Machine Learning II Project Report:

Facial Emotion Image Classification

By Saivee Phatak & Sunishka Sil

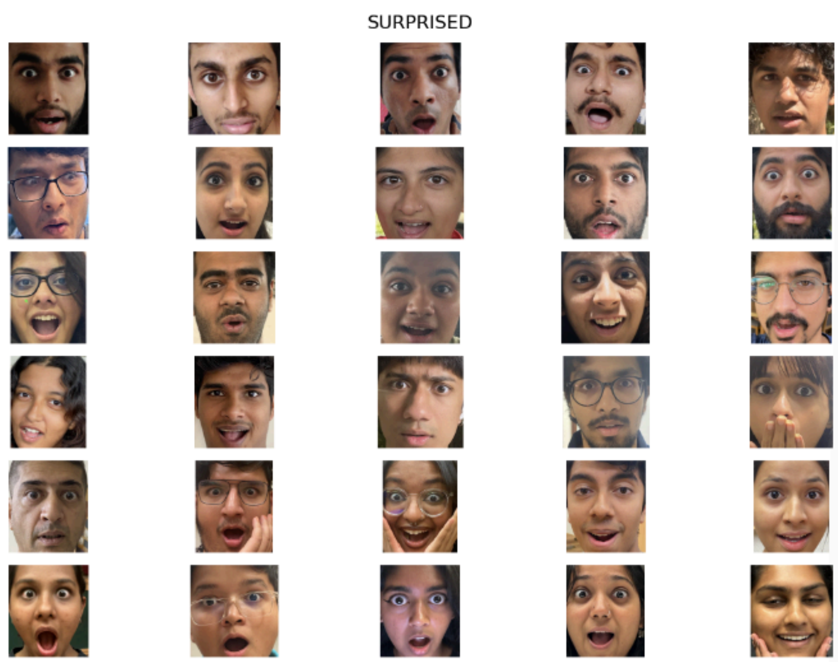
Project Introduction

We decided to pick facial expressions of emotions as our focus for this project. The classification of exhibited emotions has significant implications for the domains of psychotherapy, student well-being, healthcare, communication and marketing. We were particularly keen on testing canny edge detection followed by VGG16, ResNet, GoogleNet and an attempt to test our own model on an unstandardized set of face images.

Dataset

We clicked front-facing, close-up images of 30 college students exhibiting five different emotional expressions. The emotions we picked for our data include (i) Neutral state, (ii) Happiness, (iii) Sadness, (iv) Anger, and (v) Surprise. Every student that agreed to be a part of our data collection was asked to pose in all five emotions. In totality, our dataset consisted of 150 unedited and colorized images. With the nature of our dataset and the problem statement we have, our aim was to achieve a decent multi-classification accuracy for five different emotions with our collected data.

Here is a sample of our data:



Link to Original Images:
https://drive.google.com/drive/folders/1jftwtQwVFijMK9l0DodOfDylZ63g3tPk
Link to Augmented and Original Images folder:
https://drive.google.com/drive/folders/11iWV--APT9bYbadv7BP-4gLQpCIJDexu

Pre-Processing

- ***Storing and Loading:*** We stored all our images in a Google Drive directory with 5 sub-folders for our 5 emotion classes. From our directory, we loaded our images into a key-value dictionary in Google Colab, with every image path and its class label of emotion. Using matplotlib, we plotted all our images as sub-plots under every emotion label to ensure our data points were consistent with our directory
- ***Formatting:*** We converted all our RGBA images into RGB format and all the images in our dictionary to JPEG images and stored them in a different output destination path with emotion labeled folders.
- ***Normalization:*** We normalized our images, dividing them by 255, to adjust the pixel values to a range of [0,1] and counter the high-resolution nature of the images we clicked.
- ***Augmentation:*** We resized our images to standard 244x244 squares, and created their high exposure and low brightness version copies.
- ***Grayscale + Canny Detection***: We created a loop to iterate through the images, plot their grayscale versions and perform canny edge detection on the grayscale images to produce the detected edges next to them. We lowered the threshold from the usual 100-200 to 50-150.

Model Testing & Comparison

80% of the dataset was split into train files and 20% was used in test files. These were stored in separate train and test destination folders, once again with their own emotion class-labeled sub-folders.

Model Attempted	Parameter Tuning	Best Possible Accuracy%
VGG16 #1	Epochs = 8, steps per epoch = 5	57%
ResNet #1	Epochs = 6	33%
GoogleNet #1	Epochs = 14	68%
Our Own CNN Model #1	Epochs - 10	80%

For each model, we started with a lower number of epochs and worked our way up; we recorded the parameters at which the model performed its best with the dataset. Out of all the models we tried, ResNet had the poorest performance while our own modified CNN model had the best performance.

- VGG16's performance only went up to 57% with 8 epochs, after which increasing the number of epochs did not improve the accuracy.
- GoogleNet had the best achieved accuracy after parameter tuning amongst the 3 pre-existing models we tried, and this may be because of its deep architecture and ability to capture diverse features at different scales.
- This may not have been reflected in ResNet because the model's shortcut skip connections may not have been a good fit for interpreting and classifying the complex images of facial emotions. VGG16, though being relatively simpler in architecture than ResNet, still has a better average performance with a limited number of epochs.
- The CNN model we created followed a principle of increasing number of feature extractions after every max-pooling layer. We assume that this must have helped in getting an optimum accuracy score because as the size of the images gets smaller, we are able to iteratively retain the most important features following this architecture.

Outcomes

- We were only able to collect a limited number of data points for this project, and we believe that a much larger dataset of these images would have yielded better results in almost every model we attempted.

- ❖ Even though we had asked our participants to exhibit a distinctly separate expression for each emotion, some people's idea of expressing one of the emotions was very different from that of others. This may have reduced homogeneity within the different emotion classes of our dataset.
- ❖ Loading and pre-processing this batch of images was particularly challenging for us; we had to restart our pre-processing steps multiple times until we were sure that they were adequately ready for being tested by the classification models.
- ❖ On performing Canny Detection on the images, we realized that the complexity of edges in every human face may have been an obstacle for the models we tested on the data. Canny edge detection displayed these complex edges more clearly with pre-processed zoomed in pictures.
- ❖ We concluded that a pre-existing model like GoogleNet is a decent fit for emotion classification tasks such as this one if the dataset is larger and better processed, but even then a standard CNN model with custom changes can outperform a complex network like GoogleNet.
- ❖ For facial data in particular, an image classification model that is specifically created for facial mapping and interpretation is required for an excellent accuracy score.