Facial Recognition Using Deep Learning

Abigail Carbaugh

Elizabethtown College

carbaugha@etown.edu

**ABSTRACT**

One difficulty for people with Autism is recognizing and understanding facial expressions that others are portraying to them. Facial expressions through conversations are how someone gets their mood across to another, and without this, it is very difficult for someone to understand how another is feeling. I am basing this project on a dataset that has various photos of people with different expressions. I also plan on trying to use created images of myself and others in the project after training to see if it can accurately conclude the expression being portrayed. I would like to achieve an accuracy of 80% or higher. The plan for this project is to use neural networks in order to create this high scoring accuracy.

# INTRODUCTION

Understanding facial expressions is a vital human survival tool in today’s world. Facial expressions display one’s personal emotions and can help indicate one’s intentions in a social situation. Studies have previously shown that “…individuals rarely interact with context-less faces [1],” so the act of understanding these expressions is important in these face-to-face interactions. Neurodivergent people, however, have a more difficult time recognizing these expressions at higher intensities, making them unable to distinguish what a person is trying to convey. This model, along with other technological advances, would be able to be integrated into things like smart glasses in order to help an individual detect these emotions that cannot do so on their own.

Previous attempts to tackle this problem have been done in the past, but they only use photos from the dataset being used, and not applying the application on other photos. Previous works have also only used three basic emotions (happy, surprise and neutral) that the model has to recognize. This does not allow the model to be used universally. If the model can only classify three expressions, there is limited applications to the project.

In order to overcome these previous work’s shortcomings, I plan to, along with the photos from the dataset, use personal photos of me and others in order to see if this model can be universally used across photos, not just from the dataset. I also plan on using more than three emotions like previous works did. I plan on using seven basic expressions (anger contempt, disgust, fear, happiness, sadness and surprise) in my model in order to classify the photos instead of limited expressions in order to make it more applicable.

In creating this project, I hope to use neural networks in order to produce a model with high accuracy rate in recognizing facial expressions. I also hope to make it more applicable to real world scenarios by using a wider range of detectable facial expressions. In order to test is applicability and real-world use, I plan on using personal photos in addition to the dataset.

The dataset for this project originates from a Kaggle facial recognition challenge where contestants tried to create a model with high accuracy that could predict a facial expression an image is trying to convey. The dataset consists of 48x48 pixel greyscale images of different faces. This dataset, unlike others found, does have the seven categories of facial expressions I would like to use in my model, eliminating the need to create my own new categories. The data is presplit into training and testing datasets, with training containing the columns pixels and emotion and testing only containing pixels. The outcome of this challenge was to accurately predict the emotion column for the testing data.

To create this neural network model, the Keras package will be utilized. This package is a high-level API for building neural networks [2]. It also allows us to create custom layers and models so that one can set up their own transformations and weights to a layer of the model without many restrictions. To evaluate performance of the model, the accuracy on the training and test sets will be primarily used. It is expected that the model will achieve at least a 60% accuracy as the top ten contestants from the original Kaggle challenge received that percentage or higher. My long-term goal for this project is to have my model perform better in accuracy than the winner of the challenge, who had a 71.2% accuracy score. Once all models have been created and evaluated, I plan to make a simple application that can take a user’s photo and once uploaded, will predict the facial expression.

# BACKGROUND

There are seven expressions that psychologists have recognized to be universal expressions. These expressions are anger, contempt, disgust, fear, happiness, sadness and surprise. Each of these emotions, signaled by facial expressions, “…turn on different cognitive and physiological reactions in order to prime the body for action [3].” For neurotypical people, one without atypical neurological patterns, recognizing these facial expressions is done by the brain within 200 milliseconds [4]. For neurodivergent people, however, may not be able to recognize an expression that quickly. Within the range of emotions, there are also intensities of that emotion. With these different intensities of facial expressions, people with autism spectrum disorder (ASD) may not be able to understand the emotion one is portraying if it is not high intensity. There have been many studies done on people with ASD regarding facial emotion recognition which have found that those with ASD will perform normally with recognizing these expressions with high intensity. With low intensity, however, they were not as easily recognizable [5].

