Alexandria University
Faculty of Engineering
Computer and Systems Engineering
Department

CSE: Pattern Recognition
Assigned: Mon, May. 05, 2025
Due: Tue, May. 13, 2025

Assignment # 3: Speech Emotion Recognition

1 Problem Statement

Speech is the most natural way of expressing ourselves as humans. It is only natural then to extend this communication medium to computer applications. We define speech emotion recognition (SER) systems as a collection of methodologies that process and classify speech signals to detect the embedded emotions. Below we will show the needed steps to achieve the goal of the assignment.

2 Download the Dataset and Understand the Format

1. We will use CREMA dataset that is available at the following link:

   https://www.kaggle.com/dmitrybabko/speech-emotion-recognition-en

2. Write your own function that loads an audio and listen to each of the classes you have and plot the waveform of the audio.

3 Create the Feature Space

We will create two feature spaces from the audio.

1. You can work in the time domain, or you can work in the frequency domain. There are multiple of features that can help improving the model:

   (a) Zero crossing rate: The rate of sign-changes of the signal during the duration of a particular frame.

   (b) Energy: The sum of squares of the signal values, normalized by the respective frame length.

2. Convert the audio waveform to a mel spectrogram and use this as the feature space.

4 Building the Model

1. Split the data into 70% training and validation and 30% testing.

2. Use 5% of the training and validation data for validation.

   Don't forget to set seed 42 and stratify to true!!

3. **CNN Model**:

   Build a CNN architecture of your own, a simple example is shown in the figure. For the time domain or frequency domain feature space, the feature space will be 1-dimensional; therefore, in the architecture we will be using 1D convolutions. While in melspectrogram feature space, the audio is represented as an image, therefore we will be using 2D convolutions. **N.B.** Do not use the built-in models, but you can implement them from scratch.

5 Big Picture

Compare the performance of the learned models (Different features, different learning models) by realizing the following.

1. Compute the accuracy and F-Score for each model.

2. Plot the confusion matrices and find the most confusing classes.

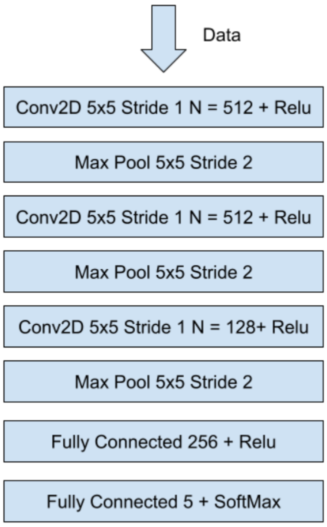
Data
Conv2D 5x5 Stride 1 N = 512 + Relu
Max Pool 5x5 Stride 2
Conv2D 5x5 Stride 1 N = 512 + Relu
Max Pool 5x5 Stride 2
Conv2D 5x5 Stride 1 N = 128+ Relu
Max Pool 5x5 Stride 2
Fully Connected 256 + Relu
Fully Connected 5 + SoftMax

Figure 1: Example CNN architecture.

6 Bonus

- The Best result in the class will get 20 bonus points.

- The second-best result will get 10 bonus points.

7 Submission Notes

- Work in groups of 3 students.

- Submit a **Jupyter Notebook** or a well-commented Python script containing your code.

- Submit a **PDF report** detailing your approach, results, and analysis.

- Ensure reproducibility by setting random seeds and structuring code properly.

Note

- While discussions are allowed, all code and reports must be your own work. Plagiarism will result in a penalty.