TENVOS ASSIGNMENT

Q1. (Writing)- In relation to audio classification, could you provide an explanation of Mel Frequency Cepstral Coefficients (MFCCs), Convolutional Neural Networks (CNNs) for audio data, or audio feature extraction? Could you elaborate on the details of these concepts in the realm of audio classification and provide examples of their application in real-world scenarios? Please substantiate your discussion with relevant scientific references in an appropriate format.

A:

1. Mel Frequency Cepstral Coefficients (MFCCs):

- **Background:**
- The input data used for an audio classification system consists of various audio features. These features capture different characteristics of an audio signal and are useful when it comes to classifying audio clips.
- Some different kinds of audio features are:
  - <u>Spectral:</u> Capture information related to frequency of an audio signal. E.g., Spectral centroid, MFCCs etc.
  - <u>Temporal:</u> Relate to temporal characteristics of an audio signal. E.g., Zero-crossing rate, Root mean square energy. (RMS).
  - <u>Prosodic:</u> Features that convey the variations in pitch, duration, intensity etc.
  - <u>Statistical:</u> Capture information about the distribution of the audio signal. E.g., mean, standard deviation, skewness etc.

- The Mel Frequency Cepstral coefficient (MFCC) is a feature extraction technique or a feature representation of audio data.
- As a feature extraction technique MFCC is used to convert the audio signal from time domain to the Mel- frequency domain. The derived coefficients (MFCCs) are then used as feature vectors to represent the spectral characteristics of the audio signal.
- They are derived from the Mel-spectrogram and used for various audio analysis tasks such as emotion classification, speaker recognition, music analysis etc.

- **Significance of the Mel Spectrogram and the Mel Scale:** The Mel-spectrogram is useful for analyzing the audio signal from a frequency vs time perspective. The Mel-spectrogram captures how the frequency content of a sound signal changes over time. The different colors in the Mel-spectrogram help us gauge the intensity/magnitude of the signal at that point in time. The Mel Scale is a perceptual scale of frequencies that approximates the human ear's responses to different frequencies. It is based on the concept that the human ear is more sensitive to the difference in pitches at a lower frequency than at a high frequency.

- **Steps to calculate MFCCs:**
  - Convert the audio data into short overlapping frames.
  - Apply a windowing function to these frames.
  - Convert the frames from time domain to frequency domain using Fourier Transform.
  - The magnitudes obtained from Fourier transform can be squared to make the Power Spectrum.
  - Apply triangular Mel Filter banks to this power spectrum. The filterbanks calculate the energy within each frequency band.
  - <u>Mel Scaling</u> – Take logarithm of filter bank energies to mimic the human perception of loudness.
  - Perform Discrete Fourier Transform (DCT): To decorrelate the filterbank coefficients and extract the MFCCs.
  - Cepstral coefficient selection - The lower order DCT coefficients are retained as the MFCCs as they contain more relevant information than the higher order ones.

- **Different types of MFCCs:** Different MFCCs correspond to different spectral characteristics of the audio signal. Some of them are:
  - **MFCC 1(C0):** Represents the overall power level of the audio signal.
  - **MFCCs (2 to 13):** Represent the shape of the audio signal and capture variations in the energy.
  - **Higher Order MFCCs (beyond 13):** Represent finer details about the audio signal but may also contain noise.

- **Real Life Applications of MFCCs:**

  i.Emotion Recognition: MFCCs can be used to detect the emotion present in a given audio note. Open-Source tools such as openSMILE use MFCCs for capturing audio features and detecting the emotion in it. Sometimes if not directly, the mean and standard deviation of the MFCC data can be useful for telling if a given audio note is happy or sad as described in the papers 1 and 2.

  ii. **Music Genre Classification:** MFCCs have been used in python libraries such as Librosa and Essentia for music genre classification. This paper, describes the same.

  iii. **Speech Recognition:** Google's Speech to Text API uses MFCCs and other acoustic features for automatic speech recognition.

  iv. **Speaker Identification:** MFCCs prove to be a useful audio feature when it comes to recognizing the speaker of the audio note. As stated in this paper, the speaker can be detected with greater accuracy as we use more Mel-filters to derive the MFCCs.

  v. **Environmental Sound Classification:** The ESC-50 Dataset has been prepared by researchers for environmental sound classification by using MFCCs as one the features to describe various sounds.

2. Convolutional Neural Networks (CNNs):

- **Background:**

○ Convolutional Neural Networks are a type of deep learning models that are designed to analyze multi-dimensional data such as images, videos etc.

○ CNNs use convolutional layers to detect patterns and features in data.

