Milestone Project: Facial Emotion Detection

Yonghoon Yoon

Problem Definition

Deep learning can be used to process images and audio to recognize the meaning behind objects within those forms of media. In this case, pictures of faces can be processed to recognize the emotion that is exhibited. This type of computer vision is among many efforts to have artificial intelligence mimic human abilities of perception, and eventually reaction.

This type of computer vision where the machine identifies facial emotions can have wide applications and uses, particularly where there are customer interactions. For example, facial emotion detection can be used to record feedback during customers' meals at restaurants. In other examples, companies like Intectics use this technology to gain customer feedback on sports fans' reactions to games and apply it to sportswear sales.

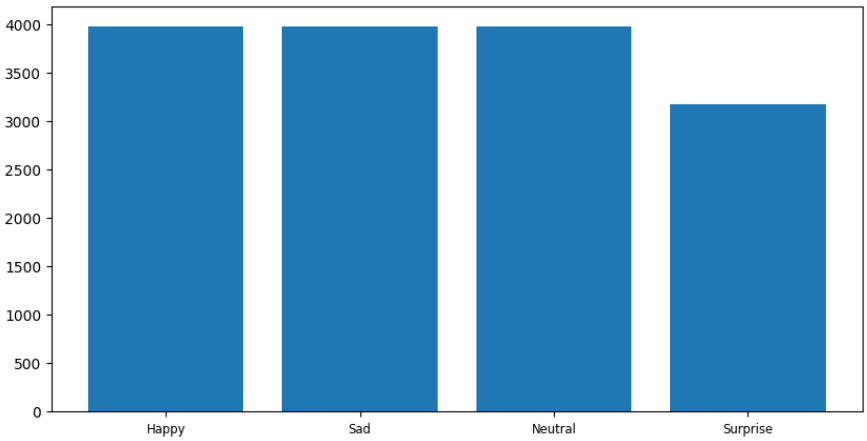
The goal of this project is to use Deep Learning and Artificial Intelligence techniques to create a computer vision model that can accurately detect facial emotions. The model should be able to perform multi-class classification on images of facial expressions, to classify the expressions according to the associated emotion.

We are using data science to build models from scratch or use transfer learning to add onto what's been already trained so we can have them train to correctly identify the emotions displayed in the images.

Data Exploration

The data set in a zip file consists of 3 folders, i.e., 'test', 'train', and 'validation'.
Each of these folders has four subfolders:

- **'happy':** Images of people who have happy facial expressions. (3976 images)
- **'sad':** Images of people with sad or upset facial expressions. (3982 images)
- **'surprise':** Images of people who have shocked or surprised facial expressions. (3978 images)
- **'neutral'**: Images of people showing no prominent emotion in their facial expressions at all. (3173 images)



The data is somewhat imbalanced, but unless there is a significant difference in the models' ability to detect surprise images, it does not look too concerning.

The data is split into:

- Train set: 15109 images
- Validation set: 4977 images
- Test set: 128 images

There is a potential problem with this imbalance since the test set is very small compared to the train and validation set. Therefore, pre-processing a different split where there is a larger set of test data because

In terms of visual distinctions, some patterns emerge within each class.

- For the category of 'happy', there are usually concave up mouths with teeth showing. Eyes tend to be smaller due to smiling.
- For 'sad', tears are noticeable in some, along with closed and concave down mouths. Eyes tend to be a bit bigger than 'happy' images.
- For 'neutral', there are a lot of flat months with 'regular' shaped eyes.
- For 'surprise', eyes are big and often there are hands on peoples' faces.

Upon viewing some of the images, images of non-human characters as well as random images of lines are also present in the data. The amount does not seem significantly big.

Building Various Models

For each model, data loaders were used, then compiled to process fitting. Then they were evaluated on the test data.

