CS301 Course Project Report

Thoa Le, Anas Hadiouche, Tejkumar Patel

***ABSTRACT***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The main objective of our project is to use deep learning algorithms to enhance facial recognition systems by adding emotion detection. Our approach includes using an existing deep learning library to detect patterns on the surface of the human face in digital images. With its built in pattern matching algorithms, our results are deemed successful with a higher accuracy rate at detecting the individual and their emotion.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Introduction**

Our project focuses on the need to include emotions to further enhance facial recognition systems and their results. Emotions play a huge role in people’s lives, and oftentimes they are expressed through images taken of the person. The way facial recognition systems work is by inputting an image and using a series of algorithms to determine who the face belongs to and whether or not they exist in a database. It is vital for the system to extract information from an image and be able to read the main expression as it would give a better understanding of who the person is. Hence, the importance of reading the emotion can lead to a more accurate result of the facial recognition system. As we summed up our project, we came to several conclusions of our results. The algorithm that we used performed significantly well at detecting the face of a digital image and overall produced a high accuracy rate. However, when we tried to determine the main emotion of the face, we achieved close to 90% accuracy. In the failed percentage of our results at determining the emotion, we realized that some of the cases are due to the fact that the face could have been portrayed as another emotion as well.

**Related Work**

There are plenty of facial recognition systems that use different algorithms to detect faces in images. The algorithms correlate to our project because they use more or less the same approach in analyzing images and using deep learning algorithms to extract certain information. The primary approach is to use a method that applies neural networks as that would be the most efficient at being able to match patterns in the image to previous existing images. When determining the race of an individual through images, the system has to go through the same deep learning algorithm and extract certain information that will be used to compare. The only difference in methods for detecting different feature spaces is the patterns that are used to compare the input image to pre existing images. For example, when a facial recognition system tries to detect the race of an individual, it has to compare the resultant patterns with patterns of images that portray different races. The patterns that we use to compare emotions are rather global and work for all individuals whereas the ones used to compare races are diverse.

**Data**

The data that we will use for our project is a data set that consists of 48x48 pixel grayscale images of faces, and it is called FER-2013. We retrieved this data set from Kaggle, an online repository that publishes data sets for anyone to use for their machine learning problems. This particular data set has 28,709 examples in its training set and 3,589 examples in its public test set. Fortunately, we did not have to augment the data in any way as each image was already adjusted in a way that the algorithm would run smoothly without issues such as imbalance, contrast variation, or intra-class differences. Since it is a significantly large dataset, we cut down our experimental dataset down to five to ten images per emotion.

**Methods**

As discussed earlier, the issue at hand is the need to apply emotion detection in facial recognition systems. The approach that we will use is a pre-existing method that has been created by a research group of Facebook. It is called DeepFace, and it is a deep learning architecture that can detect faces in digital images. This architecture consists of a nine layered neural network that has 120 million connection weights. The layers in a neural network correspond to each other and result in an output layer that will ultimately match with previously existing patterns. We wanted to use this method because of its incredibly high accuracy at detecting faces in digital images. Another reason why we went with this method is because of its ease of use. By importing the library and applying the image path as input, the method does the rest. It uses a two step approach of aligning the image in 2D using six specific points of the face and then converting to a 3D alignment. As shown in Figure 1, each step of the process of DeepFace is portrayed. The resultant image is used as input in the neural network and progressively compares its patterns to millions of different nodes. There are other existing methods that we could have used as alternatives but this is the one that we found to work for us.

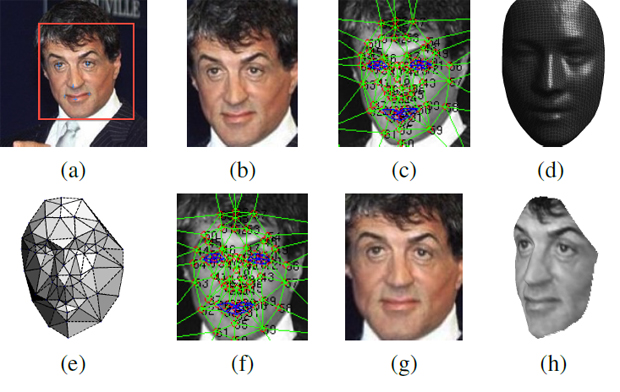


Figure 1. Steps of DeepFace, a deep learning algorithm created by one of Facebook’s research groups

**Experiments**

Our experiments consisted of us extracting images from the FER 2013 dataset and testing them with our script. In addition to the images provided in the FER 2013 dataset, we decided to add an additional hyperparameter by including several images of people with masks. Even though we want our system to be as accurate as possible in reading emotion off of a person’s face, it is also important to include some hyperparameters to further test the efficiency of our script.



