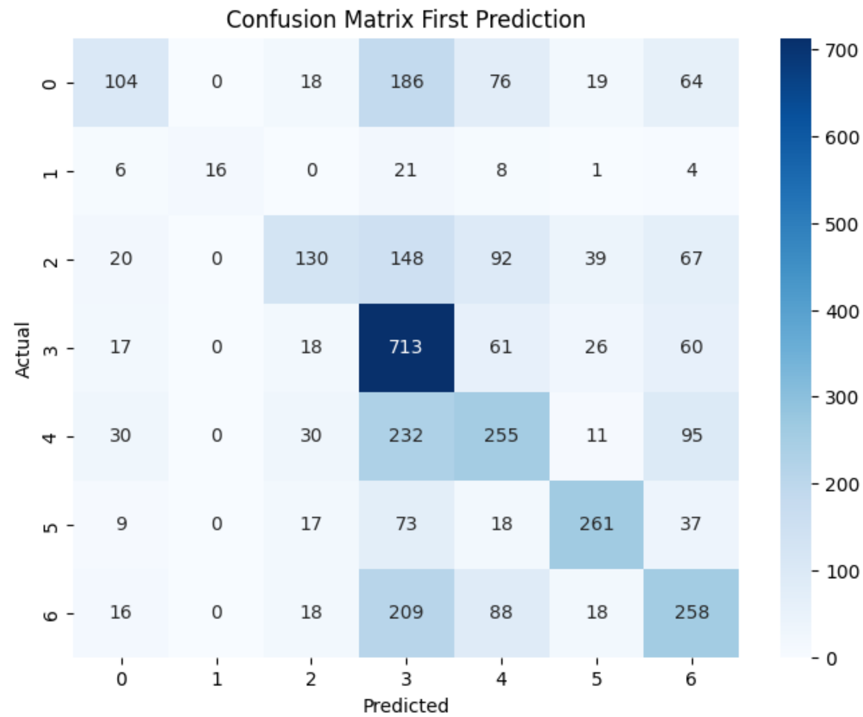
Results

The results can be broken up into many pieces, and they are attached below.

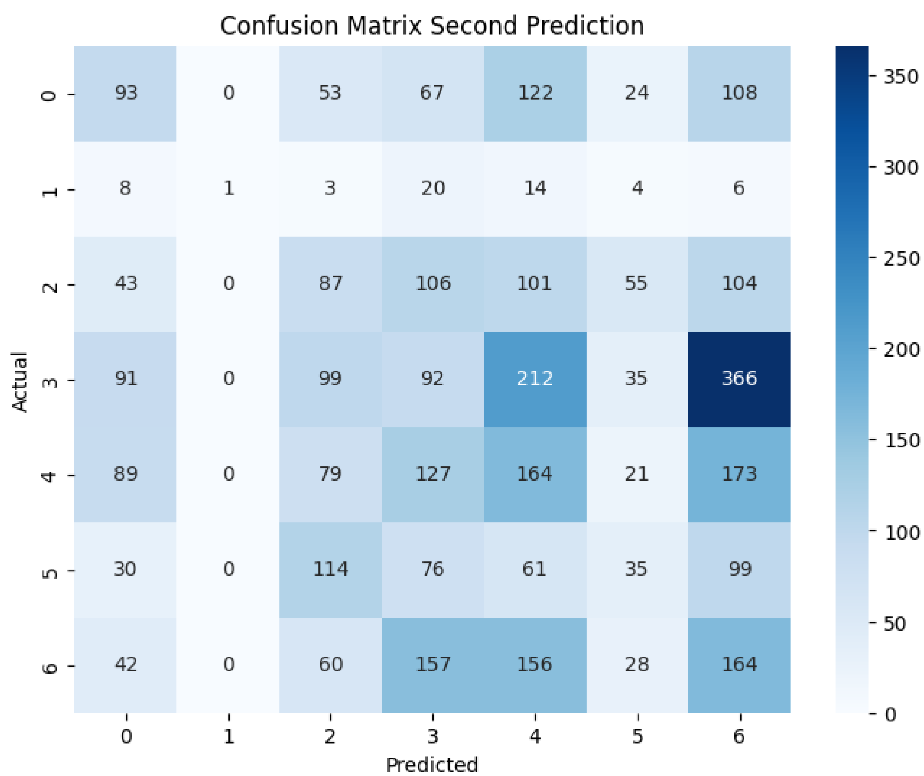
The classification report table:

	Precision	Recall	F1-score	Support
0 (Angry)	0.51	0.22	0.31	467
1 (Disgust)	1.00	0.29	0.44	56
2 (Fear)	0.56	0.26	0.36	496
3 (Happy)	0.45	0.80	0.58	895
4 (Sad)	0.43	0.39	0.41	653
5 (Surprise)	0.70	0.63	0.66	415
6 (Neutral)	0.44	0.43	0.43	607
Accuracy			0.48	3589
Macro Avg.	0.58	0.43	0.46	3589
Weighted Avg.	0.51	0.48	0.46	3589

Confusion Matrix of First (Highest) Prediction:



Confusion Matrix of Second (Second Highest) Prediction:



Example images and predictions:

Success rate of predictions:

Highest prediction:

Random Forest Classification correct classifications: 1737

Random Forest Classification incorrect classifications: 1852

Random Forest Classification rate of positives: 0.48397882418500976

Second highest prediction:

Random Forest Classification correct classifications: 636

Random Forest Classification incorrect classifications: 2953

Random Forest Classification rate of positives: 0.17720813597102256

Success rate of highest OR second highest: 0.6575647812761215

Most similar emotions between the two models:

Top 10 most similar emotions:

Emotions: Happy and Neutral, Count: 890

Emotions: Happy and Sad, Count: 635

Emotions: Sad and Neutral, Count: 389

Emotions: Fear and Happy, Count: 260

Emotions: Angry and Happy, Count: 231

Emotions: Angry and Sad, Count: 204

Emotions: Fear and Surprise, Count: 184

Emotions: Happy and Surprise, Count: 157

Emotions: Surprise and Neutral, Count: 140

Emotions: Fear and Sad, Count: 120

Performance Metrics

Through these tables and results, we can clearly see the metrics used to measure performance, including a heatmap of a confusion matrix showcasing results. A heatmap shows success by having higher numbers on the diagonal axis, and a higher number indicates higher successes. The photos attached showcase example images and predicted emotions, and one can see the difficulty in predicting particular emotions. The success rate and most similar emotions showcase how well the model did with its predictions, as well as understanding the difficulty in predicting particular emotions.

Interpretation

These results provide interesting insight into the success rate of the model. With these binary labels, we see an initial success rate of 48%. Although not a significant number, when we combine these results with the second highest prediction, we have a 65% success rate, a much more significant number. Although this is too low to effectively implement into a consumer-facing artificial intelligence machine, with some better hyperparameter tuning, we can work this success rate up to a better, more usable number.

Interestingly, the model seemed to very easily recognize a “Happy” emotion, as seen in our confusion matrix. However, we notice in the last table, that it also had trouble differentiating between “Happy” and “Neutral.”

In a similar vein, the model had significant difficulty predicting “Disgust,” which can be largely attributed to the fact that there is a much lower amount of training data for disgust, as exemplified in our emotion distribution histogram. Because of this, we are unable to achieve the correct classification for this pseudo-complex emotion, as it is largely classified as something similar.

The most anomalous data, however, is the confusion between “Happy” and “Sad.” These emotions are entirely different in a human aspect, however the model had difficulty in a few hundred instances in differentiating these. It is difficult to attribute this, as this could stem from many different scenarios. Some of these could be the quality of the image, where a side-profile image can affect results, as well as poor classification of these images due to differing facial expressions among different demographics of individuals.

However, noticing and observing that a “Neutral” emotion varies among every individual, let us combine the results of “Neutral” as also a correct observation. Through this classification, we achieve a roughly 70% success rate, a number that is shows promising results for the success of an emotion detection model.

Discussion

Overall, the performance of my model was strong, taking about 2 minutes to train all binary models on the dataset given. For the interpretation and processing my data, this was all constant time. The given results and success rate is fairly high for a simple model such as this, with little hyperparameter tuning. The model was able to successfully differentiate among differing emotions, with a degree of confidence and through a logical OR statement to classify two emotions. The model, for this scope of project and limited time given, as well as lack of skills in advanced neural networks and extensive supervised learning experience, was a strong success, as it showed ability to create, interpret, and analyze a dataset and model involving image and facial recognition, arguably a harder dataset and project than other projects done in the past.

The dataset given through the Kaggle website provides solutions submitted by other individuals. I used some of these for inspiration on how I can represent my data, such as through a heatmap, which proved quite useful in interpreting data. Many of these other solutions involve convolutional neural networks and different classifiers to adjust the data. Many involve complex preprocessing and in-depth hyperparameter tuning to achieve desired results. Existing methods are largely successful, however beyond the initial scope of this project.

Personally, these images are not distinctively clear on their labeled emotion, as doing some personal testing, I determine different emotions on particular images. Generalizing this, I believe that some of this data is difficult to accurately predict, and given our seven emotions, a few of them are fairly similar. Because of this, it is difficult to create an acceptable margin of error for our results, and a slight bias is seen to exist based on the dataset provided, a bias favoring particular emotions. This sheds light into how humans perceive and display emotions differently, and how it can be difficult as humans to classify emotions, and further, significantly harder for a machine to classify emotions effectively.

Conclusion

This project served as an effective method to analyze facial data and determine emotions given the dataset provided, and with a combined accuracy of nearly 70% given understandings and reinterpretations of the data, largely this is a success. Although the initial model gave a 48% accuracy, it served as a baseline to showcase the difficulty in classifying human emotions.

Potential improvements and future work related to this project would be some advanced hyperparameter tuning and testing different classifiers for the data. Other work includes a support vector machine to model the data, which gave a high rate of success with hyperparameter tuning. Future endeavors would be to explore these options, ideally to ensure a higher rate of success for the initial model, rather than concatenating the results with implications of the data, which served as my interpretation and conclusion.

References

The only reference included is the source data, linked above as a hyperlink.