○ Some of the layers in a CNN are:

- Input Layer: To receive the raw data.
- Convolution Layer: Uses the convolution operation to apply kernels to the input data. This generates feature maps.
- Activation Layer – To introduce non-linearity in the network.
- Pooling Layer – To reduce the spatial dimensions and the computational complexity of the data.
- Fully Connected Layer – These layers make predictions based on features.
- Dropout Layer – This layer randomly drops out some activated neurons. This is a regularization technique used to prevent overfitting.

○ Output layer – Produces the model's predictions.

- **CNNs for audio classification**: CNNs are more suitable for data that is multi-dimensional in nature or has a grid like structure. However, audio data is 1D in nature and therefore needs some pre-processing before it can be given as an input to a CNN. Some of the techniques that can be used are:

○ Using a Mel-spectrogram – Converting the audio into a time-frequency representation, so that it can be used by the CNN.

○ Using the Raw- waveform: Audio can be given as a raw input in the form of a waveform to a CNN using a 1D convolutional layer. This way it doesn't need an additional dimension.

- **Real Life Examples of CNNs being used for audio classification:**

○ **Speech Recognition:** This paper elaborates on the details of how a CNN can be used for speech recognition by converting the audio data into a spectrogram. The weight-sharing and the pooling capabilities of the CNN really help in improving the performance of this speech recognition system.

○ **Emotion Recognition:** LSTMs which are an advanced category of CNNs are more effective for the task of emotion recognition. This paper discusses a novel idea of using frame features and the emotional saturation over time frames to train an attention-based LSTM. LSTM being appropriate for sequential data will capture the emotion distributed over frames quite accurately.

○ **Language Recognition:** Bi-LSTMs have been found to be particularly good at recognizing spoken language based on voice notes of 3sec, 10 sec and 30 sec as described in this paper.

○ Apart from the above, CNNs or a flavor of them are also known to be good at tasks such as Environmental Sound Classification, Music Genre Classification, Sound Instrument detection etc.

Q2: (App)- Imagine you have a task to develop a mobile application that allows users to record their voices, apply audio effects, and then classify the resulting sound (for example, music genre, emotion, etc.) using a provided API. Can you describe the architecture of this app depending on your classification use case, the key components and their functionalities, as well as how you would plan and manage the development process? Would you mind providing a hello world example to record the voice?

A: Following is the high-level architecture for this app:

Frontend:

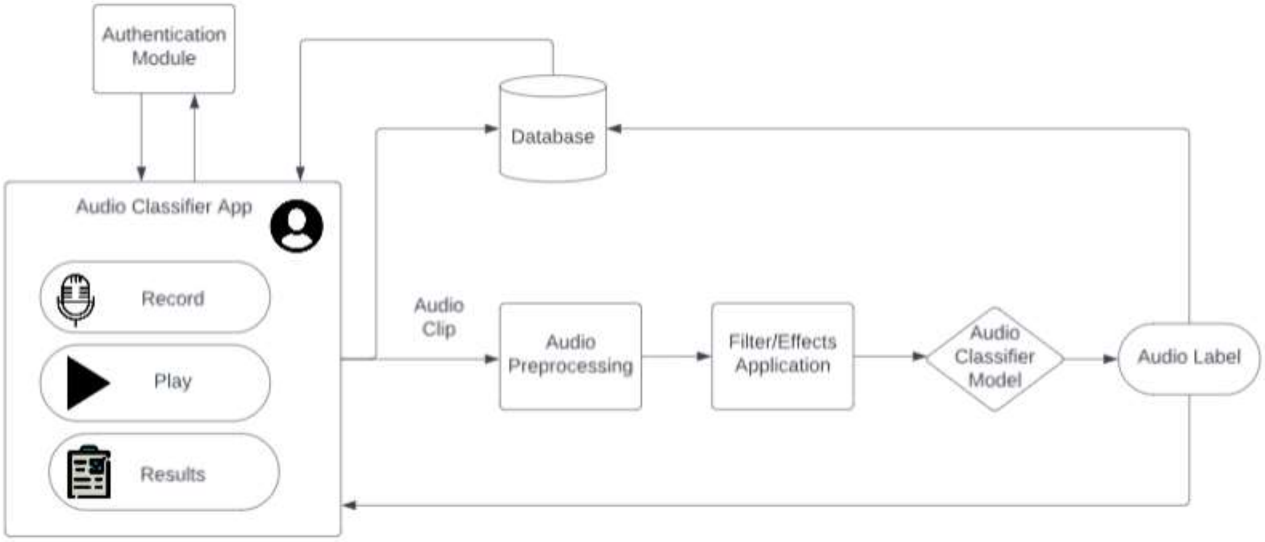
- **User Interface:** The app needs to have a user interface with the following functionality:
- Recording a voice note.
- Choice of audio effects/filters such as echo, reverb, or cartoon voice to choose from.
- Playback feature before and after applying filters.
- Classify button – that outputs the label after the classification task.