Psychologists have done studies on people with ASD to try to help individuals recognize emotions, but many have fallen short as autism is a spectrum. Everybody diagnosed with ASD is different as some may exhibit many symptoms while some have very few, so teaching others to pick up on expressions may be more easy or difficult. Zhenjie Song, author of *Facial Expression Emotion and Recognition Model Integrating Philosophy and Machine Learning Theory* [6], suggests that machine learning models, specifically convoluted neural networks (CNNs) may be the answer to helping. She found that CNNs can efficiently and accurately recognize facial expressions. Song had found that finding datasets for this problem was very difficult as many were very small. She proposed using image augmentation in order to extend the dataset used. Song had been able to achieve an accuracy of 74%. Song did not say how she classified the images, however, so we do not know how many categories the images could be classified as. Song also went into emotion recognition through gestures and speech, but since facial expressions are one of the most important things with face-to-face interactions, I chose to only focus on that aspect. Song also was not able to achieve the accuracy of 80% or higher, like this project intends to do.

Xiaofeng Lu, author of *Deep Learning Based Emotion Recognition and Visualization of Figural Representation* [7], also tried to solve this problem using CNNs. Lu had been able to achieve a much higher accuracy score or 98.75%, much higher than Song. Lu proposed a convolutional neural network-Bi-directional Long Short-Term Memory algorithm (CNN-BiLSTM) in order to achieve this high accuracy score. Lu found this algorithm not only provides great results but provided significant “…experimental reference for research on learners; emotion recognition and graphic visualization of expressions in an intelligent learning environment.” Lu used the Keras package, as this project plans to use, in order to create the architecture for this project. Instead of using a found dataset, Lu used image and audio data of learners in a class as they answered questions. Upon evaluating the model, Lu found that the function loss of the CNN-BiLSTM model decreases to 1.33% after 100 training periods, which reduces influence of function loss on the model. By using this real-time emotion recognition, Lu was able to create a model having an accuracy score of no less than 90%, and plans on optimizing the model further, as of this paper’s release. Lu had been able to achieve much higher results with their model, but the model also focused on using real-time data and images.

By using these two papers as guides, I plan on using a CNN model with my own parameters in hopes of achieving a high accuracy score. With the initial goal of the model having an accuracy score of 80% or higher, I plan on using the Keras package to be able to make my own layers and models.

# DESIGN

As previously stated, this project will be using the Keras package through TensorFlow with Python. This project is done mainly on a Google Colab notebook in order to utilize the GPU function for runtime, making it faster and more efficient to run.

Figure : Image before and after the ViT model converts it into patches.

Diagram

Description automatically generatedFor the model that will be the base model, a combination of convolutional 2D and Max Pooling 2D layers will be used for the first four layers. The model will then have a dropout layer added, and then the model will be flattened. A dense layer is then planned to be added with the activation function being ‘relu.’ Another dropout layer will be added, and then a final dense layer with activation function ‘softmax’ will be implemented.

Figure : Basic architecture of a CNN model [8].

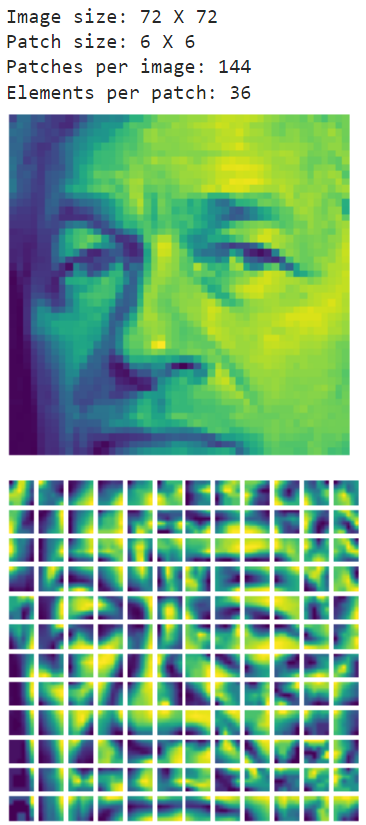
Other models will also be implemented in order to see which model architecture will perform better. Along with the base model, a Visual Geometry Group with 16 layers depth (VGG16), a Residual Network (ResNet), and a Vision Image Transformer (ViT) models will be used.

Figure 3: Count plot of the number of each emotion in the training dataset.

