CSCE 689 Final Project Writeup

Jacob Fenger
Texas A&M University
Jacobfenger@tamu.edu

Jacob Hallenberger
Texas A&M University
Jhallenberger@tamu.edu

I. Introduction

For this project, we trained a Convolutional Neural Network (CNN) to classify the emotions expressed in pictures of faces. We then created a website where users can upload photos of people and our model will output the predicted class. Photos that users upload will then be used to update our model so it gets more accurate as users upload more photos.

II. Literature Review

Emotion recognition using facial expressions has been studied many times in the past. A recent paper regarding this topic is a paper titled *'Emotion Recognition Using Facial Expressions'* by Tarnowski et. al. In the paper, they tried to classify seven different emotional states based on facial expressions. They explored two different methods for classification: a three nearest neighbor classifier and a two-layer neural network with 7 neurons in the hidden layer.

They explored several different results including subject-dependent classification and subject-independent classification accuracies. In my opinion, the subject-independent results are more important as it can show a more general accuracy when it comes to emotion recognition. The nearest neighbor classifier had an average accuracy of 63% while the neural network had an accuracy of 73%.

One interesting tidbit mentioned in the discussion section of this paper was the exploration of face visibility and how it affected emotion recognition. As one would expect, the classification accuracy of the neural network decreased by 20methods must be used to increase emotion recognition accuracy when individuals faces are only partially viewable.

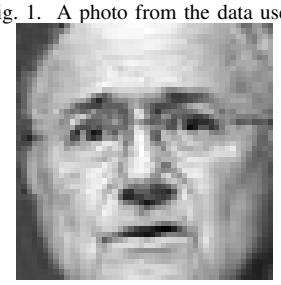
III. Problem Formulation

With the increasing popular of photo and video use everywhere in the world today, emotion recognition of facial expressions could be used to enhance or analyze certain aspects of life. Additionally, the integration of a model into an application is something that is not covered too much that my team wanted to learn more about.

IV. Proposed Solution

To classify emotions of facial expressions, we used a basic Convolutional Neural Network (CNN) as our classifier. This model is then saved for use in the application. Flask, a Python web-framework, is used to host the model and user upload capabilities for classification. Keras is used to create the CNN as well as adapt the model when new user data is added.

Fig. 1. A photo from the data used.



To deal with the user uploaded photos, some preprocessing is necessary before we can classify the image. We must grayscale the image, perform face recognition to isolate the face, and then resize the image before classification. This is all done very easily with OpenCV in Python.

V. Data Description

The data used for training and testing the classifier is from a Kaggle competition: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

This data set consists of 35887 images of faces (48x48 pixel grayscale) that are already preprocessed. The classes for the faces are: angry, disgust, fear, happy, sad, surprise, and neutral. Figure 1 shows an example of what is contained within the dataset.

VI. Results

Our CNN achieved an accuracy of 64% with the test set. Results could be improved by adding more data to our training set. Each emotion was also not represented evenly with only a few samples actually being labeled as disgust. One problem with some of the data that was used to train the CNN was that some photos had people's faces covered up with may affect certain results. One way to deal with this images is to remove them from the training set, or to try and acquire more data that could help with the accuracy of these types of images.

VII. Conclusions

The main take away from our project is that it is not too hard to add a model to an existing web application. Keras and other libraries make it very easy to save models and also update them. Future work would be to improve classifier performance as well as make a better looking website for the user. Better model adaptation techniques could be looked at to improve performance as the user base grows.