- **User profile (Optional):** Depending on the implementation, we can choose to have a separate profile for each user, where a history of their past recordings can be seen.

Backend:

- **Authentication Module (Optional):** Depending on the implementation, along with the User Profile feature in the UI, we can implement an authentication module in the backend, to support login to the profile.
- **Audio Preprocessing:** The recorded audio needs to be pre-processed before classification. Removing background noise, trimming the audio clip to the desired length, breaking the audio signal into small frames, or converting it into a Mel-Spectrogram are some of the pre-processing techniques that can be used here.
- **Applying Effects –** Once the recorded audio clip is cleaned, the effect selected by the user can be applied to it.
- **Audio Classification -** Depending on the memory resources and computational complexity involved in this project, an appropriate ML model can be used to classify the given audio clip. The predicted label of this ML model serves as the output for this app.
- **Database (Optional) –** A database can be integrated into the system if the user needs to retain all the voice recordings and their classifications as part of their profile.

Following diagram describes the high-level architecture of this app:



Q3(ML)- Please consider a real-life project where you need to develop an audio classification system that can identify the speaker's emotion from the audio file. What would be your approach for this task? Discuss the steps you would take, from gathering and preprocessing the data, to choosing a model, training it, and evaluating its performance. Also, consider any potential challenges and how you would mitigate them. Support your pointers with appropriate scientific references whenever necessary.

A:

Following are the different phases of building an emotion detection system:

1) **Data Collection:** This system would need labelled data to train on. Some of the sources for gathering this data could be:

- Recorded customer service calls.
- Podcasts or Audio Books.
- Self – recorded monologues.
- Publicly available datasets from Kaggle, Github
- Voice samples from government research bodies.
- Open source datasets like LibriSpeech, CommonVoice and VoxForge.
- Create your own dataset – Floating a survey and asking users to record voice notes for different emotions.

2) **Data Pre-processing:** This gathered data needs to be cleaned and brought into a consistent format before training the model on it.

Some of the techniques/procedures that should be done on data are:

- **Labelling the data –** All the collected data should have proper labels.
- **Standardization –** All the audio clips/ notes should be of the same duration or should be at least in the same duration range. They should also be in the same format such as .wav files.
- **Noise Removal –** Data should be cleaned of any background noises or any other unnecessary noise.
- **Normalization –** All the features may not have the same range or distribution. Such features may need to be normalized/ scaled.
- **Handling Missing Data –** Some of the feature values may be missing for certain data rows. Such cases need to be handled.
- **Dataset balancing –** Sometimes the dataset may not contain enough samples for a particular label. Some of the different kinds of samples that the dataset should have are:
  - Audio clips for all different types of emotions.
  - Covering different accents and dialects.
  - Spoken by all genders.
  - Spoken by people from different age groups.
- **Data Augmentation –** we can create more samples in the dataset by adding more noise, tweaking the pitch or frequency.

3) **Feature Extraction:** Not all the features in the dataset may be relevant to the task of emotion detection. Some of the features that can represent emotions effectively are:

- MFCCs
- Prosodic Features
- Statistical features
- Spectral features

These features need to be extracted appropriately without any loss before training the model. Sometimes the features extracted from raw audio data may be high – dimensional. Therefore, we may need to do some **dimensionality reduction.**

4) **Data Splitting:** We need to split the data into training, testing and validation sets.

- Training data – Will be used to train the model.
- Validation data – Will be used to tune the hyperparameter of the model and check for any overfitting/underfitting.

- Test data - Will be used to check how the model behaves when provided with never seen before input data. Is also used to evaluate the model.

5) **Model Building and Training:** Speech Emotion Recognition can be done using a variety of algorithms starting from traditional approaches such as Support Vector Machines (SVM), k-Nearest Neighbors along with deep learning techniques like CNNs and RNNs to the state-of-the-art attention based LSTMs and hybrid models. Choosing the right model according to our use case and resource availability is important. This model then needs to train on the training data set.

