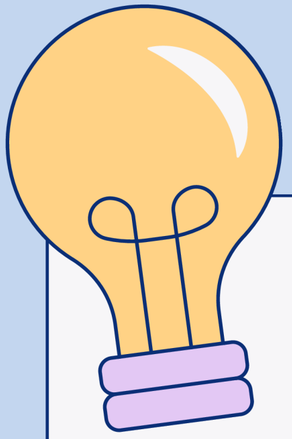


MINI PROJECT

Enviromental Sound Classification Using CNN

Mentees

- ***Rohan Garg***
- ***Rupam Das***
- ***Anikeit Khanna***
- ***Ayush Kumar***

Mentors

- Arin Dhariwal
- Deham Rajvanshi
- Divyam Agarwal
- Shreyansh Dwivedi

Introduction

- In this project, we developed a deep learning model for classifying urban sound events using the UrbanSound8K dataset.
- The primary goal was to build an efficient Convolutional Neural Network (CNN) model that can accurately classify different sound categories.
- The project involved feature extraction, data preprocessing, hyperparameter tuning, and evaluation of the trained model.

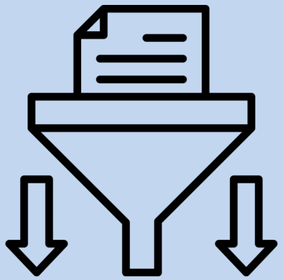
DATASET USED

The dataset used is UrbanSound8K, which contains 8,732 labeled sound excerpts (≤ 4 seconds long) from 10 different classes of urban sounds:

- **Air Conditioner**
- **Car Horn**
- **Children Playing**
- **Dog Bark**
- **Drilling**
- **Engine Idling**
- **Gunshot**
- **Jackhammer**
- **Siren**
- **Street Music**

Each sound clip is provided in .wav format along with a CSV metadata file that includes file name, class label, and fold assignment (used for cross-validation).

Feature Extraction Process



To convert raw audio into a format suitable for deep learning, we extracted **Mel-Frequency Cepstral Coefficients (MFCCs)**, which are widely used in speech and audio recognition tasks.

Steps Involved:

- **Load the audio file using librosa.**
- **Extract MFCC features (40 coefficients per frame).**
- **Take the mean across all frames to obtain a fixed-size feature vector.**
- **Store features and labels in a structured format.**

Hyperparameter Tuning and Model Training

We built a 1D CNN model to classify the extracted MFCC features. The network consists of convolutional layers, max-pooling layers, batch normalization, dropout, and fully connected layers.

Hyperparameter Tuning:

We experimented with different hyperparameters:

