# Speech Emotion Recognition using SVM and DCNN

By-Nishant Bhardwaj

SVM and DCNN in Speech Emotion Recognition (SER) :

Speech Emotion Recognition It's a process where we identify different emotions humans are feeling when talking. Within the context of HCI research (human-computer interaction), it is one the important research fields which aims at making computing more natural by enabling machines to perceive and respond to human emotions. Some examples of applications for SER are customer service, healthcare delivery, education provision and even entertainment.

Purpose of Speech Emotion Recognition (SER) in Day-to-Day Life

Speech Emotion Recognition (SER) has numerous practical applications in day-to-day life. It enhances the way humans interact with technology and improves various aspects of communication and service delivery. Here are some key purposes of SER in daily life.

Emotion Detection in Call Centers: SER can be used to monitor customer service interactions. It detects customer emotions such as frustration anger, or satisfaction. This allows companies to provide timely support. Calls can be escalated to supervisors. As a result overall customer satisfaction improves

Personalized Responses: By recognizing emotional state of a customer, automated systems can tailor responses to be more empathetic and appropriate. This leads to a better customer experience.

Mental Health Monitoring: SER can assist in monitoring patients with mental health issues. It analyzes their speech for signs of stress. It also looks for depression or anxiety. This can provide valuable insights to therapists and healthcare providers.

Assistive Technologies: For individuals with speech impairments or emotional regulation issues SER can assist in communication by identifying emotions. It provides appropriate responses or interventions.

Support Vector Machine (SVM) for SER

Support Vector Machine (SVM) is a popular machine learning algorithm for classification tasks. It works by finding the hyperplane that best separates the data into different classes. SVM is effective in high-dimensional spaces and is widely used for its robustness and accuracy in SER tasks.

Steps for SER using SVM:

Feature Extraction: Extract features like MFCCs from the audio signals.

Data Preprocessing: Normalize the features and prepare the dataset.

Model Training: Train the SVM model using labeled training data.

Prediction: Use the trained SVM model to predict emotions in new audio data.

Deep Convolutional Neural Networks (DCNN) for SER

Deep Convolutional Neural Networks (DCNN) are a type of deep learning model particularly suited for processing grid-like data such as images and spectrograms. DCNNs can automatically

learn hierarchical representations of data. This capability makes them effective for complex tasks like SER.

Steps for SER using DCNN:

Feature Extraction: Extract traditional features (like MFCCs) or use raw audio signals.

Data Preprocessing: Convert audio signals into spectrograms or other time frequency representations.

Model Architecture: Design a DCNN. Use layers like convolutional layers pooling layers, fully connected layers.

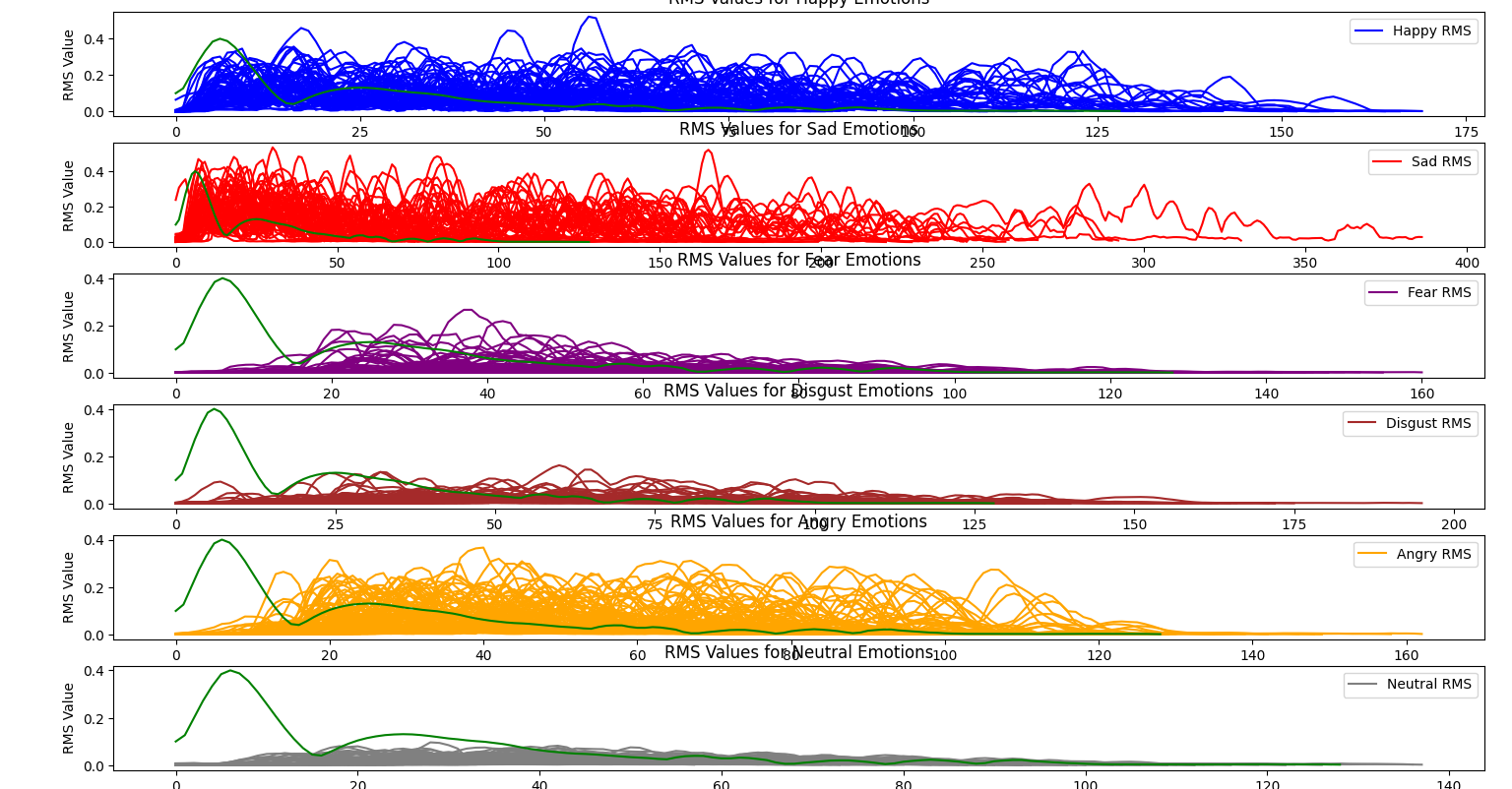
Model Training: Train DCNN using labeled data. Predict emotions in new audio data with trained model.

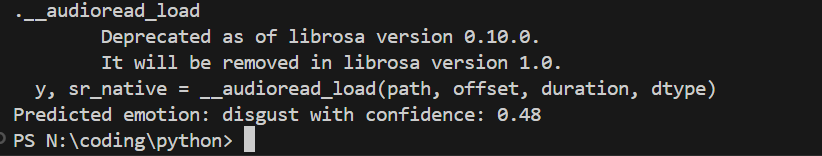
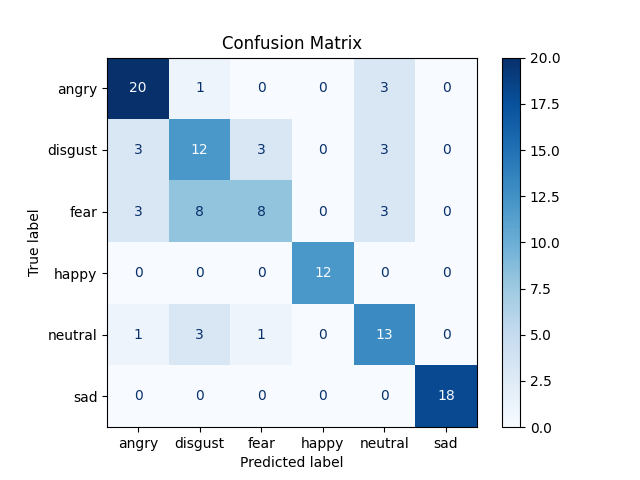
Comparison

SVM: Requires manual feature extraction. Suitable for smaller datasets. Easier to implement. And interpret.