The following models were used to predict facial emotion, and their bahaviors:

Model 1 (CNN)

- Accuracy:

Model 2 (CNN)

- Key difference from Model 1: no dropout

Model 3 (Transfer: VGG16)

Model 4 (Transfer: Resnet)

Model 5 (Transfer: EfficientNet)

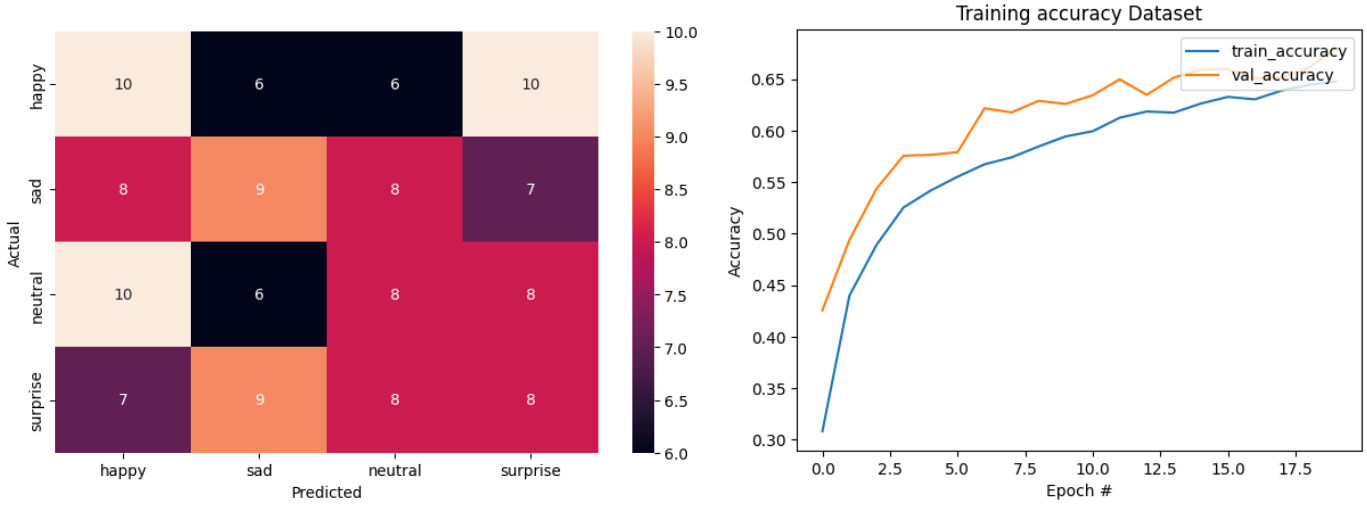
Model 6 (Complex CNN Architecture)

- The Complex CNN Architecture was created to use the fully connected block and additional layers of convolutional layers for better prediction. The model unfortunately did not perform as well as was hoped.

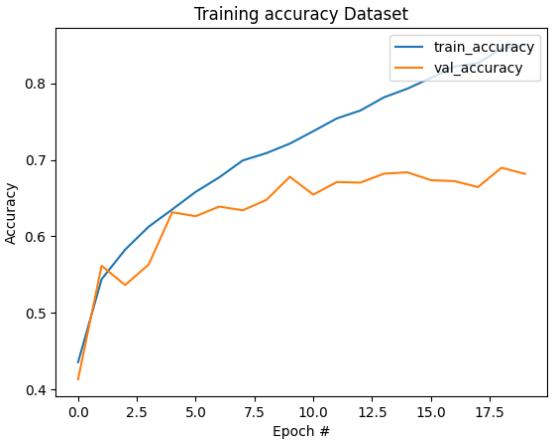
The transfer learning models were helpful in that they are already pretrained, but more exploration will be done to increase accuracy.

