**Introduction**

In this project, the purpose is to train a neural network to recognize emotion in english human speech.**Speech Emotion Recognition (SER)** will be defined to be the act of teaching a neural network to recognize and identify emotions, from a given number of emotions, found in english auditory phrases. The motivation is to mimic how humans can intepret additional meaning in phrases which is not always reflected in the definition of the words in that sentence.

**Literature Review**

Emotion recognition systems based on digitized speech have three fundamental components: signal preprocessing, feature extraction, and classification. In earlier efforts to recognize emotions from the speech signal, there were many implementations based on **SVM** (Support Vector Machine), **GMM** (Gaussian Mixture Model), and **HMM** (Hidden Markov Models).

Deep learning is an emerging research field in machine learning and has gained much attention in recent years. Deep learning techniques such as **DBM** (Deep Boltzmann Machine), **RNN** (Recurrent Neural Network), **RvNN** (Recursive Neural Network), **DBN** (Deep Belief Network), **CNN** (Convolutional Neural Networks) and **AE** (Auto Encoder) are considered a few of the fundamental deep learning techniques used for SER, that significantly improves the overall performance of the designed system.

**Models**

The LSTM model was chosen as the first model since it "...is able to learn long range dependencies using a combination of forget, input and output gates" (Hochreiter & Schmidhuber, 1998). Thus for a time based recognition problem like this one, it can be applied.

LSTM is a special type of RNN (Recurrent Neural Network) designed for handling sequential data, such as speech signals. SER involves modeling time sequences of speech signals, as emotions are often manifested during the temporal evolution of speech. LSTM networks have strong memory capabilities for processing sequential data, allowing them to capture long-term dependencies without being affected by the vanishing gradient problem. This makes LSTM perform well in handling long sequence data (e.g., speech signals) and better capturing emotional information within speech.

The CNN Model was chosen as the second model since it "...uses the already supplied dataset to it for training purposes, and predicts the possible future labels to be assigned" and "it processes all the layers, and hence detects all the underlying features, automatically" (SSLA, 2013). Since our datasets are labelled, this is a prediction/recognition task and we have preprocessed a bunch of the audio's features, it can be applied.

CNN excels at feature extraction through local perception in images, and similarly, for representations like spectrograms or mel-frequency cepstral coefficients (MFCCs) of speech signals, CNN can extract features through local receptive fields.

**Experimental Setup**

The main dataset that will be used throughtout this project is the dataset labelled **Speech Emotion Recognition (en)** compiled by **Dmitry Babko.** It is a combination of four popular datasets which are labelled **CREMA-D** (Crowd-sourced Emotional Multimodal Actors Dataset), **RAVDESS** (Ryerson Audio-Visual Database of Emotional Speech and Song), **SAVEE** (Surrey Audio-Visual Expressed Emotion) and **TESS** (Toronto Emotional Speech Set).

About data preprocessing, in this project we augment the audio by adding noise, pitching and stretching. By analysing the preprocessed data, we extracted 7 features: **MFCC** (Mel Frequency Cepstral Coefficcients), **RMS** (Root Mean Square), **ZCR** (Zero Crossing Rate), **F0** (Fundamental Frequency), **Jitter**, **Shimmer** and **Speech Rate**.

The LSTM model contains 2 **LSTM** layers, 2 **Dropout** layers, 2 **Dense** layers, 1 **Flatten** layer, and 1 **BatchNormalization** layer. The **LSTM** layers will be the layers which are mainly tasked with processing the data. The **Dropout** layers will prevent overfitting. The **Flatten** layer will turn the data into something the Dense layer can process. The **BatchNormalization** layer will be used to standardize our data further. And finally the **Dense** layers will be tasked with producing the final classification. We will use the rmsprop optimizer due to it's fast convergence on large models. We will also use the categorical cross-entropy loss function also known as softmax, because of it's use for multiclass categorization and because we have one hot encoded our labels.

This CNN model comprises 5 **Conv1D** layers, 5 **BatchNormalization** layers, 5 **MaxPooling1D** layers, one **Flatten** layer, and two **Dense** layers. The **Conv1D** layers will be the layers which are mainly tasked with processing the data. The **BatchNormalization** layers will be used to standardize our data further. The **MaxPooling1D** layers will reduce the size of our data. The **Flatten** layer will turn the data into something the Dense layer can process. And finally the **Dense** layers will be tasked with producing the final classification.

Regarding data splitting, we will use **80%** of the data as the training set and **20%** of the data as the testing set.

After trainning, to be able to understand the results fully we must first reason why the metrics we will be using are appropriate. We will use 4 metrics: **Accuracy**, **Recall**, **Precision** and **F1 Score**. It is quite natural and intuitive to use **accuracy** for this recognition task since our dataset is finite and labeled. It measures the overall correctness of the model's predictions and is an appropriate metric. **Rcall** is suitable for the task as it aims to identify the proportion of true positives among the relevant emotions. Since our dataset is finite and labeled, recall can be appropriately used to evaluate the model's ability to capture all positive instances. **Precision** is a relevant metric as it aims to identify the proportion of correctly classified emotions within the model's choices. Given our finite and labeled dataset, it is appropriate to use precision to assess the model's accuracy in predicting positive instances. The **F1 Score** is a harmonic mean of recall and precision and provides a balanced evaluation of the model's performance. Since our dataset is finite and labeled, the F1 Score is an appropriate metric to consider both precision and recall in a single measure.

**Results**

**Conclusions**