DCNN: Capable of automatic feature extraction. More complex. Requires larger datasets and computational resources. Often provides higher accuracy due to its deep learning capabilities.

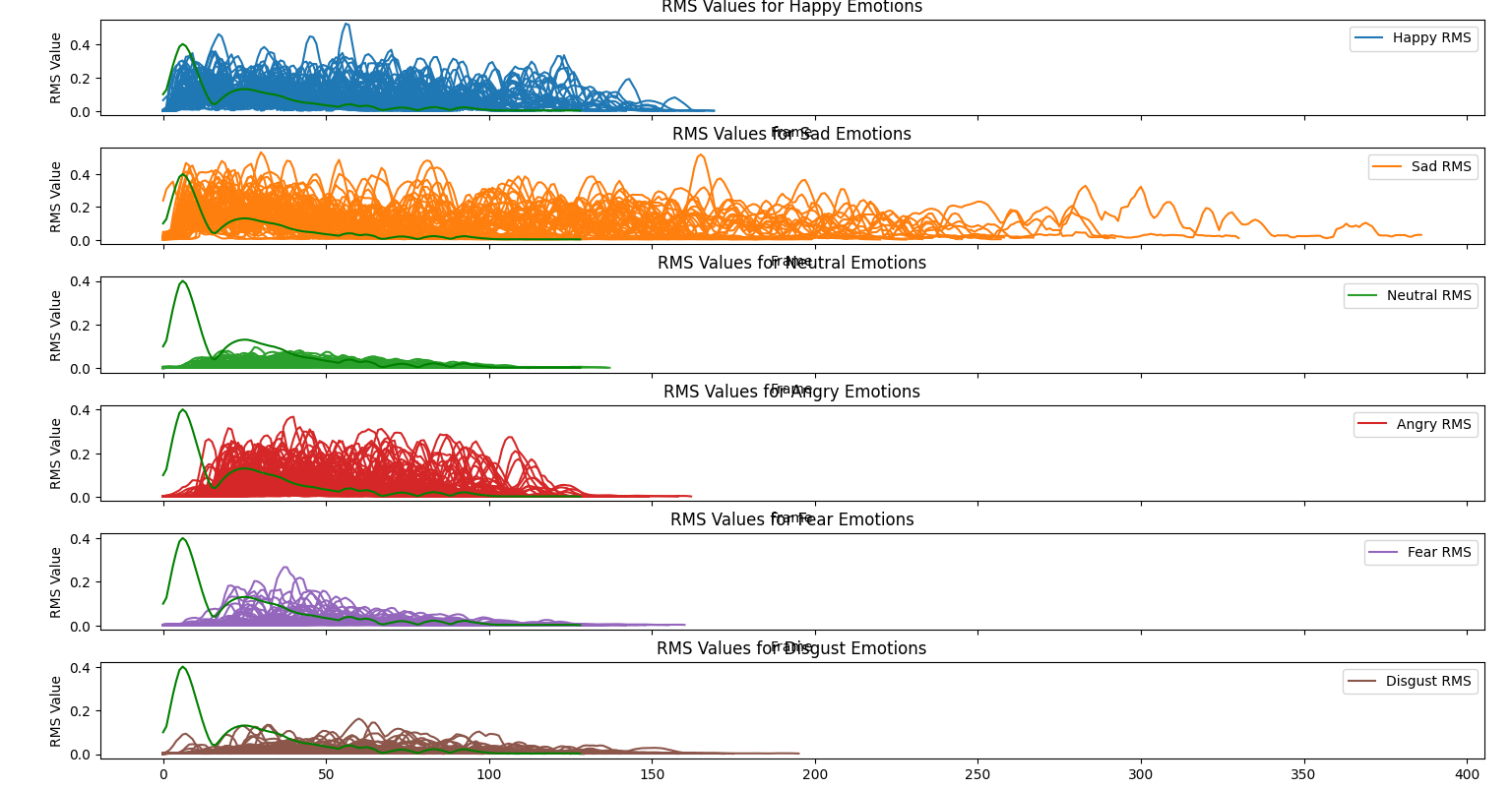
| **Aspect** | **DCNN with MFCC** | **DCNN with LPC** | **DCNN with STFT** | **SVM with MFCC** | **SVM with LPC** | **SVM with STFT** |
| --- | --- | --- | --- | --- | --- | --- |
| **Feature Extraction** | Extracts MFCC features | Extracts LPC features | Extracts STFT features | Extracts MFCC features | Extracts LPC features | Extracts STFT features |
| **Data Representation** | 2D input (time-frequency) for DCNN | 2D input (time coefficients) for DCNN | 2D input (spectrogram) for DCNN | 1D feature vector for SVM | 1D feature vector for SVM | 1D feature vector for SVM |
| **Complexity** | High (needs larger datasets and computational power) | High (needs larger datasets and computational power) | High (needs larger datasets and computational power) | Moderate (simpler model, faster training) | Moderate (simpler model, faster training) | Moderate (simpler model, faster training) |
| **Automatic Feature Learning** | Yes (learns features hierarchically) | Yes (learns features hierarchically) | Yes (learns features hierarchically) | No (relies on manually extracted features) | No (relies on manually extracted features) | No (relies on manually extracted features) |
| **Handling Non-linearities** | Excellent (deep layers capture non-linear patterns) | Excellent (deep layers capture non-linear patterns) | Excellent (deep layers capture non-linear patterns) | Good (kernel trick for non-linear separation) | Good (kernel trick for non-linear separation) | Good (kernel trick for non-linear separation) |
| **Training Time** | Long (due to deep architecture) | Long (due to deep architecture) | Long (due to deep architecture) | Short (depends on feature vector size) | Short (depends on feature vector size) | Short (depends on feature vector size) |
| **Overfitting Risk** | High (needs regularization and dropout) | High (needs regularization and dropout) | High (needs regularization and dropout) | Moderate (depends on model complexity) | Moderate (depends on model complexity) | Moderate (depends on model complexity) |
| **Interpretability** | Low (complex, black-box model) | Low (complex, black-box model) | Low (complex, black-box model) | High (clear decision boundaries) | High (clear decision boundaries) | High (clear decision boundaries) |
| **Performance** | Generally higher with large datasets | Generally higher with large datasets | Generally higher with large datasets | Competitive with smaller datasets | Competitive with smaller datasets | Competitive with smaller datasets |
| **Scalability** | Scales well with more data and computational power | Scales well with more data and computational power | Scales well with more data and computational power | Limited by feature extraction and model size | Limited by feature extraction and model size | Limited by feature extraction and model size |

