Capstone - Recognizing Human Emotion in Facial Expressions

Executive Summary

This project suggests using a CNN model created by transfer learning from the base EfficientNetB2 architecture and classification layers tuned with Keras Tuner to predict a category of human emotion from grayscale facial images. It performs well for low to medium stakes applications, but should be deployed with care for use cases with high consequences for misprediction. Due to the small set of emotion categories it also has a noticeable difficulty distinguishing between subtle expressional shifts. An ideal use case for the model would be as a contextual sensor, providing additional context information as an input to a more complex application such as a natural language processing (NLP) human machine interface. In a real time application such as this, the model could be further specialized to account for patterns of human emotional shift; allowing previous predictions to serve as weights on future attempts in a time series specialization of the base emotion detector.

Problem Summary

Recognizing the emotional state of people is one of the most important aspects of correctly understanding information exchanged in human interactions. The meaning of words and phrases can take on substantially different meanings, depending on the emotion carried by the speaker. In face to face interactions, cues of the emotional state of the communicator are relayed by body language, tone and emphasis of voice, and especially with facial expressions. Reading a person's face can make the difference between experiencing confusion and offense, or correctly understanding parody and sarcasm. Text based communication, such as on some social media platforms, is a great example of how this emotional context is easily lost and outrage or overreaction are very common. When interacting with machines, this information is nearly always lost, leaving the intent of the user in a somewhat ambiguous state and can leave users frustrated. If machine systems become capable of correctly recognizing the emotional state of their human interlocutor from their facial expression then they

will have much more context for how to interpret the language based interactions with the user.

Solution Design

This project's objective is to create a robust computer vision model that can predict a person's emotional state from their facial expression as captured in a 2D grayscale image. It has explored several CNN architectures known to be suitable for image classification: VGG16, VGG19, ResNet50, ResNet50v2, EfficientNetB2, and EfficientNetv2B2. Each pretrained model was evaluated with untuned hyperparameters to establish a baseline expectation for that algorithm. Then each underwent hyperparameter tuning using Keras Tuner to select the best parameters for the convolutional layers and to establish the classification layer architecture. These tuned models were then trained and finally evaluated on the test data. In order to further maximize accuracy, the test time prediction step performs data augmentation on each test sample to generate multiple 'perspectives' of the input and then uses a mean average of the classification output of each for the final prediction.

Key Insights

In every model, the 'sad' and 'neutral' categories are by far the most likely to be misclassified with each other (figure 1). It is understandable that these categories are tricky to get right as they tend to use the same set facial muscles and shapes, differing by degree rather than quality. There is also a wide variation in the 'resting' face among humans, leaving these more subtly presenting emotions difficult even for humans to recognize unless they are familiar with the individual's body cues. If further performance is needed from the model in these categories, the training set will need to be updated with a broader set of examples for these classes.

Additionally, in this project hyperparameter tuning was limited to 30 trials and tuned fitting was limited to 50 epochs as a matter of available resources. Given more resources to explore a larger space it is quite plausible that the tuned model would be able to better differentiate the 'sad' and 'neutral' classes.

Recommendations for Implementation and Application

Because this solution uses relatively low fidelity inputs for prediction, it can be deployed on a variety of systems that do not have access to high quality inputs. However, it should also be noted that due to the averaging prediction model, the runtime prediction cost is higher than the traditional prediction step, due primarily to the input augmentation. Of note, the prediction computational cost may be ameliorated by converting the logic used in this project to be additional model layers within the classification block of the network. Doing so would enable GPU/TPU acceleration of the input augmentation and averaging during the prediction step.