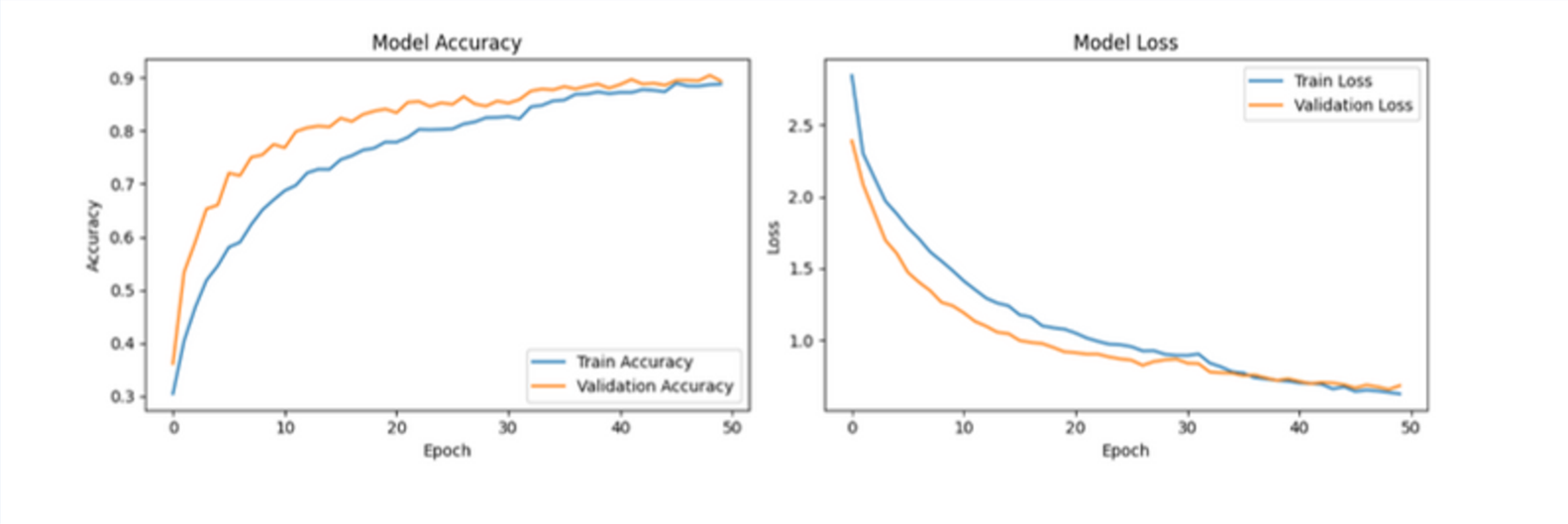
- Number of filters: 64, 128, 256
- Kernel size: 3
- Pooling size: 2
- Dropout rates: 0.3 to 0.5
- Regularization: L2 (0.001)
- Optimizer: Adam
- Batch size: 64
- Epochs: 50