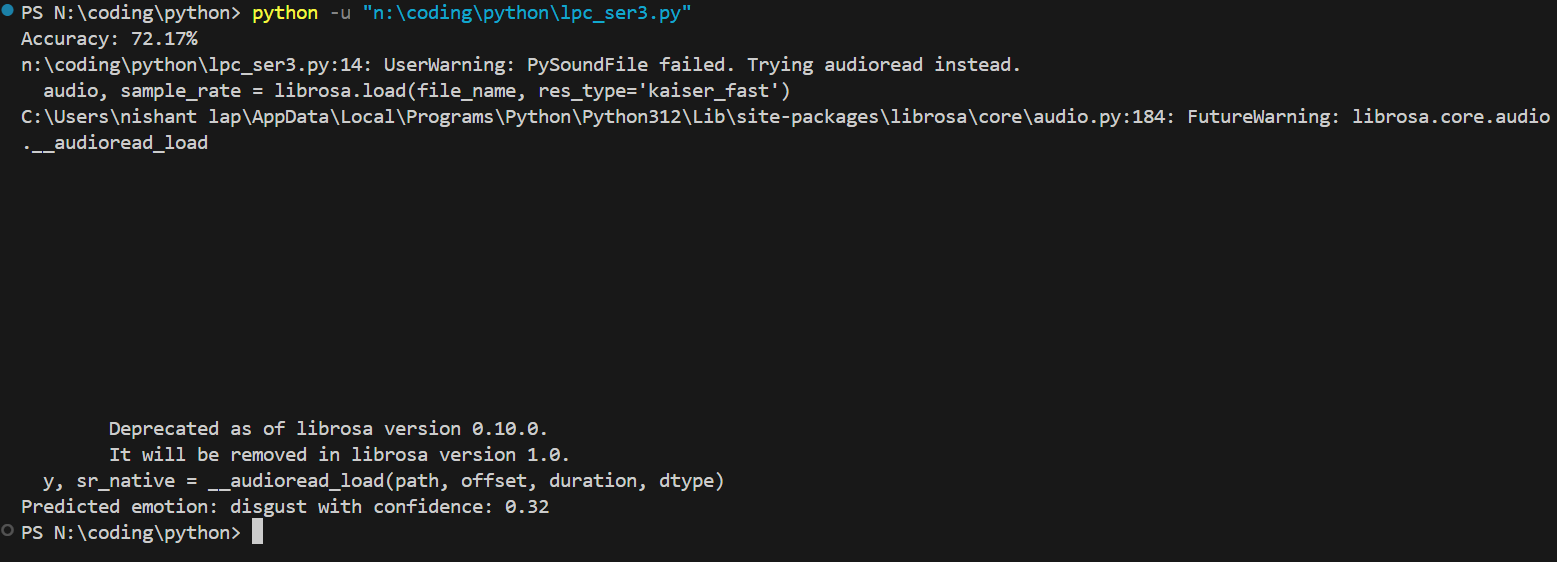
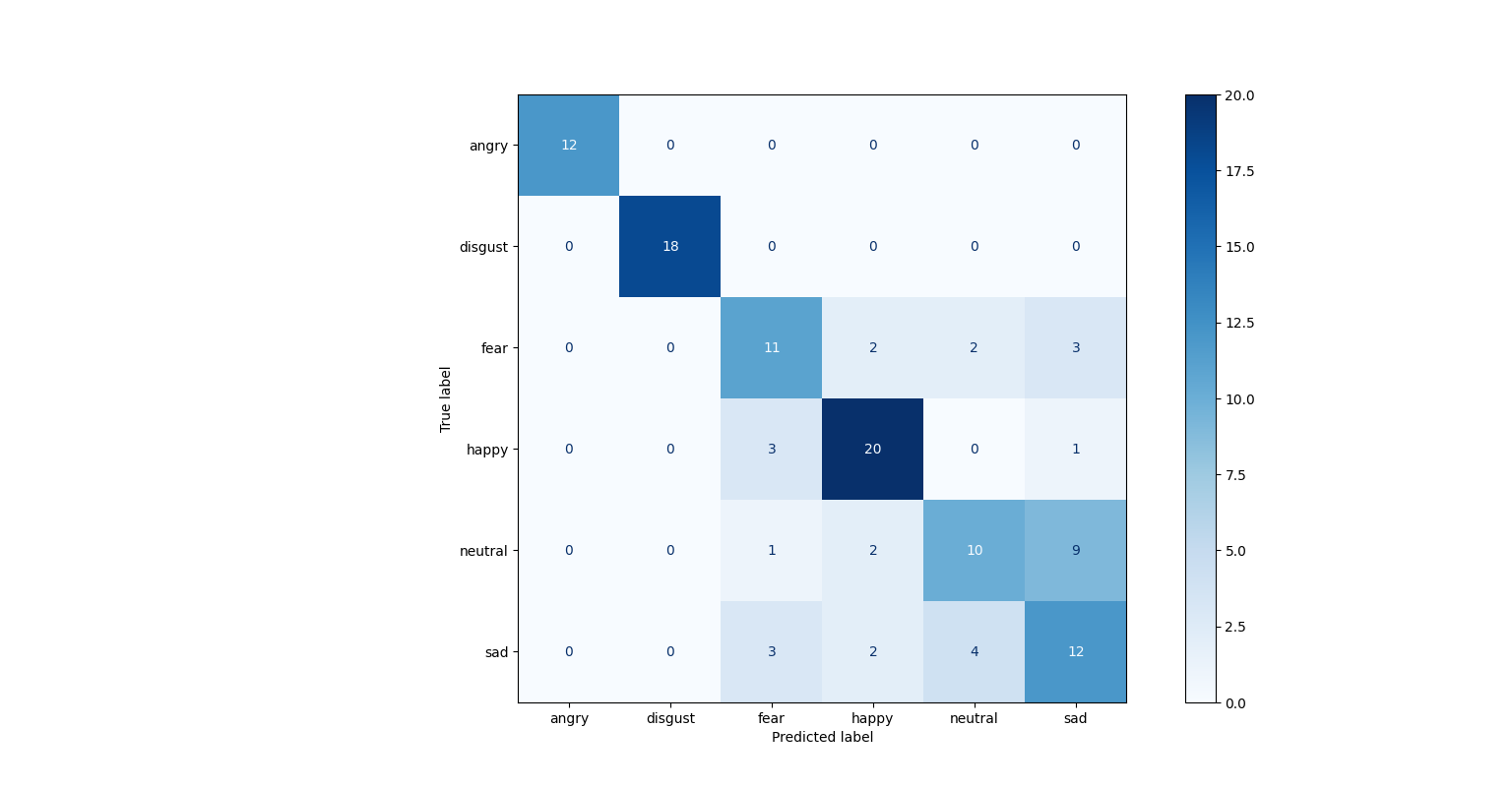
MFCC with SVM method

MFCC stands for Mel-Frequency Cepstral Coefficients

Mel-Frequency: It's a representation of sound frequency more consistent with human perception. It's based on Mel scale which relates perceived frequency of sounds to physical measurements