A VGG16 model is a simple convolutional neural network architecture. This model was based on an analysis of how to increase the depths of the networks. A VGG model consists of blocks, and each block is made up of 2D Convolutional and Max Pooling layers [9]. By using Keras, a VGG16 model can be imported. By importing the model, all that was needed to do is add layers that will allow the model to take in the dataset used for this project.

The ResNet model was also imported by using Keras. The ResNet architecture uses shortcut connections to solve the vanishing gradient problem [10]. Like with the VGG16 model, once imported, more layers were added to allow the dataset to work with the model.

The ViT model applies the Transformer, another model, architecture with attention to sequences of image patches and does not use convolutional layers. This model was also imported from the Keras package but works differently from the VGG16 or ResNet models. A ViT model uses a multilayer perceptron that has both a dense and dropout layer. From here, a patch maker converts images into patches and an encoder will perform linear transformation on image patches. Lastly, it defines a classifier with the vision transformer with the encoded patched acting as the input. No additional layers were needed to be added to this model.



A picture containing text

Description automatically generated

Chart, bar chart

Description automatically generatedThe optimizer being chosen for all models will be Adam, and when compiling the model categorical cross-entropy will be used. Dimensionality reduction will also be implemented in the design after the initial model to see how the accuracy compares to the original model.

# EXPERIMENTS

In data preprocessing it was found that there was three datasets: training, testing, and one labeled ICML. The ICML dataset was used for the final testing for determining the winner of the original competition. The training dataset was the dataset used for both training and testing as the testing dataset provided did not have a designated emotion column to validate if the model was correct.

The first thing done in order to make the data more uniform was taking the training dataset at the “pixels” columns (which is our X\_train) and reshape the array to 48x48 and change the pixels column type to float32 because the original type was object. Once this was done, the array was stacked. This allowed me to have a sequence of arrays along a new axis. A count plot was created in order to find the number of counts per expression in the training dataset.

It is shown from the count plot that there are less images labeled disgust than any other one. It was considered being dropped, but I would like to see how that column can ultimately affect the outcome and accuracy of the model. As done with Song’s experiments, image augmentation must be conducted.

Chart, bar chart

Description automatically generatedThe first 10 images of the training dataset were then plotted using the X\_train data, showing the number in the array the image is, and what emotion is being depicted according to the y\_train data. The data taken in for the image has been reshaped to 48x48 with the colormap set to gray to maintain continuity throughout the photos.

A picture containing graphical user interface

Description automatically generatedTo balance the dataset, images with the label of ‘disgust’ were taken and placed into a separate dataset. From here, the dataset was copied until it reached around the same number of images as the others in the training dataset. The disgust dataset was then added back to the original training dataset. The dataset was then taken through image augmentation using the ImageDataGenerator. The ImageDataGenerator takes the original photo inputs and transforms them randomly according to parameters specified.

Figure 5: Each model and their corresponding accuracy rates.

The next step is to create a sequential CNN model that generates certain model information. As previously stated, the model will contain of a mixture of convolutional 2D and max pooling 2D layers will be used for the first layers. Then, a mix of dropout, flattening and dense layers will be added with the proper activation functions with the optimizer chosen being Adam with categorical cross-entropy as the loss function. Once initial modeling was conducted, it was found that the base model only produced an accuracy of 21%. Each model, after experiments, used a portion of the training and testing data from the train\_test\_split as a validation dataset and the model was saved for later application testing. When conducting the VGG16 model experiment, it was found that it could produce an 85% accuracy rate. This was exciting as it was much higher than the winner of the competition’s accuracy score. The ResNet model was found to have a 55% accuracy rate, not the best but still conducted better than the base model.

Dimensionality reduction will also be taken into consideration with this project as it will reduce the number of random variables in order to obtain a set of principle variables for the dataset. This will give the model less complexity, shorten computation times for the model, as well as helps prevent overfitting and collinearity of the data. Dimensionality reduction will hopefully, in the end, improve the learning feature accuracy as well as reduce the training time that the model needs. This will be implemented in the ViT model.

The ViT model was the last experiment conducted as it was the most intricate model. With this model there are custom layers that do the image augmentation and patch building/encoding within the model instead of doing everything separately. Once this model was finished running, it was found to have an 89% accuracy rate, the best out of all the models.