Comparison of Various Techniques and Their Relative Performance

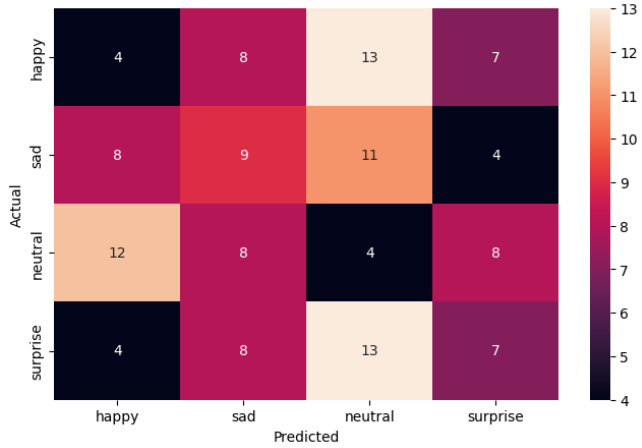
Model 1 (CNN)

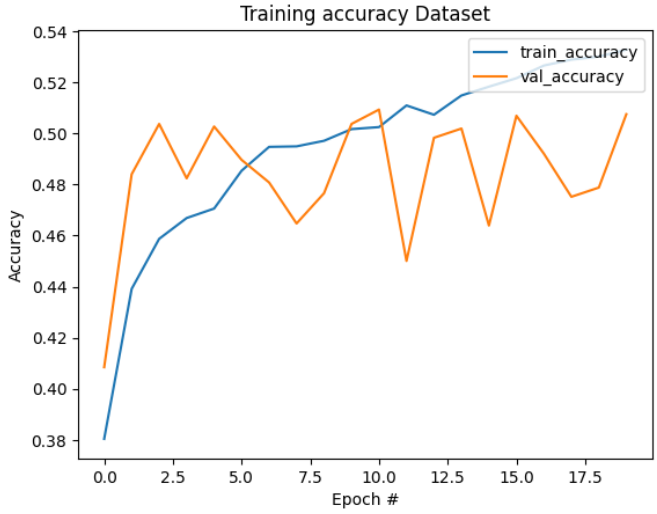


Model 2 (CNN)

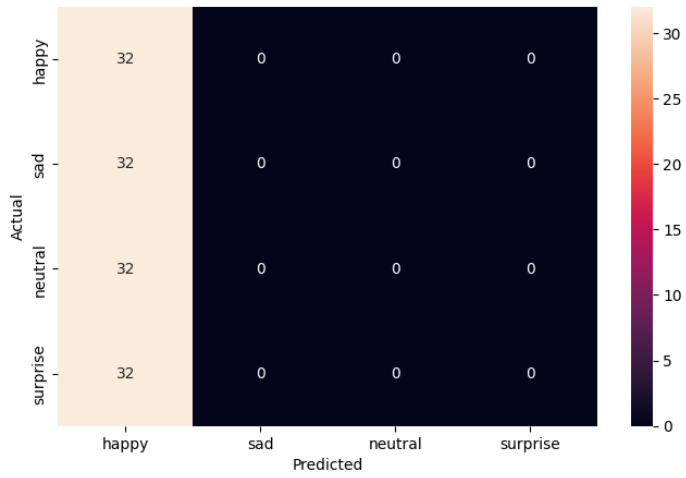


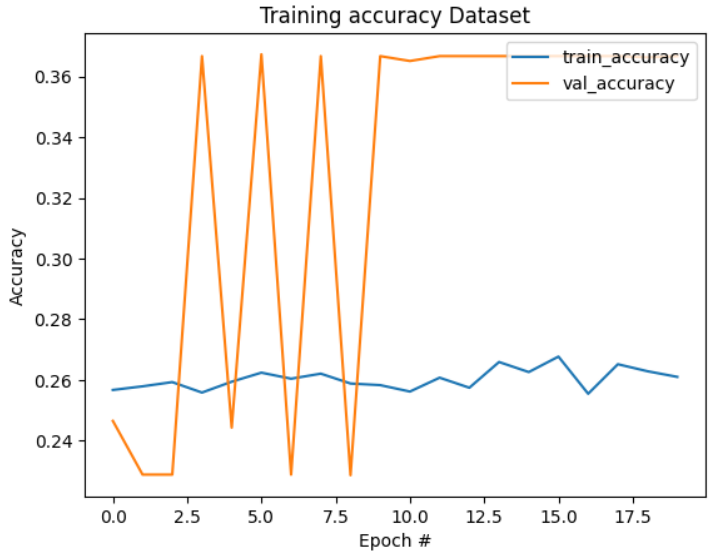
Model 3 (Transfer: VGG16)



