**Project Report**

**For Final Project**

**Voice Based Sentiment analysis using English Speech Sentences**

**By:**

Engr. Muhammad Junaid Asif

[mjunaid94ee@outlook.com](mailto:mjunaid94ee@outlook.com)

+92-331-4896216

To:

Muhammad Umair

Instructor for NLP Module, AI Bootcamp

AtomCamp x NUST

February 25, 2025

**Introduction**

The objective of this project is to develop a voice emotion detection system that accurately identifies emotions from audio recordings. This technology can be useful in various applications, including mental health monitoring, customer service optimization, and enhancing user experience in interactive voice systems. The primary emotions targeted in this project are neutral, calm, happy, sad, angry, fear, disgust, and surprise.

**Technical Approach**

**Methodology**

The methodology involved several key steps:

1. **Data Collection**: Utilized the RAVDESS emotional speech dataset, which contains audio samples labeled with different emotions and their intensities.
2. **Data Preprocessing**:
   * Audio files were loaded and converted into a suitable format for analysis.
   * Spectrograms (Mel spectrograms) were generated from the audio signals to serve as input features for the model.
3. **Model Development**:
   * A Parallel Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) model architecture was implemented to capture spatial and temporal features of the audio data.
   * The model was designed to classify emotions based on the extracted features.
4. **Training and Evaluation**:
   * The dataset was split into training, validation, and test sets.
   * The model was trained using stochastic gradient descent with appropriate loss functions and evaluation metrics.

**Tools and Frameworks**

* **Programming Language**: Python
* **Libraries**:
  + librosa for audio processing and feature extraction.
  + pandas and numpy for data manipulation.
  + torch for building and training the neural network model.
  + matplotlib and seaborn for data visualization.
* **Development Environment**: Jupyter Notebook or local Python environment.

**Development Process**

**Step-by-Step Implementation**

1. **Data Loading**:
   * Utilized os and pandas to traverse the dataset directory and compile metadata (emotion, intensity, gender, file path).
2. **Data Visualization**:
   * Created bar charts to visualize the distribution of emotions, intensity, and gender using matplotlib.
3. **Audio Signal Processing**:
   * Loaded audio files using librosa, applied normalization, and generated Mel spectrograms.
4. **Data Augmentation**:
   * Implemented a function to add Additive White Gaussian Noise (AWGN) to the training data for robustness.
5. **Model Definition**:
   * Created a custom neural network class that includes multiple convolutional and LSTM layers.
6. **Training Loop**:
   * Implemented a training loop that updates model weights based on the loss calculated from predictions.
7. **Evaluation**:
   * Evaluated the model's performance on the validation and test datasets, calculating accuracy, recall, and F1 scores.
8. **Results Visualization**:
   * Plotted confusion matrices and loss curves to visualize model performance.

**Outcomes and Results**

**Key Learnings**

* The importance of data preprocessing and augmentation for improving model performance.
* Understanding how different neural network architectures can effectively capture audio features.

**Insights**

* The model achieved a reasonable accuracy, demonstrating the feasibility of emotion detection from audio.
* The confusion matrix highlighted specific emotions that were more difficult to classify, pointing to areas for improvement.

**Improvements Made**

* Adjusted learning rates and batch sizes during training to optimize performance.
* Implemented dropout layers to reduce overfitting.

**Challenges and Solutions**

**Challenges Faced**

* **Data Imbalance**: Certain emotions had significantly fewer samples, making it harder for the model to learn those classes effectively.
* **Model Complexity**: The initial model was too complex and took too long to train.

**Solutions**

* Employed data augmentation techniques to artificially increase the number of samples for underrepresented emotions.
* Simplified the model architecture and used transfer learning from pre-trained models to reduce training time.

**Future Improvements**

**Recommendations for Future Work**

* **Expand Dataset**: Incorporating additional datasets or more diverse audio samples could improve model robustness.
* **Real-time Processing**: Implementing a real-time emotion detection system for live audio input.
* **Enhanced Features**: Exploring additional features such as pitch, tone, and speech rate to improve emotion classification accuracy.
* **User Feedback Integration**: Developing a feedback loop mechanism where the system learns from user corrections to improve over time.

In conclusion, this project demonstrated the potential of using machine learning techniques to analyze emotional content in speech. Through careful data handling, modeling, and evaluation, a functional emotion detection system was built, paving the way for future advancements in this field.