Figure 2.. Neutral Expression

Some insight into the failed hits of our model is where the image could be interpreted as another emotion. The image in Figure 2 was in the FER 2013 dataset as a “sad” face, but could easily be interpreted as neutral due to certain patterns. If the face truly held sad emotion, the eyebrows would be scrunched up, the lips would hold the opposite of a smile, the eyes would be closed or angle downwards, etc. However, in this instance, the face can quite easily be determined by the system as a neutral looking face.

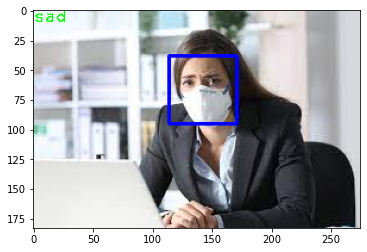
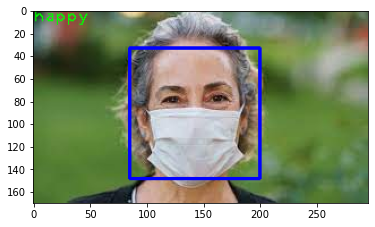
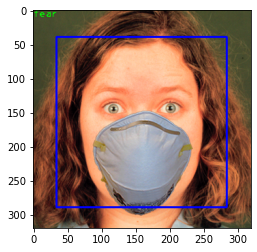
\

Figure 3. Hyperparameter of Faces with Masks

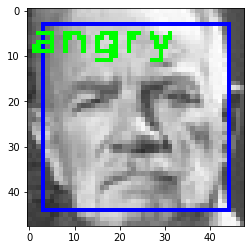
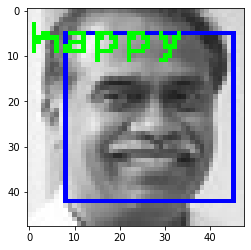
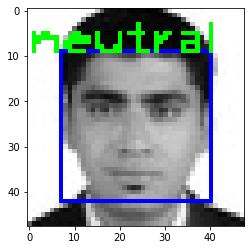
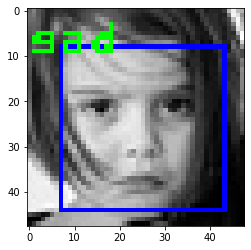
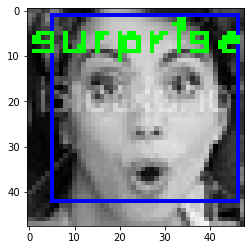
The three images in Figure 3 are the results of our experiment using the hyperparameter of faces with masks. This further highlights our script's ability to use past datasets to “train” the model that will read the image’s emotional tells even though the mask hides the mouth, which can hold a lot of value in dictating the emotion. The shape of mouths, that do and don’t show teeth, can easily distinguish between happiness and sadness. Each image uses the eyes and eyebrows to identify the emotion from the model that we trained earlier. Using the model to accentuate those facial features provides the script with making better reads of the emotion. 

Figure 4. Results of our method

The DeepFace framework uses a neural network in its deep learning algorithm and uses patterns from four million trained images. The neural network is made up of several layers, the input, hidden, and output layers. Each layer has nodes that contain information of patterns to pass on to the next layer. The process in training this system is to use complex images that contain body parts, hairstyles, clothes, and any additions to improve the accuracy of the neural networks. For the neural network to be effective, they must be applied in order of:

* Detection - finding the features that represent a face
* Alignment - positions the face forward to grasp the patterns
* Representation - relays the image/face found
* Classification - detects the main emotion of the face

Cascade Classifiers use kernels and features, which are calculated with kernels. Each feature will calculate the sum of the white and black pixels which in turn will classify a part of the face. If the feature is looking at the eyes and nose, it may classify the eyes as dark pixels and the nose as white. In essence, the darker parts of the face will have the feature see it as black pixels and lighter parts as white pixels respectively. Once all the features are put together and “complete the puzzle,” the face will be used in any way that it is needed.



Figure 5. Emotion recognition code using our dataset

**Conclusion**

To conclude on our project, we were able to grasp a better understanding on how machine learning can be applied to real world problems. As we have discussed, our problem consisted of obtaining emotion detection to enhance facial recognition systems, and we were able to achieve that to a certain degree. Although our only method was to use neural networks to find patterns in images and collect the output, we believe that a combination of other methods would further improve the accuracy of our results. Another extension that could have worked in combination with the neural network would be to use emotion intensity maps. Our accuracy in detecting emotion through the surface of an image of a human face was high but it cannot fully grasp the intense abstractness of emotion. With emotion intensity maps, it can highlight discriminative emotional regions and conclude on a final recognition stage.