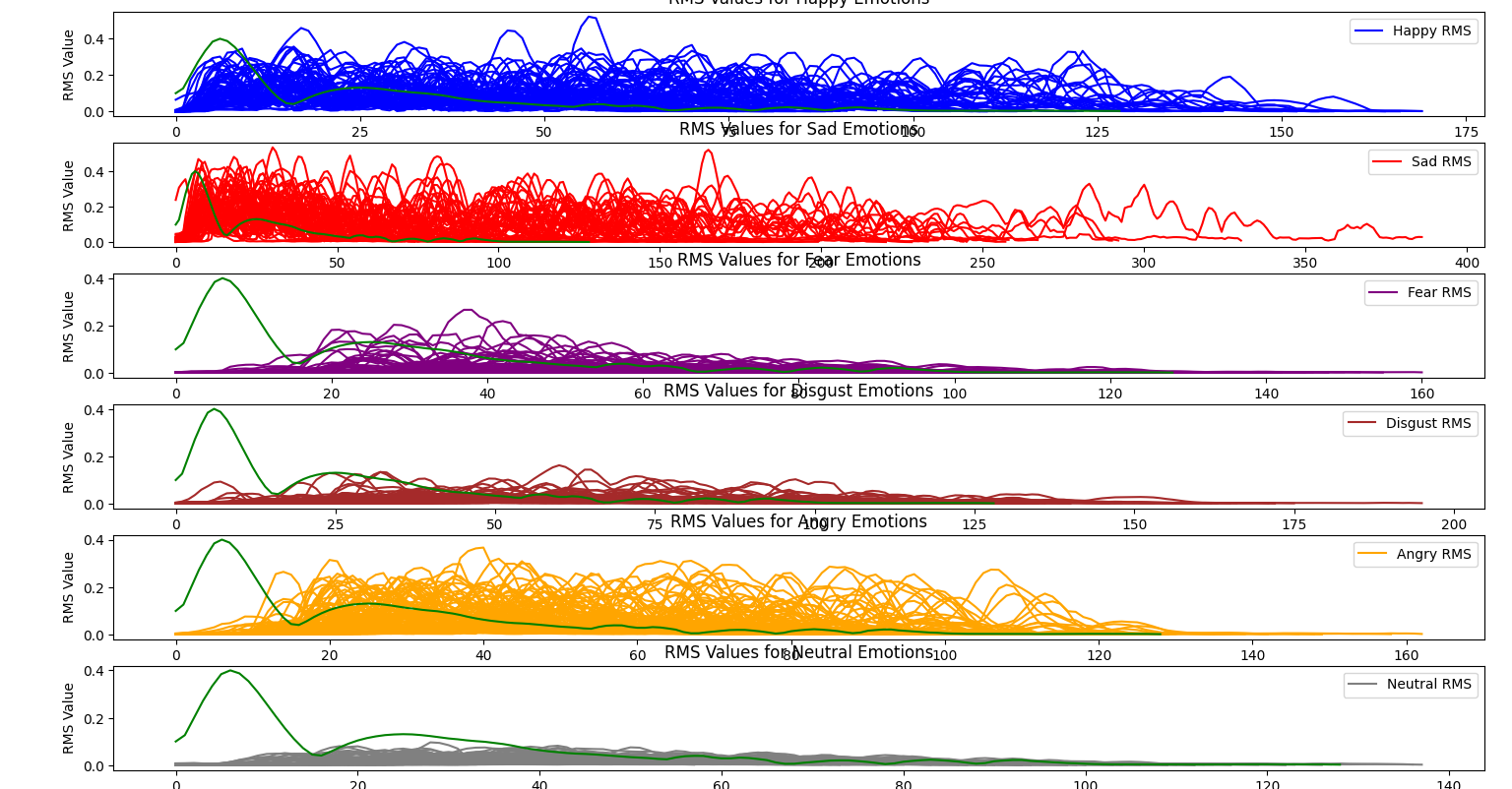
Cepstral: Cepstral analysis involves taking Fourier Transform of log of power spectrum of signal. This helps in separating characteristics of vocal tract (source) from those of vocal folds (filter)