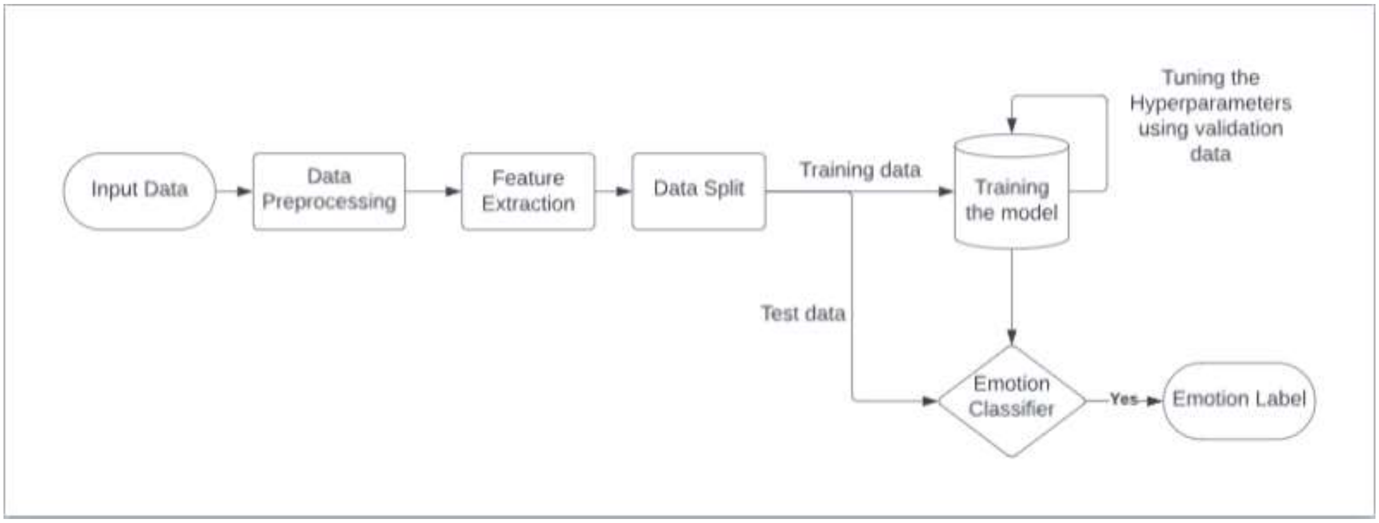
6) **Model Evaluation, Hyperparameter tuning / Regularization:** Now that we have trained our model, it is time to evaluate it using the validation dataset. We can use techniques like k-fold cross validation to improve the system's performance. In the case of a CNN, the no of filters, convolutional layers can be tweaked to achieve the best results. The model is evaluated for its accuracy, precision, and recall rates at this stage to check for overfitting/underfitting. These evaluation metrics are just used to make the model better and are not actually a measure of its performance.

7) **Model testing on test dataset:** The model that we have built is now provided with samples from the test dataset. The model has not seen these samples previously and it now attempts to classify the emotions conveyed in them. The predicted labels are then compared against the test dataset's actual labels to check how well the model performs on test data. Again, the accuracy, precision and recall and other evaluation metrics are calculated to check the model's performance.

8) **Model Deployment:** Once we are sure of the model's performance, it can be deployed on a cloud or on a device as per the project requirements.

9) **Monitoring and Maintenance:** The deployed model needs to have periodic monitoring and maintenance. Every now and then, the model's evaluation metrics need to be checked and the model might need additional training based on the samples it encounters every time.

This system can be visualized using the below diagram:



Challenges in building an emotion detection system:

1) **Data Collection –** Speech Emotion Recognition is a niche task. The number of publicly available, good quality datasets for this task is less. Moreover, finding a dataset covering specific emotions in a specific language is even more difficult. Gathering data by means of data augmentation or extracting chunks of useful data from datasets of another domain can be useful in this case.

2) **Quality of data –** The dataset being used for this task needs to be balanced and should have adequate samples covering audio notes of all dialects, accents, emotions, and age groups. The data samples should also be properly labelled and consistent. Data pre-processing and Dataset balancing techniques should help solve this problem.

3) **Feature-Selection methods –** Often the feature – selection for such tasks is done manually or is domain dependent. This paper discusses the lack of a standardized and generic algorithm for feature selection.

4) **Restrictions on Reusability –** The model built or the dataset that is cleaned and processed for one language cannot be used for another. This is because the tone/pitch in which an emotion is expressed differs in each language. This paper, talks about the same problem.

5) **Emotions being contextual and ambiguous. –** Different people may express different emotions in a different way and in a different accent. Sometimes, there may be a combination of emotions being expressed by a person. For e.g., anger is often combined with feelings of sadness or disgust. At such times there arises the problem of ambiguity in emotions. If such data samples were to be used for our task, then we need to make sure that they are labelled as the emotion that is conveyed most strongly in them.

6) **Privacy and Ethics Concerns –** Audio data collection requires people to record voice notes. In such cases, the users must be notified properly about the usage of this data. No data that has been recorded without the user's permission should be used.

References:

1) Audio Signal Processing for Machine Learning – Valerio Velardo (YouTube) (For all the concepts related to MFCCs and Spectrograms)
2) MFCCs for music modeling – Beth Logan

-Anusha Kalbande