Final CNN Architecture

```
from tensorflow.keras.regularizers import l2

model = Sequential([
    # First Convolutional Block
    Conv1D(64, 3, activation='relu', input_shape=(40, 1), kernel_regularizer=l2(0.001)),
    MaxPooling1D(2),
    BatchNormalization(),
    Dropout(0.3),

    # Second Convolutional Block
    Conv1D(128, 3, activation='relu', kernel_regularizer=l2(0.001)),
    MaxPooling1D(2),
    BatchNormalization(),
    Dropout(0.3),

    # Third Convolutional Block
    Conv1D(256, 3, activation='relu', kernel_regularizer=l2(0.001)),
    MaxPooling1D(2),
    BatchNormalization(),
    Dropout(0.4),

    # Fully Connected Layers
    Flatten(),
    Dense(256, activation='relu', kernel_regularizer=l2(0.001)),
    Dropout(0.5),
    Dense(len(label_encoder.classes_), activation='softmax')
])

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

Training Strategy

Loss function: Categorical Cross-Entropy
Optimizer: Adam
Callbacks used:
Early Stopping (stops training when validation loss doesn't improve)
ReduceLROnPlateau (reduces learning rate when performance stagnates)



Observations and Results

Performance Metrics

After training, the model achieved the following performance on the test set:

- **Test Loss: 0.6332206130027771**
- **Test Accuracy: 90.15056490898132%**

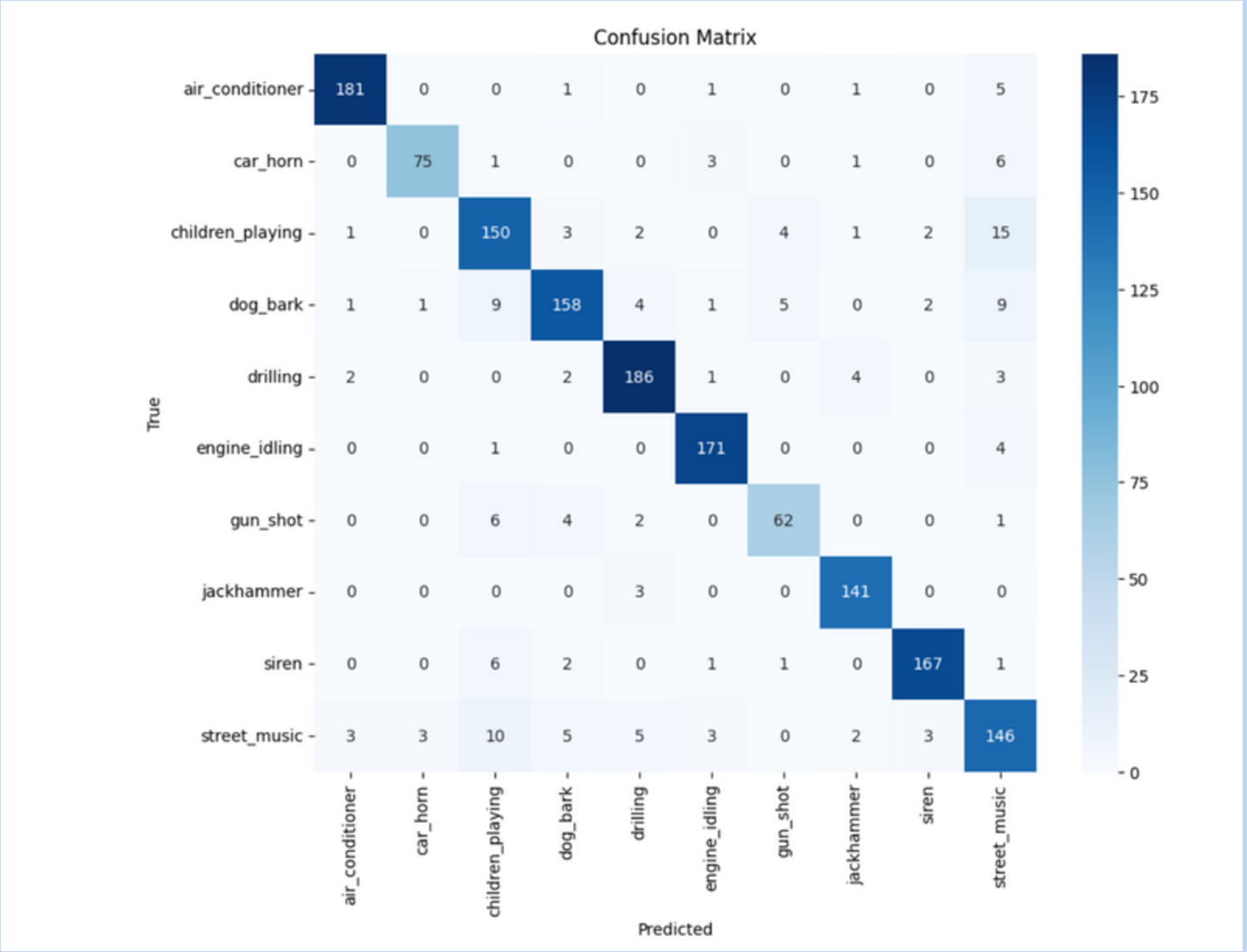
Confusion Matrix & Classification Report

Key Observations:

Diagonal Dominance: The strong diagonal indicates that most predictions are correct.

Misclassifications:

- "Children Playing" has been confused with "Street Music" **15 times**.
- "Dog Bark" has been misclassified as "Children Playing" **9 times**.
- "Gun Shot" has been confused with "Children Playing" and "Dog Bark."
- "Street Music" has been **misclassified the most** across multiple categories**.**



Predictions on New Audio Files

	precision	recall	f1-score	support
air_conditioner	0.96	0.96	0.96	189
car_horn	0.95	0.87	0.91	86
children_playing	0.82	0.84	0.83	178
dog_bark	0.90	0.83	0.87	190
drilling	0.92	0.94	0.93	198
engine_idling	0.94	0.97	0.96	176
gun_shot	0.86	0.83	0.84	75
jackhammer	0.94	0.98	0.96	144
siren	0.96	0.94	0.95	178
street_music	0.77	0.81	0.79	180
accuracy			0.90	1594
macro avg	0.90	0.90	0.90	1594
weighted avg	0.90	0.90	0.90	1594

Conclusion

This project successfully implemented an audio classification system using a CNN model trained on MFCC features from the UrbanSound8K dataset. The model achieved 90% accuracy and demonstrated strong performance in recognizing distinct urban sounds

THANKS FOR LISTENINIG :)