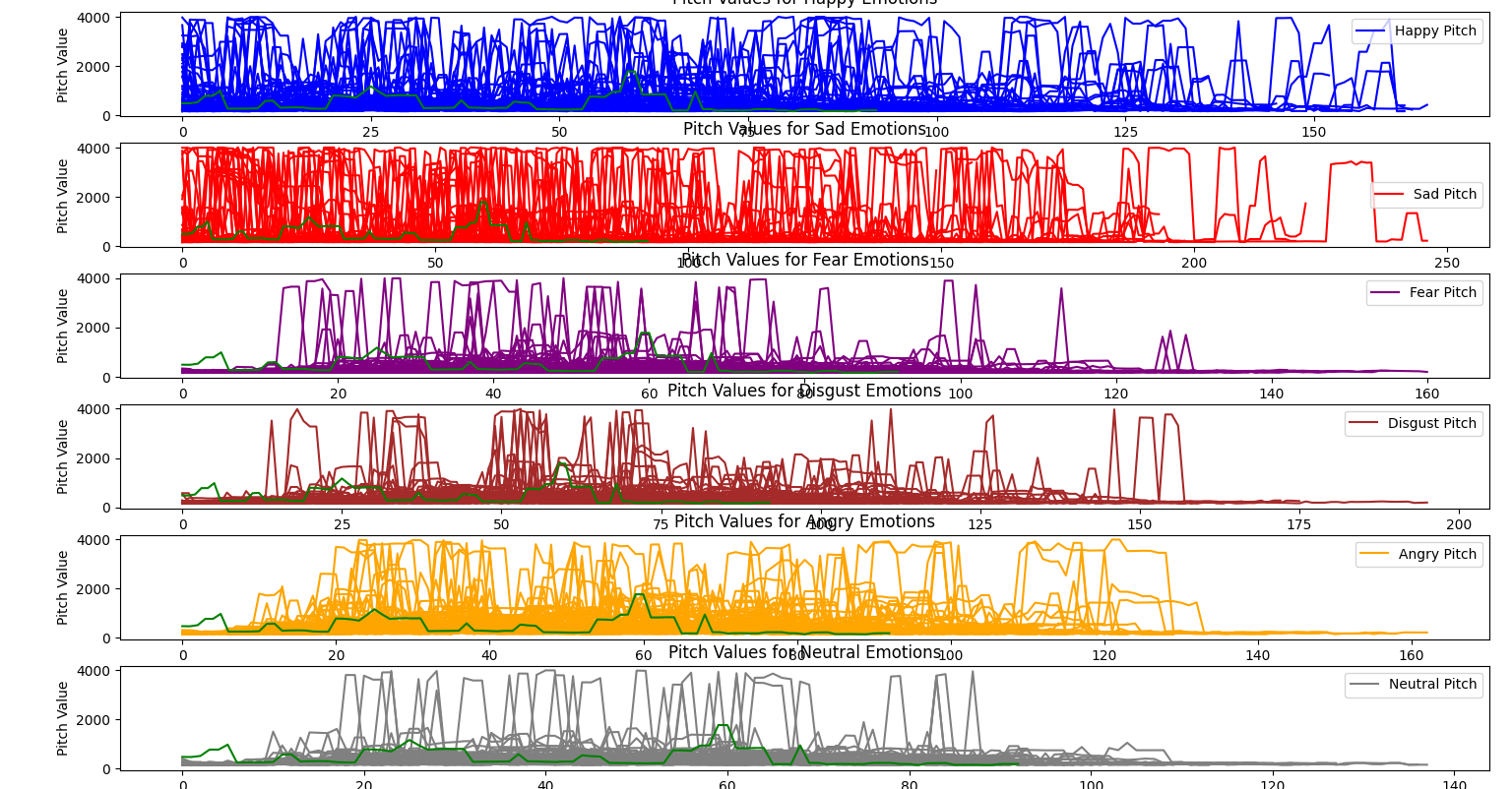
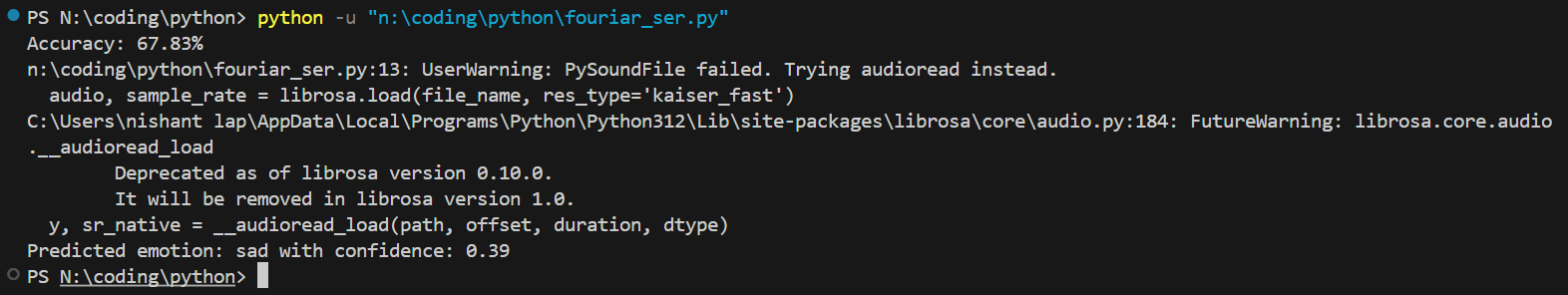
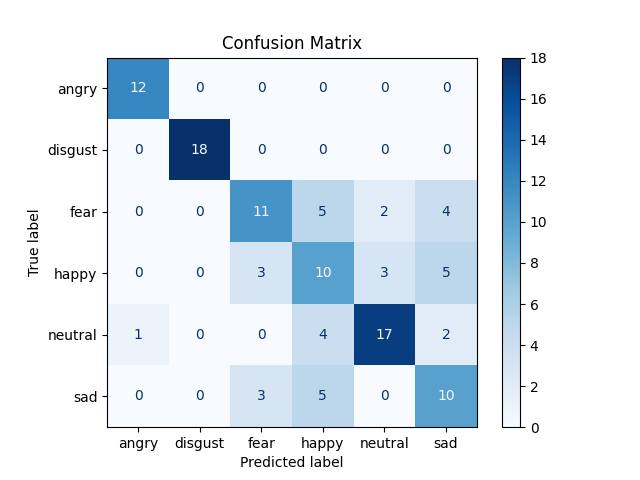
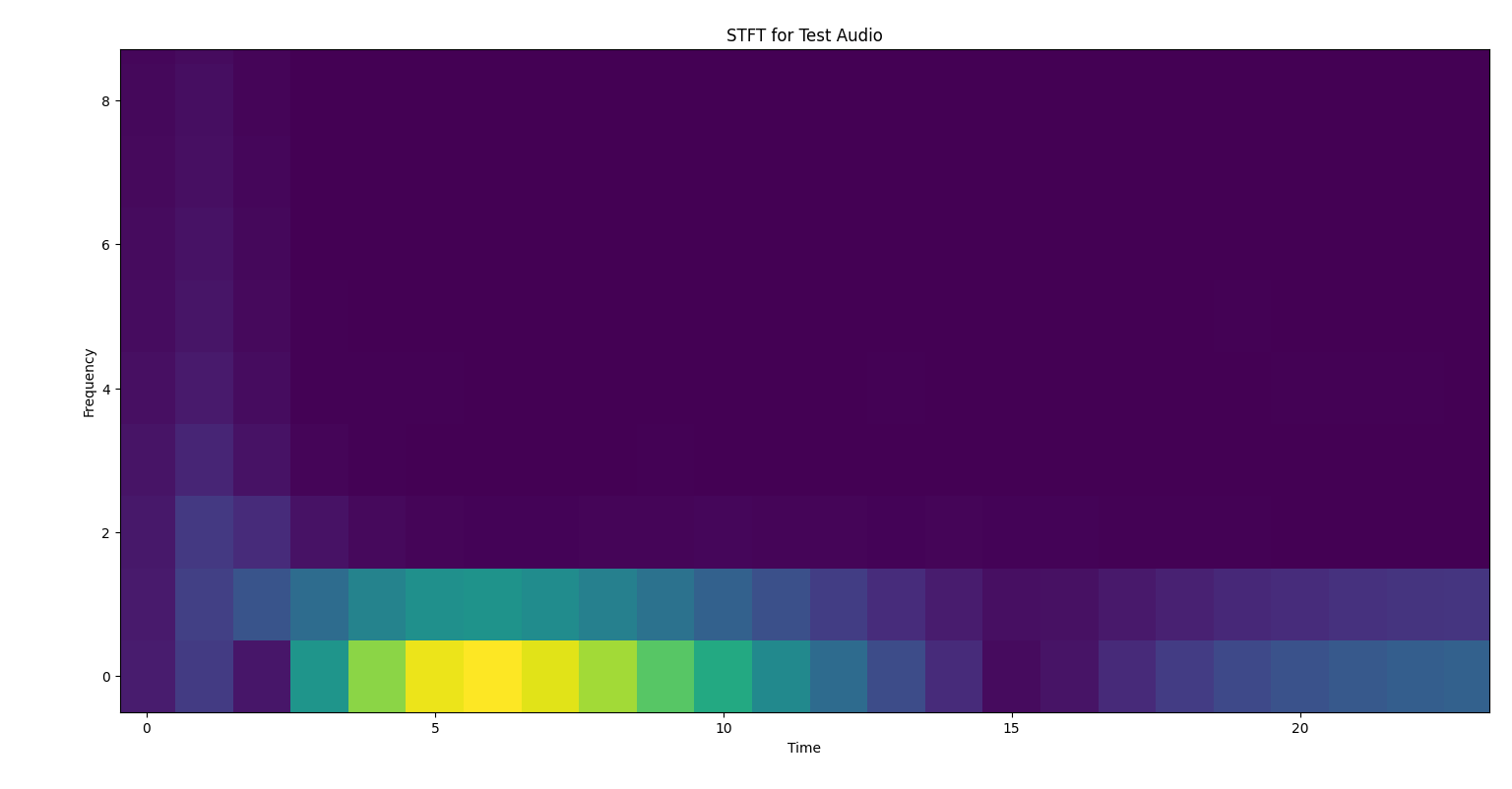
Coefficients: The resulting cepstral coefficients are often further processed. This extracts relevant features. These coefficients capture essential aspects of audio signal in compact form. This is useful for tasks like speech recognition. Also for speaker identification. And emotion recognition.



LPC stands for Linear Predictive Coding. Another widely used technique in speech and audio processing. Particularly used for speech compression. And analysis. Here's a breakdown.

Linear Predictive: LPC assumes speech signal can be modeled as weighted sum of past values. Weights determined by analyzing signal.

Coding: LPC analyzes speech signals. By modeling them as linear combination of past samples. By identifying coefficients of this combination LPC can represent speech signal with fewer parameters.

