Project Report: Urban Audio Intelligence

Comparative Analysis of CNN, LSTM, and Random Forest for UrbanSound8K Classification

Pratyaksha Jha
Roll Number: 240150025
B.Tech - Data Science and Artificial Intelligence
Indian Institute of Technology (IIT), Guwahati

January 8, 2026

Abstract

This project develops an urban noise classification system using the UrbanSound8K dataset. By transforming raw audio into Mel-Spectrograms, we evaluate CNN, LSTM, and Random Forest architectures. Results demonstrate that CNNs provide the highest accuracy by effectively capturing spatial-spectral features in the audio data.

Contents

1 Introduction

As urban density increases, the acoustic environment becomes a critical factor in city planning and public safety. Urban sounds are highly varied, containing both stationary noises (e.g., air conditioners) and impulse noises (e.g., gunshots). Manual classification of these sounds is labor-intensive and impractical for real-time applications.

This project aims to automate the detection of 10 specific urban sound classes. By converting these sounds into the visual domain via Mel-Spectrograms, we treat the audio classification problem as a pattern recognition task, allowing for the application of Deep Learning architectures.

2 Motivation and Problem Statement

The primary motivation for this project stems from the need for automated environmental monitoring. Specific use cases include:

- **Smart City Infrastructure:** Real-time noise pollution mapping.
- **Security:** Automated alerts for high-stress sounds like sirens or gunshots.
- **Industrial Monitoring:** Detecting equipment failure through sound patterns.

Problem Statement: Build an intelligent sound recognition system that identifies different urban noises using audio ML models to help cities monitor and control noise levels more effectively.

3 Theoretical Framework

3.1 Mel-Spectrogram Generation

Digital audio is typically represented as a 1D time-series. However, machine learning models often perform better when features are extracted in the frequency domain.

1. **Short-Time Fourier Transform (STFT):** We apply a sliding window to the audio, calculating the frequency spectrum for each window.
2. **Mel Scale Transformation:** Frequencies are mapped to the Mel scale, a perceptual scale of pitches judged by listeners to be equal in distance from one another.

$$m = 2595 \log_{10} \left(1 + \frac{f}{700}\right)$$

This transformation captures the spectral "texture" of the sound, which is essential for distinguishing between sounds with similar frequencies but different rhythms, like a jackhammer and a drill.

3.2 Architectural Theory

Convolutional Neural Networks (CNN): CNNs are designed to learn hierarchical features. In this project, the 2D filters in the convolutional layers detect "edges" and "textures" in the spectrogram, which correspond to specific frequency harmonics and temporal beats.

Long Short-Term Memory (LSTM): LSTMs are a type of Recurrent Neural Network (RNN) capable of learning long-term dependencies. Audio is inherently sequential; an LSTM processes the spectrogram one time-step at a time, maintaining a "memory" of previous frequencies to identify evolving sounds like a rising siren.

4 Methodology

4.1 Data Characteristics and Preprocessing

The UrbanSound8K dataset contains 8,732 labeled sound excerpts.

- **Standardization:** All files are resampled to 22.05 kHz. To ensure uniform input for the neural networks, clips are padded or truncated to exactly 4.0 seconds ($88,200$ samples).
- **Feature Extraction:** We utilized a window size of 2048 and a hop length of 512, resulting in a 128x173 feature matrix for each clip.
- **Parallelization:** Using `joblib.Parallel`, we reduced preprocessing time by over 70% by distributing feature extraction across all available CPU cores.

4.2 CNN Implementation Details

The implemented CNN follows a sequential architecture:

1. **Conv2D Layer:** 32 filters (3x3), ReLU activation.
2. **Max Pooling:** 2x2 window to reduce dimensionality.
3. **Batch Normalization:** To stabilize learning and accelerate convergence.
4. **Flattening:** Transitioning from spatial features to a 64-unit Dense layer.
5. **Softmax Output:** 10 units representing class probabilities.

4.3 LSTM Implementation Details

The LSTM model treats the 173 time-steps as the sequence length and 128 Mel-bands as features. It consists of two stacked LSTM layers (128 and 64 units) with Dropout layers (0.3) to prevent overfitting on the training set.

5 Results and Observations

5.1 Quantitative Performance

During testing, the LSTM model consistently outperformed the others.

Model	Training Acc	Test Acc	Precision	Recall
CNN	84.5%	74.8%	0.77	0.75
LSTM	92.5%	81.8%	0.82	0.82
Random Forest	99%	72.2%	0.73	0.72

Table 1: Comparison of Model Evaluation Metrics

5.2 Observations and Error Analysis

A key observation from the **Confusion Matrix** was the confusion between "Drilling" and "Jackhammer." Spectrally, these sounds are very similar, both exhibiting high-energy broadband noise. Conversely, "Gun Shot" had the highest precision due to its unique impulsive nature in the spectrogram (a vertical spike).

6 Future Scope

While the current implementation achieves high accuracy, several avenues exist for future enhancement:

- **Data Augmentation:** Techniques such as time-stretching, pitch-shifting, and adding background white noise could be implemented to make the model more robust to real-world variances.
- **Transfer Learning:** Leveraging pre-trained models like *VGGish* or *YAMNet* (trained on millions of YouTube audio clips) could significantly improve the classification of rare urban sounds.
- **Real-time Deployment:** Optimizing the model using *TensorFlow Lite* for deployment on edge devices like Raspberry Pi or mobile applications for live environmental monitoring.

7 Conclusion

The project successfully demonstrates that visual-spectral representation is a robust method for environmental audio classification. While Random Forests provide a quick baseline, they fail to capture the temporal-spectral relationships that Deep Learning models like CNNs exploit. The CNN architecture proved most effective, balancing computational efficiency with high accuracy.

8 References

1. Salamon, J., Jacoby, C., and Bello, J. P. (2014). *A Dataset and Taxonomy for Urban Sound Research.*
2. Andrew Ng - DeepLearningAI, "Neural Networks and Deep Learning," YouTube Playlist. Available here.
3. Valerio Velardo - The Sound of AI, "Audio Signal Processing for Machine Learning," YouTube Playlist. Available here.