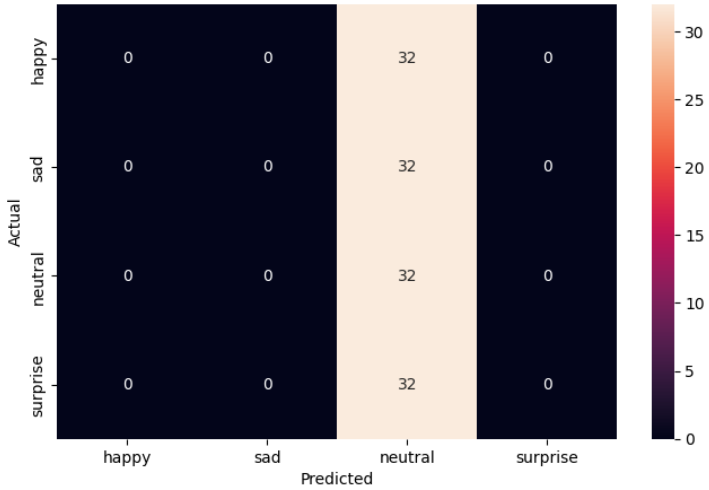


Model 4 (Transfer: Resnet)

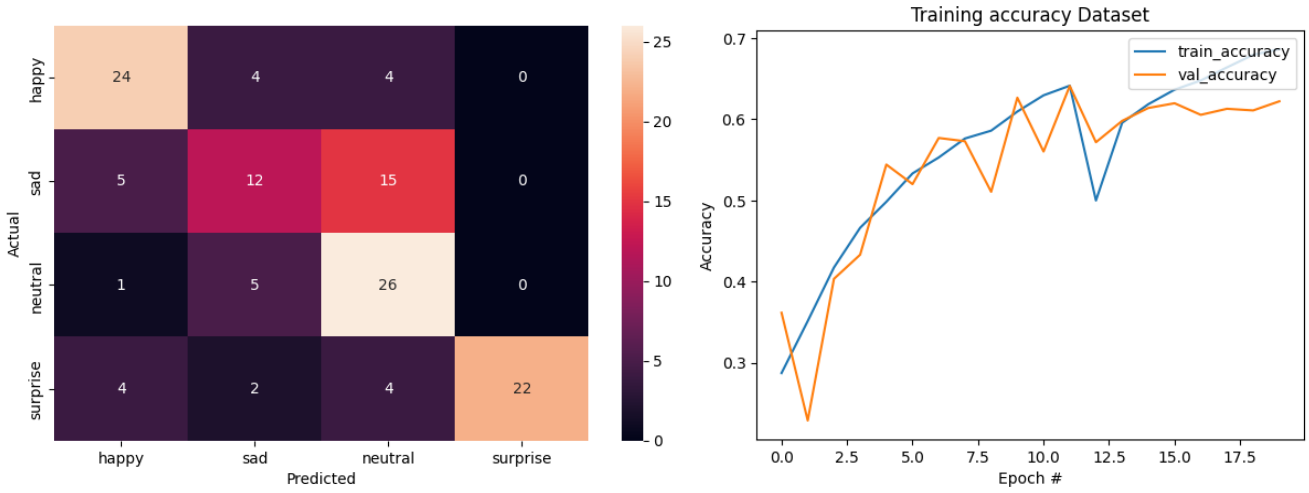




Model 5 (Transfer: EfficientNet)



Model 6 (Complex CNN Architecture)



Proposal for the Final Solution Design

I chose the final architecture because it involves more fine-tuning and as it shows on the history plot, it performs the best with the least overfitting.

To make it better, I will be working with hyperparameter tuning more, and reduce ovefitting further. In addition, I will be looking into the color scale differences. Lastly, it would be important to look at the time cost so that the models can be the most effective in cost and performance. Mention epochs for price.