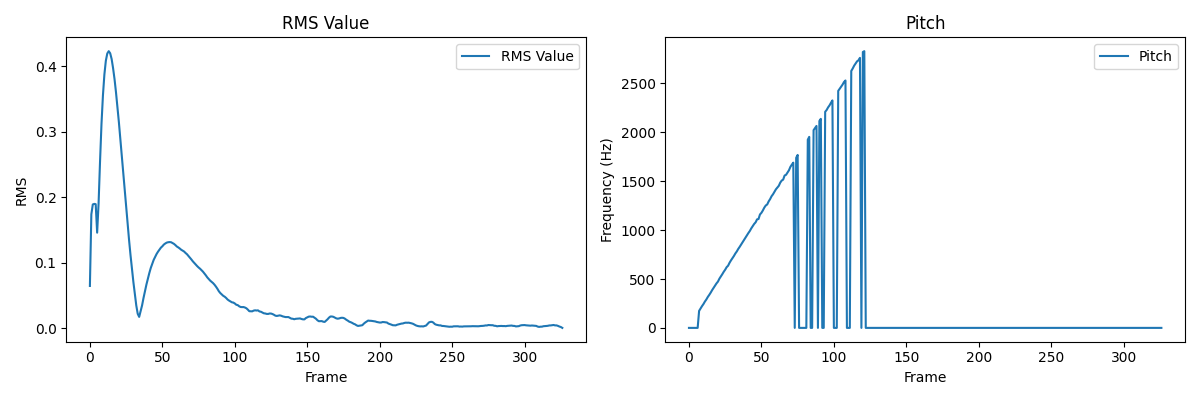
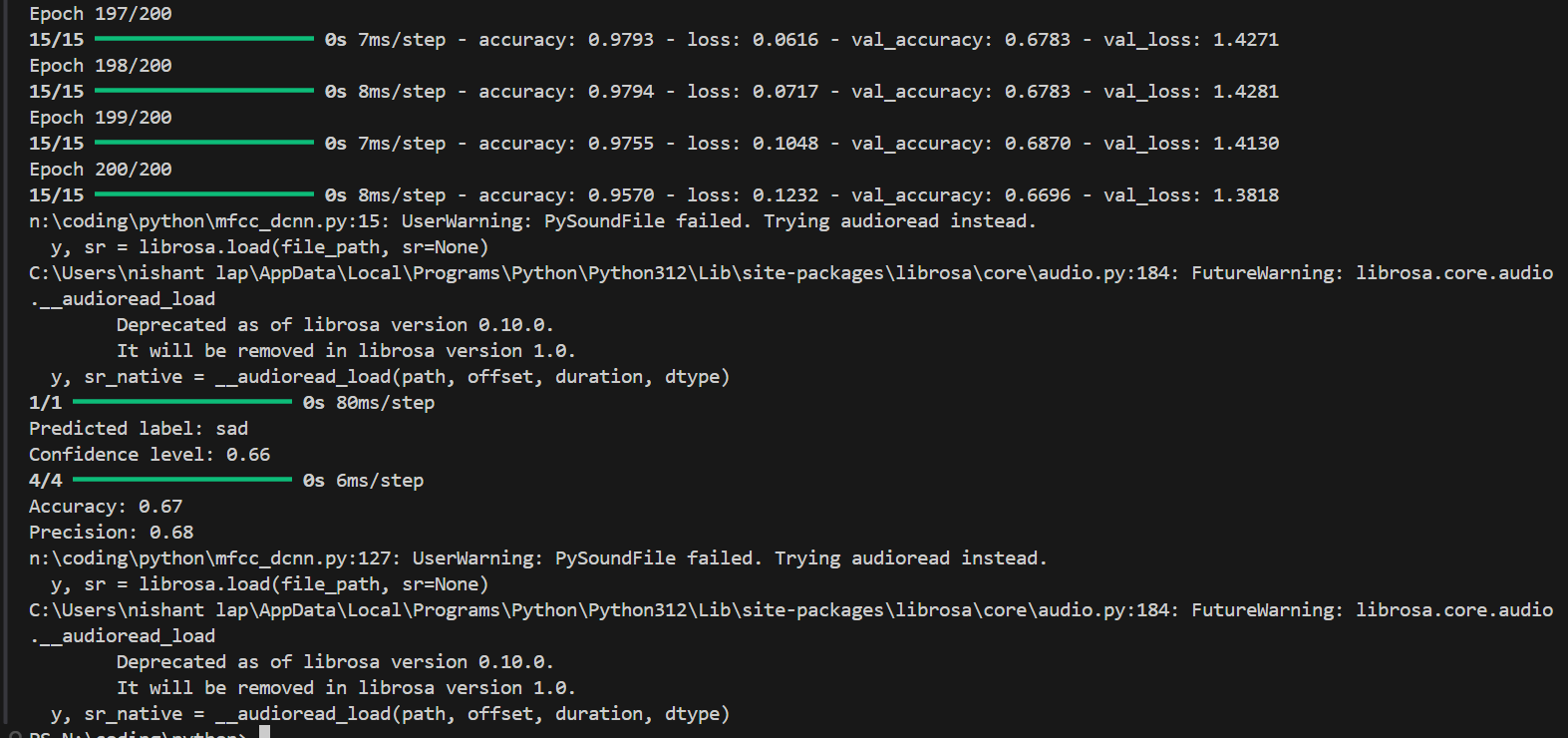
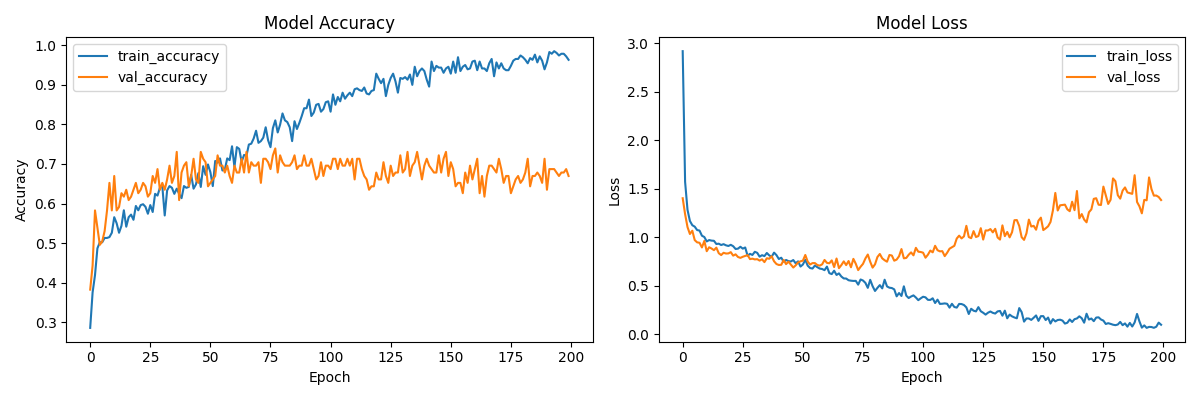
­Short-Time Fourier Transform (STFT) is mathematical technique used to analyze the frequency content of signals that vary over time. It provides way to understand how the frequency spectrum of a signal evolves.

