Instruction Document for Audio Classification Project Using UrbanSound8K Dataset

# Project Overview:

This project involves audio classification using the UrbanSound8K dataset. Two models are compared: a Neural Network (NN) and an Audio Spectrogram Transformer (AST) model. The goal is to process the audio files, extract features, and train both models to predict sound classes. The AST model achieved an accuracy of 91.95%, while the NN model achieved 90.7%.

# Project File Structure:

/AST\_Project  
├── /data  
│ ├── /urban\_sound\_8k  
│ ├── /ast\_features.pkl  
├── /test  
│ ├── /X\_test.npy  
│ ├── /y\_test.npy  
│ ├── /y\_pred.npy  
│ ├── /confusion\_matrix.png  
│ ├── /training\_history.csv  
│ ├── /sample\_predictions.png  
│ ├── /classification\_report.txt  
├── /notebooks  
│ ├── 1\_preprocess\_data.ipynb  
│ ├── 2\_train\_nn\_model.ipynb  
│ ├── 3\_train\_ast\_model.ipynb  
├── /scripts  
│ ├── train\_nn\_model.py  
│ ├── train\_ast\_model.py  
│ ├── feature\_extraction.py  
│ └── utils.py

# Requirements:

Before executing the scripts, ensure the following Python packages are installed:

pip install tensorflow==2.10  
pip install pandas  
pip install numpy  
pip install scikit-learn  
pip install matplotlib  
pip install seaborn  
pip install tqdm

# Google Colab Environment Setup:

If you are working in Google Colab, you will need to mount Google Drive to access datasets and store model outputs. Use the following command to mount Google Drive in Colab:

from google.colab import drive  
drive.mount('/content/drive', force\_remount=True)

# File Path Structure:

Ensure the following file paths are set up correctly in Google Drive (or your local machine if you are not using Colab):

- Dataset Folder: "/content/drive/My Drive/AST\_Project/data/urban\_sound\_8k/"  
- Features Folder: "/content/drive/My Drive/AST\_Project/ast\_features.pkl"  
- Test Folder: "/content/drive/My Drive/AST\_Project/test/"

# Scripts and Notebooks:

The execution sequence of the files is as follows:

## 1. Step 1: Preprocess Audio Data (feature\_extraction.py)

This script extracts features from the UrbanSound8K audio dataset and stores them as pickle files.

python scripts/feature\_extraction.py

What it does:

- Loads the UrbanSound8K dataset. - Preprocesses the audio files (e.g., convert them to spectrograms). - Extracts AST features. - Saves the features to the file ast\_features.pkl. Similarly, for NN Features as well – Saves the features to the file cnn\_rnn\_features.pkl.

Ensure the audio files are in the correct directory (/data/urban\_sound\_8k/).

## 2. Step 2: Train Neural Network Model (train\_nn\_model.py)

This script trains a Neural Network model on the features extracted in Step 1.

python scripts/train\_nn\_model.py

What it does:

- Loads the feature data (ast\_features.pkl) and splits it into training and testing datasets.- Defines and trains a Neural Network (NN) model using the training data.- Saves the trained model and evaluation results.

## 3. Step 3: Train AST Model (train\_ast\_model.py)

This script trains the Audio Spectrogram Transformer (AST) model on the features extracted in Step 1.

python scripts/train\_ast\_model.py

What it does:

- Loads the feature data (ast\_features.pkl) and splits it into training and testing datasets.- Defines and trains an AST model using the training data.- Saves the trained model and evaluation results.

## 4. Step 4: Evaluate and Visualize Results (evaluate\_results.py)

This script evaluates the trained models and visualizes the results using confusion matrices, training curves, and sample predictions.

python scripts/evaluate\_results.py

What it does:

- Loads the saved predictions, true labels, and trained models.- Plots the following:- Confusion matrix and normalized confusion matrix.- Training and validation accuracy and loss curves.- Sample predictions for visual inspection.- Class distribution of the dataset.- Saves all visualizations in the /test/ directory.

## 5. Step 5: Save Evaluation Metrics (evaluation\_metrics.py)

This script generates and saves the classification report for both models.

python scripts/evaluation\_metrics.py

What it does:

- Loads the true labels and predictions.- Generates a classification report (e.g., precision, recall, F1-score).- Saves the classification report as a .txt file in the /test/ folder.

# Detailed Execution Sequence:

1. Run Feature Extraction Script:  
 - Ensure that you have the UrbanSound8K dataset in the correct folder.  
 - Execute the feature extraction script first:  
 python scripts/feature\_extraction.py  
  
2. Train Neural Network Model:  
 - Once the features are extracted, run the NN training script:  
 python scripts/train\_nn\_model.py  
  
3. Train AST Model:  
 - After training the NN model, run the AST model training script:  
 python scripts/train\_ast\_model.py  
  
4. Evaluate and Visualize Results:  
 - After both models are trained, run the evaluation script to visualize the results:  
 python scripts/evaluate\_results.py  
  
5. Generate Evaluation Metrics:  
 - Finally, run the metrics script to generate the classification report:  
 python scripts/evaluation\_metrics.py

# Saving and Retrieving Results:

- Model Files: Both models (NN and AST) will be saved in the /test/ folder.  
- Predictions: Predictions made by the models will be saved as y\_pred.npy in the /test/ folder.  
- Visualizations: All plots (confusion matrix, sample predictions, etc.) will be saved in the /test/ folder as images.

# Conclusion:

By following this sequence and running the respective Python scripts in the order outlined, you can preprocess the data, train the models, evaluate them, and visualize the results. This will allow you to compare the performance of the NN and AST models in classifying urban sound data from the UrbanSound8K dataset.