DEEP LEARNING COURSE
PROJECT BASED LEARNING - SAMPLE PROJECT

Speesch Emotion Recognition Project

1. Objective

This project involves building a deep learning system for identifying and classifying emotions from speech data, a key application of artificial intelligence in human-computer interaction. Students will utilize their knowledge from the course to design, train, and deploy a model capable of recognizing emotions from audio recordings.

The project follows the deep learning pipeline outlined in the syllabus:

1. **Data Preprocessing:** Working with datasets like EmoDB and RAVDESS to prepare audio data for analysis.
2. **Feature Extraction:** Extracting meaningful audio features such as MFCCs, chroma, and spectral properties using tools like Librosa.
3. **Model Development:** Designing and training neural network models to classify speech emotions effectively.
4. **Evaluation and Deployment:** Assessing performance using metrics like accuracy and confusion matrix and deploying the model for inference on new audio data.

Through this project, students will gain practical experience in speech emotion recognition, reinforcing course concepts and preparing them for real-world applications in AI and audio processing.

2. Tools Required

- Development Environment:
  - Jupyter Notebook or Google Colab.
  - Python programming language.
- Deep Learning Libraries:
  - TensorFlow/Keras or PyTorch for model implementation and training.
- Audio Data Processing:
  - Librosa for feature extraction (e.g., MFCCs, chroma, spectral features).
- Dataset Access:
  - EmoDB and RAVDESS datasets for labeled emotional audio recordings.
- Visualization:
  - Matplotlib, Seaborn for plotting and visualizing data and model performance.
- Compute Resources:
  - Local GPUs or cloud platforms (Google Colab, AWS, Azure) for model training.

3. Learning Outcomes

- Understand the principles and challenges of Speech Emotion Recognition (SER).
- Preprocess and augment audio data for machine learning tasks.
- Extract meaningful audio features such as MFCCs and chroma features for emotion classification.
- Design and train a neural network model for speech-based emotion recognition.
- Evaluate model performance using metrics like accuracy, precision, recall, and confusion matrix.
- Deploy the model and demonstrate its ability to classify speech emotions in real-time or batch processing.

4. Deliverables

1. Preprocessed audio data and feature extraction scripts.
2. A trained neural network model capable of classifying emotions from speech.
3. Evaluation results including accuracy, confusion matrix, and other performance metrics.
4. A demonstration system showcasing the model's predictions on unseen audio samples.
5. A comprehensive project report detailing the methodology, results, and challenges.
6. Presentation slides summarizing the project and a demo video.

5. Detailed Project Plan

Week 1: Introduction and Dataset Exploration

Introduction to Speech Emotion Recognition (SER) and its applications.

Tasks:
- Review EmoDB and RAVDESS datasets.
- Understand the data structure and available emotions.
- Organize and partition datasets into training, validation, and testing sets.

Week 2: Audio Preprocessing and Feature Extraction

Learn techniques for audio data preprocessing and feature extraction.

Tasks:

- Normalize and clean audio files.
- Extract features like MFCCs, chroma, and spectral features using Librosa.
- Save extracted features for use in model training.

Week 3: Data Exploration and Class Balancing

Analyze the distribution of classes and address imbalances.

Tasks:

- Explore features using visualizations (e.g., feature histograms, heatmaps).
- Balance classes using techniques like oversampling or weighted loss functions.
- Create feature-label mappings for model input.

Week 4: Model Design and Initial Training

Design a neural network model tailored for speech emotion classification.

Tasks:

- Implement a sequential model with dense layers.
- Train the initial model using extracted features.
- Evaluate and log performance on the validation set.

Week 5: Hyperparameter Tuning

Optimize the model for better performance.

Tasks:

- Experiment with different architectures, activation functions, and optimizers.
- Use Keras Tuner or Grid Search for systematic hyperparameter optimization.
- Record the best-performing configurations.

Week 6: Advanced Model Training and Regularization

Apply advanced training techniques to improve model generalization.

Tasks:

- Add Dropout and BatchNorm layers to mitigate overfitting.
- Train the model on the full dataset with augmented audio samples.
- Analyze training logs and evaluate results.

Week 7: Model Evaluation and Error Analysis

Evaluate the model and identify areas for improvement.

Tasks:

- Use metrics like accuracy, precision, recall, and confusion matrix.
- Perform error analysis to understand misclassifications.
- Update the model or preprocessing pipeline based on findings.

Week 8: System Integration and Demonstration

Integrate the trained model into a demonstration system.

Tasks:

- Write a script for batch or real-time inference.

- Test the system with unseen audio samples.
- Prepare a demo showcasing the system's capabilities.

Week 9: Report Writing and Presentation Preparation

Document the entire project and prepare for final presentation.
Tasks:

- Write a comprehensive report detailing all steps and findings.
- Prepare presentation slides and a demo video.
- Include visuals like accuracy/loss curves and confusion matrices.

Week 10: Final Presentation and Submission

Present the project and submit final deliverables.
Tasks:

- Deliver a 5-minute presentation and live demo.
- Submit the final report, scripts, and trained model.
- Participate in Q&A and incorporate feedback.

6. Grading Rubric

7. Data Preprocessing and Feature Extraction: Quality of scripts and feature engineering (20%).
8. Model Design and Training: Originality, performance, and implementation quality (25%).
9. Evaluation Results: Metrics like accuracy, confusion matrix, and error analysis (20%).
10. System Integration and Demo: Functionality and effectiveness of the demonstration system (20%).
11. Documentation and Presentation: Clarity and completeness of the report and presentation (15%).