Key Concepts:

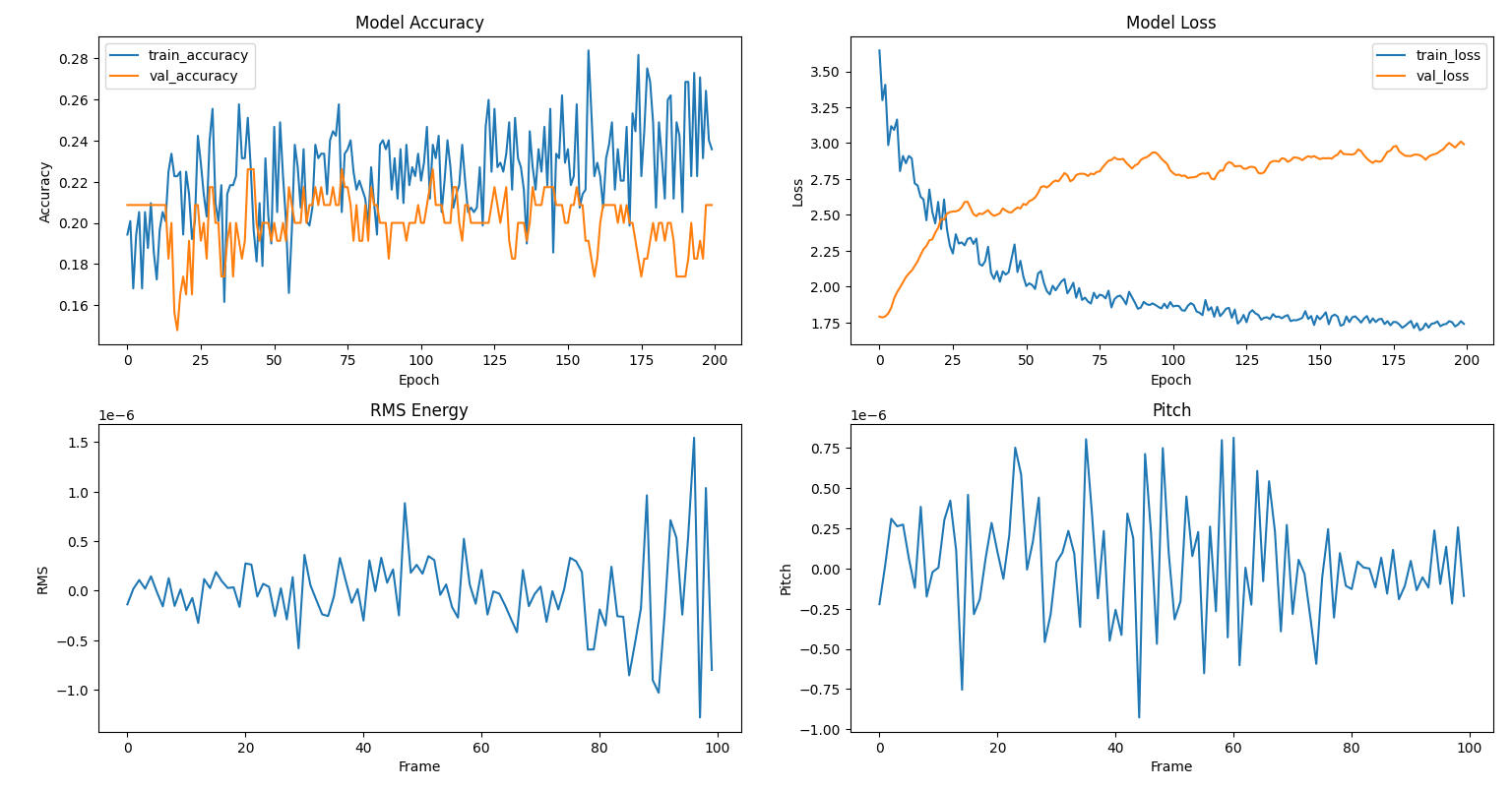
Time-Frequency Representation: Unlike the traditional Fourier Transform. It gives frequency information for the entire signal. STFT provides a time-frequency representation. This allows us to see how the frequency content changes over time.

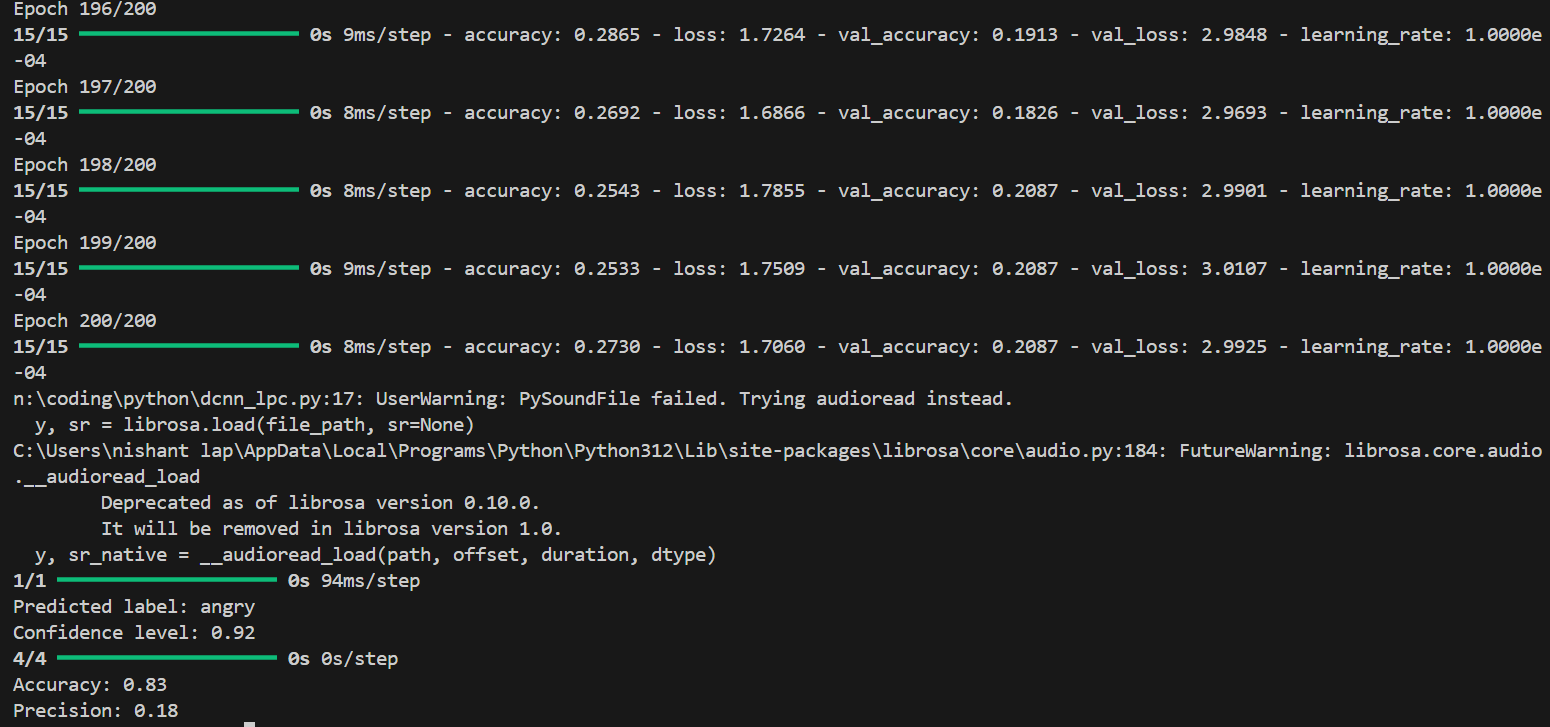
Sliding Window: STFT works by applying a Fourier Transform to a small segment of the signal. This segment or window is moved along the time axis. The Fourier Transform is repeatedly computed. This produces a sequence of spectra.

