Patrick Gao

18500 Refocus Statement of Work

03/21/2020

Our original project plan was split up into 3 components, a facial emotion recognition algorithm, a sentiment analysis algorithm, and a web application for storing journal entries labelled with corresponding emotions. The final implementation involved a physical body housed around a Raspberry Pi that would run the two machine learning algorithms and send requests to the web application hosted on a server. Since we cannot make the physical body anymore, we plan to eliminate the use of the Raspberry Pi and migrate our project to a fully server-hosted application. The process in which we develop our machine learning algorithms will remain the same except that upon completion, we are now hosting the algorithms on an AWS server instead of the Raspberry Pi. Personally, I will be working on the facial emotion recognition algorithm, which is still being built in Python using OpenCV, Tensorflow Keras, and Sk-Learn. The architecture of the neural network is still the same as described in our design report, using a pre-trained CNN VGG19 for feature extraction as input into our own trained SVM for emotion classification. Emotion classification will remain the same, split into 6 classes: anger, happiness, sadness, fear, disgust, and surprise. All our requirements for the facial emotion recognition algorithm, as well as the other components will remain the same. We aim for a 65% accuracy for our facial emotion recognition algorithm, as well as a 65% accuracy for our sentiment analysis algorithm, hopefully combined to reach an accuracy of about 75%. We were lucky to have a project that is largely unaffected by a lack of access to physical resources and workspaces since most of our project consists of algorithms taking video and microphone inputs that we can feed using our own devices. As part of our original plan, we were to develop our algorithms on our own devices and port them to the Raspberry Pi; however, now that we cannot use the Raspberry Pi, we can simply keep our algorithms on our own devices and test our inputs/outputs from there. Because the web application component of our project is slightly lacking in terms of effort, we are thinking of adding one of our reach goals to that component – an additional speech tone recognition algorithm for emotion classification. Since I am responsible for the facial emotion recognition algorithm and Vinay is responsible for the sentiment analysis algorithm, Yoojin is now responsible for the speech tone recognition algorithm. Our final goal is to combine the three algorithms to make a final emotion classification of the user's speech. We are no longer aiming to provide immediate feedback through IoT devices, and instead aiming to make emotion classifications with accuracy that meets our requirements – 75% overall emotion classification accuracy. To verify our overall project, we are unlikely to have enough user testing as we had originally planned. Instead, we can record video and audio of ourselves making journal entries and verify that we are getting our expected outputs – the emotion classification as well as adding the transcribed entry to our web application.

Yoojin Lee

Adding a bit more info about the new portion of the project. As Patrick described above, there are a few changes to our project. I will still work on the web application, but since we will not be able to make the physical product, our original plan of connecting the Web application with the Raspberry Pi will no longer be possible.

The original web application was going to communicate with the RPi and receive textual journal entries of the user. Since we will no longer be making the physical product, the tasks related to setting up communication between the Raspberry Pi and web application had to be replaced. Therefore I will work on speech tone analysis to complement Patrick's work on facial emotion analysis and Vinay's sentiment/text analysis. Speech tone analysis uses the voice waveforms to train and classify emotions. Waveforms contain information like pitch and formants that can be used as features. The classifier used to train will be an SVM because the size of our dataset is relatively small and there are not many features that need to be extracted.

I will be using Praat (http://www.fon.hum.uva.nl/praat/) voice analysis software to extract voice waves and measure values such as pitch and formants from sound recordings. The datasets I will use is the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) which can be found here: https://www.kaggle.com/uwrfkaggler/ravdess-emotional-speech-audio?fbclid=IwAR2qFqomaS_N32Oke3Lhbil-wKsI35LZFrY55tMveIkT1ib2nCZxgCiywdo and the Toronto emotional speech set (found here: https://tspace.library.utoronto.ca/handle/1807/24487).