DEEP AUDIO CLASSIFIER

MAJOR PROJECT – I (AD-708)

Team Members

- Ankita Rajput (0187AD211006)
- Prachi Rajput (0187AD211031)
- Abhiraj Chouhan (0187AD211002)

Guide Details

- Prof. Dheeraj Namdev

Intro

Idea Solution/Prototype

The primary goal of this project is to build a deep learning model that can classify different urban sound events effectively. By leveraging the UrbanSound8K dataset, which provides a rich collection of labeled audio clips, the project integrates Natural Language Processing (NLP) for enhanced data processing and interpretation. The model architecture is designed to preprocess audio features, extract relevant patterns, and apply NLP techniques to improve the understanding and categorization of audio signals, thereby increasing classification accuracy and model robustness.

Abstract

This project develops a deep audio classification system to identify and categorize urban sounds using the UrbanSound8K dataset. It focuses on processing and analyzing sound clips such as sirens, street music, and engine noises. The project incorporates detailed audio signal analysis and data exploration to ensure balanced and comprehensive input for model training. The ultimate goal is to enhance the model's ability to accurately predict and classify different audio events, supporting applications in noise monitoring, smart city management, and accessibility tools.

Technology Stack

j

Requirements

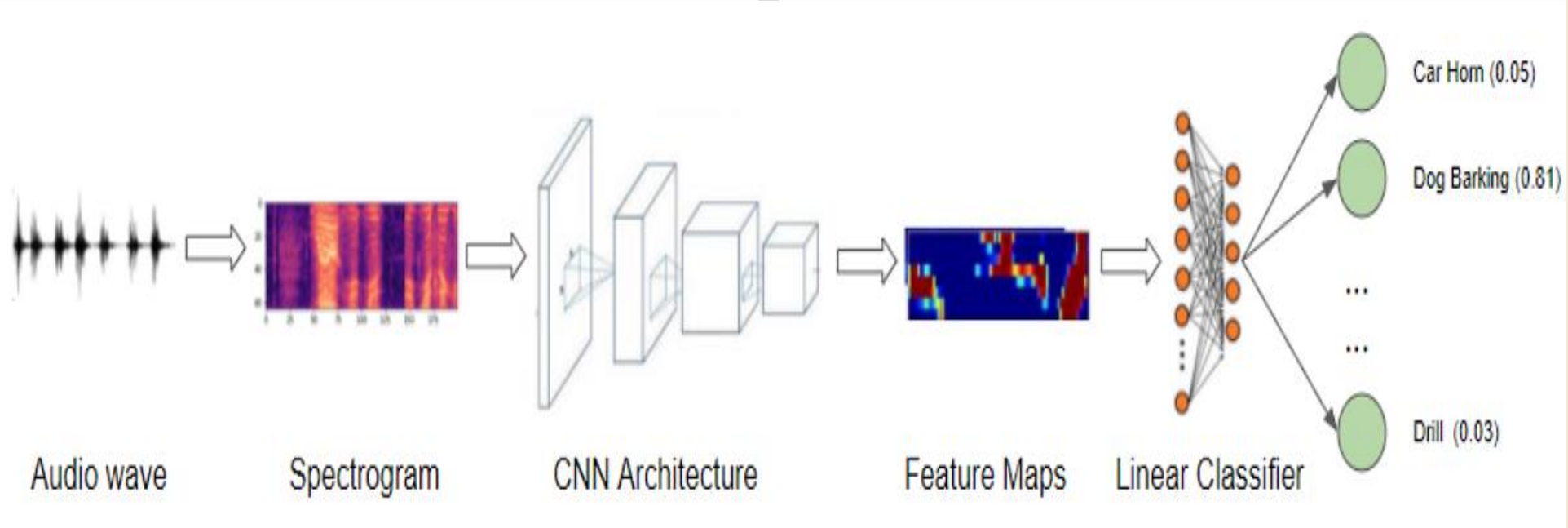
Functional Requirements

- Audio file upload.
- Classification of audio into predefined categories.
- Integration with NLP modules for preprocessing and feature extraction.
- User interface for prediction.

Non- Functional requirements

- High model accuracy and low latency.
- Robust error handling for various input formats.
- Scalability to adapt to larger datasets or real-world data streams.