Window Function: The window function (e.g. Hamming or Hanning or Gaussian) determines the shape and size of each segment. The choice of window affects the trade-off between time and frequency resolution

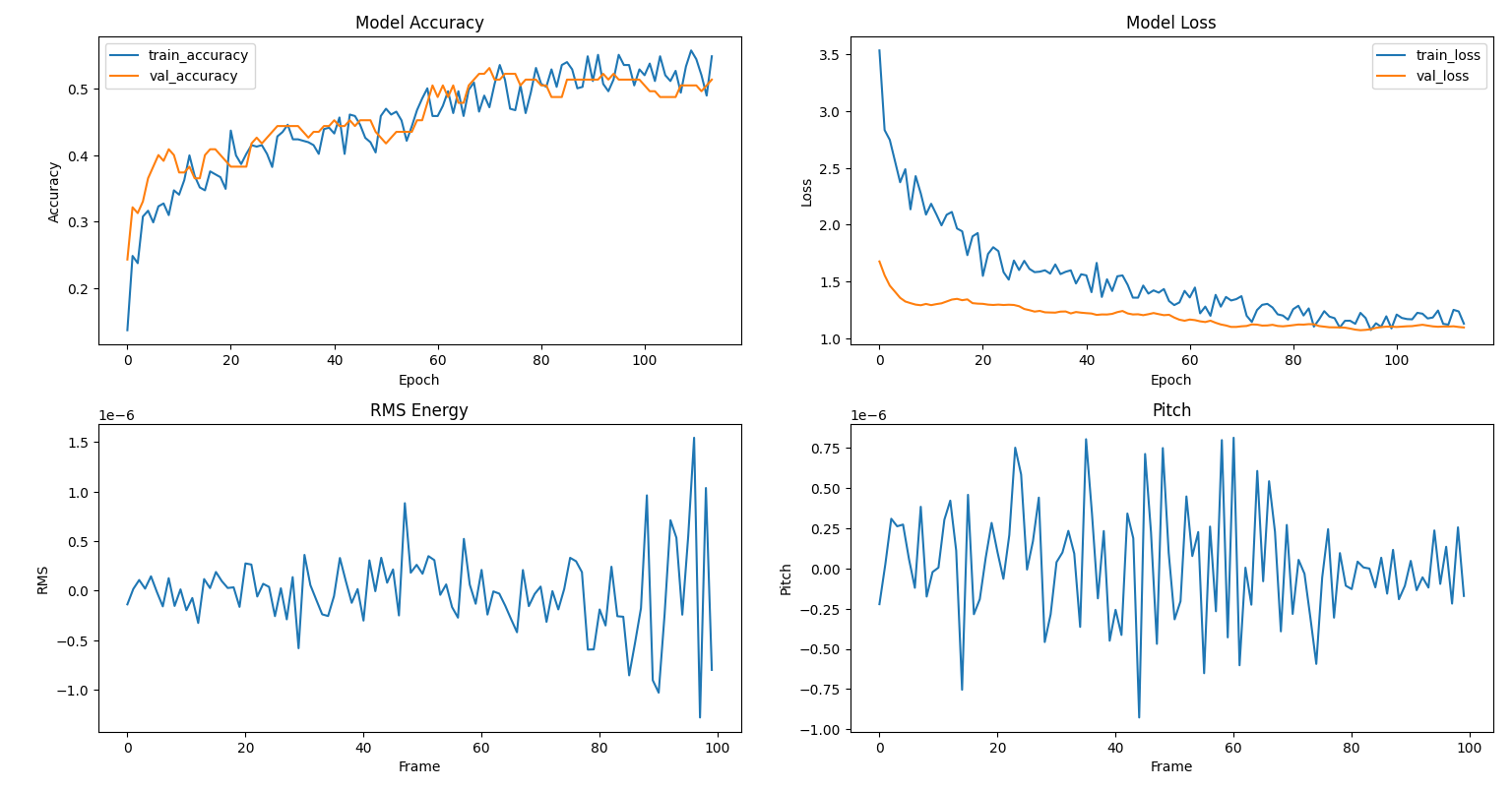
Resolution Trade-Off: A wider window gives better frequency resolution. A narrower window provides better time resolution. But poorer frequency resolution.

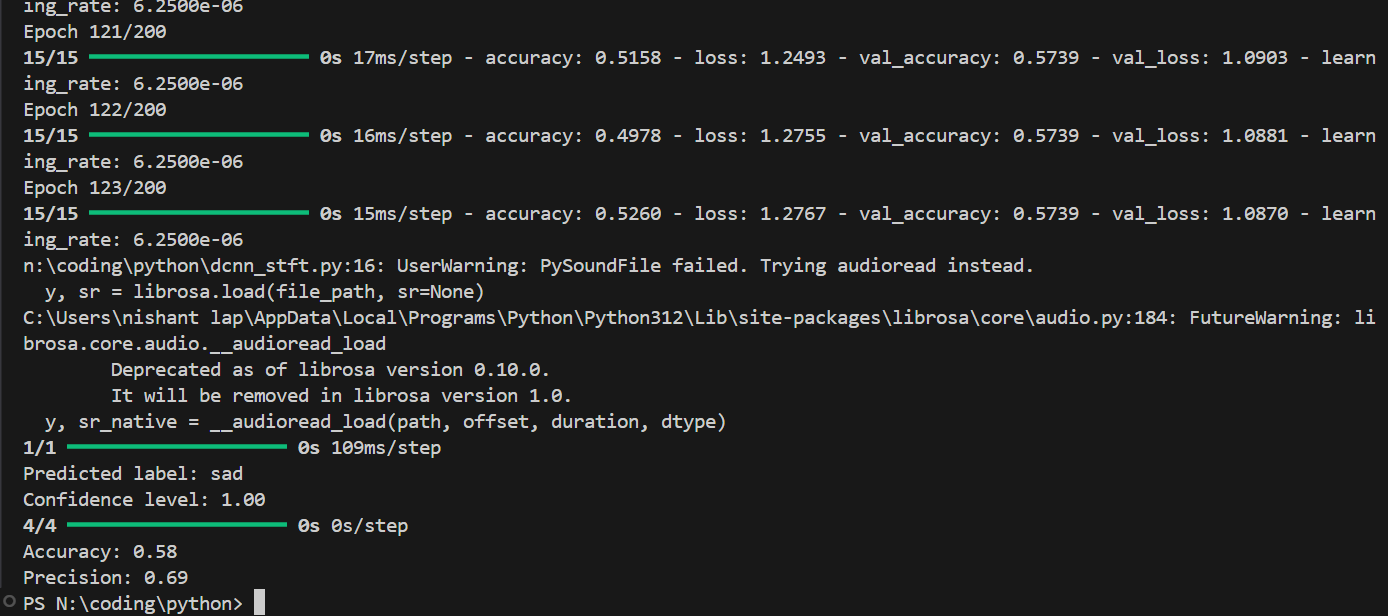
DCNN\_MFCC

DCNN\_LPC



DCNN\_STFT





|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Parameters | DCNN\_MFCC | DCNN\_LPC | DCNN\_STFT | SVM\_MFCC | SVM\_LPC | SVM\_STFT |
| ACCURACY | 0.67 | 0.83 | 0.58 | 0.69 | 0.72 | 0.67 |
| PRECISION | 0.68 | 0.18 | 0.69 | 0.50 | 0.92 | 0.48 |
| CONFIDENCE LEVEL | 0.66 | 0.92 | 1.00 | 0.48 | 0.32 | 0.39 |

Conclusion

MFCC is suitable for both SVM and DCNN. SVM benefits from compactness of MFCC. DCNN leverages its ability to learn from the coefficients.

LPC is effective for SVM due to its low dimensionality. It is also good for speech modeling. DCNN might require additional preprocessing. This is to utilize LPC features effectively.

STFT is highly suitable for DCNN. It provides detailed time-frequency information. SVM might need dimensionality reduction techniques. This is to handle high-dimensional data effectively.

The choice between SVM and DCNN depends on specific application. Data availability and computational resources are also crucial factors. DCNNs generally offer superior performance with complex and high-dimensional data. They however require significant computational power. SVMs are efficient for smaller datasets. They work well with well-defined feature spaces.