Lastly, it is planned to take personal photos and add them into the dataset in order to test the application of the model. The goal is to be able to add original photos of expressions to see if the model can also accurately predict images not in the dataset at all. This is a large aspect of the project as it would like to be featured in the final demonstration of the project. This was done within the application made. To test the validity of the model, a dataset of the images used was made to run through the model. This was done because it is easier to check the accuracy due to the model already having the label. It also will have the same resolution, allowing for continuity within the images and predictions.

Figure 4: Image after augmentation using the Keras ImageDataGenerator.

An application was then made using Streamlit which is a free and open-source platform for people to demonstrate their models. By uploading the models saved into the application notebook, we are able to call the model for the application to predict the image uploaded.

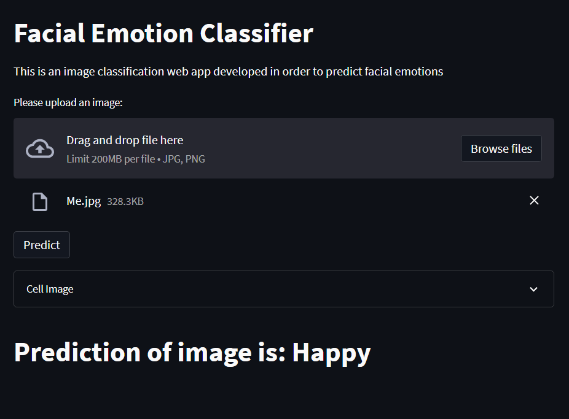


Figure 6: Basic layout of the facial emotion classifier made with Streamlit.

# TIMELINE

With the limited timeline to finish the project, all work has been planned out accordingly in order to ensure the work will be finished in time for the final presentation.

1/23/23 – Start proposal

1/30/32 – Data collection and preprocessing

2/6/23 – Exploratory Data Analysis

2/13/23 – Model Design

2/20/23 – Feature Extraction

2/27/23 – Model Implementation

3/13/23 – Model Implementation and Training

3/20/23 – Parameter Tuning

3/27/23 – Application Design and Use Cases

4/3/23 – Model Evaluation and Comparison

4/10/23 – Code Final Revision

4/17/23 – Final Paper Writeup

# REFERENCES

1. Qiang Xu, Yaping Yang, Qun Tan, and Lin Zhang. 2017. Facial expressions in context: Electrophysiological correlates of the emotional congruency of facial expressions and background scenes. (December 2017). Retrieved February 6, 2023 from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5733078/
2. Keras Team. Simple. flexible. powerful. Retrieved February 6, 2023 from https://keras.io/
3. David Matsumoto. Benefits of Reading Facial Expressions of Emotion. Retrieved February 6, 2023 from https://www.humintell.com/2021/04/benefits-of-reading-facial-expressions-of-emotion/
4. David Matsumoto. Benefits of Reading Facial Expressions of Emotion. Retrieved February 6, 2023 from https://www.humintell.com/2021/04/benefits-of-reading-facial-expressions-of-emotion/
5. Neuroscience News. 2021. Why people with autism read facial expressions differently. (August 2021). Retrieved February 6, 2023 from https://neurosciencenews.com/aans-asd-facial-expression-19065/
6. Zhenjie Song. 2021. Facial expression emotion recognition model integrating philosophy and machine learning theory. (September 2021). Retrieved February 6, 2023 from https://www.frontiersin.org/articles/10.3389/fpsyg.2021.759485/full
7. Xiaofeng Lu. 2021. Deep Learning based emotion recognition and visualization of Figural Representation. (December 2021). Retrieved February 6, 2023 from https://www.frontiersin.org/articles/10.3389/fpsyg.2021.818833/full
8. M.K. Gurucharan. 2022. Basic CNN architecture: Explaining 5 layers of Convolutional Neural Network. (October 2022). Retrieved February 6, 2023 from https://www.upgrad.com/blog/basic-cnn-architecture/

[9] Anwar, A. (2022) *Difference between alexnet, vggnet, ResNet and inception*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96 (Accessed: April 24, 2023).

[10] Boesch, G. (2023) *VGG very deep convolutional networks (vggnet) - what you need to know*, *viso.ai*. Available at: https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/#:~:text=VGG%20stands%20for%20Visual%20Geometry%20Group%20and%20consists%20of%20blocks,more%20complex%20functions%20also%20increases. (Accessed: April 24, 2023).