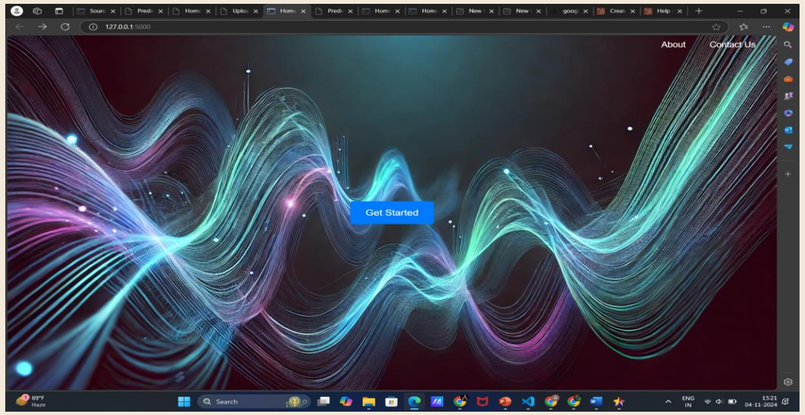
Flowchart Diagram

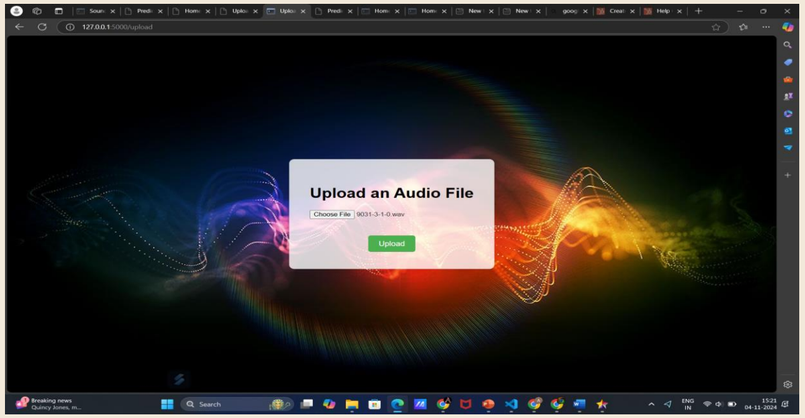


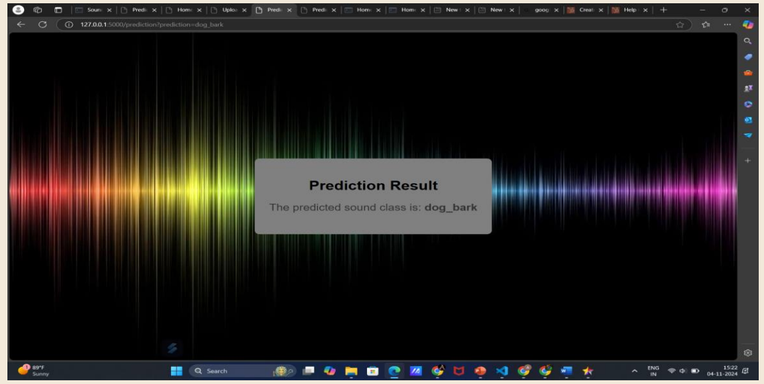
Deployment

- Hosting cost according to the servers selected.
- Domain Purchasing Cost.

Frontend Screenshots







Backend Screenshots

app.py 4 | UrbanSound.py 9+ | index.html | upload.html | UrbanSound.ipynb | prediction.html

UrbanSound.py > ...

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# %matplotlib inline

import warnings
warnings.filterwarnings("ignore")
# -

# !pip install librosa

import sys
print(sys.executable)


# !python -c "import librosa; print(librosa.__version__)"


import IPython.display as ipd
import librosa
import librosa.display

filename = 'UrbanSound8K/fold5/190893-2-0-11.wav'


# Librosa

#

librosa_audio_data,librosa_sample_rate = librosa.load(filename)


librosa_audio_data


librosa_sample_rate


librosa.display.waveshow(librosa_audio_data,sr=librosa_sample_rate)
ipd.Audio(filename)
```

app.py 4 | UrbanSound.py 9+ | index.html | upload.html | UrbanSound.ipynb | prediction.html

UrbanSound.py > ...

```
# Here Librosa converts the signal to mono, meaning the channel will alays be 1



# scipy

#

from scipy.io import wavfile as wav
wave_sample_rate,wave_audio = wav.read(filename)

wave_audio

wave_sample_rate


plt.figure(figsize=(12,4))
plt.plot(wave_audio)
plt.show()

# Here Scipy converts the signal to stereo, meaning the channels will be 2¶
#



# Metadata


metadata = pd.read_csv('UrbanSound8K.csv')
metadata.head()

metadata.shape


metadata.isnull().sum()


metadata.duplicated().sum()
```

THANK YOU