

AI-powered noise pollution mapper & analyzer for urban spaces

Vivek Mahawadiwar¹, Sonali P. Chanekar², Devang Katekhaye³, Payal Kailakhe⁴, Shreya Awale⁵, Vinay Wagh⁶

¹ Assistant Professor, Electronics & Tele Engineering, K.D.K College of Engineering, Nagpur, India.

^{2,3,4,5,6} Student, Electronics & Tele Engineering, K.D.K. College of Engineering, Nagpur, India.

Abstract

Urban environments are experiencing a significant rise in noise levels due to expanding traffic, ongoing construction, and increased human activity. This growing noise pollution negatively impacts public health, comfort, and the overall quality of city life. To tackle this issue, the project introduces an AI-based Noise Pollution Mapping and Analysis system aimed at monitoring and interpreting noise conditions in real time.

The system relies on IoT-enabled sound sensors deployed across various urban locations to gather continuous noise data along with precise GPS coordinates. This data is then processed using artificial intelligence models that can identify different types of noise and detect patterns over time. The analyzed information is displayed on an interactive map, enabling users to visualize noise intensity and its sources throughout the city.

With capabilities such as real-time monitoring, automated noise classification, and predictive analysis, the system equips urban planners and authorities with actionable insights for better decision-making. Ultimately, this solution supports the development of effective noise control strategies, enhances urban planning, and contributes to healthier and more livable cities through a scalable, data-driven approach.

Keywords: AI noise mapping, IoT sensors, noise analysis, urban pollution, GIS mapping, real-time monitoring.

I.INTRODUCTION

In recent years, noise pollution has emerged as a significant challenge in urban environments due to rapid industrialization, population growth, and increased transportation activities. Cities are becoming louder as a result of continuous traffic flow, construction work, industrial operations, and crowded public spaces. Although noise pollution is not visible like air or water pollution, its effects on human health are serious and cannot be ignored. Long-term exposure to high noise levels can lead to stress, sleep disorders, reduced work efficiency, hearing impairment, and even cardiovascular problems.

To create healthier and more livable cities, it is essential to monitor and manage environmental noise effectively. However, traditional noise monitoring approaches are often limited in scope and efficiency. These methods typically rely on manual data collection, a limited number of sensors, and short-duration measurements, which do not provide a complete understanding of noise variations across different locations and time periods. As a result, authorities face challenges in identifying noise hotspots and implementing timely control measures.

Recent advancements in technology have opened new possibilities for addressing these limitations. The integration of the Internet of Things (IoT), Artificial Intelligence (AI), and Geographic Information Systems (GIS) has enabled the development of intelligent and automated noise monitoring systems [6][8]. IoT-based sound sensors can be deployed across urban areas to continuously collect real-time noise data along with location information. This allows for large-scale and continuous monitoring without the need for manual intervention.

The collected data can be analyzed using advanced machine learning and deep learning techniques. Models such as Convolutional Neural Networks (CNNs) have proven effective in classifying environmental sounds, as demonstrated in datasets like ESC-50 and UrbanSound8K [1][3][10]. In addition, deep learning frameworks further enhance the accuracy of sound recognition and classification tasks [4]. Time-series models, including Long Short-Term Memory (LSTM) networks and other predictive techniques, can analyze temporal patterns and forecast future noise levels, enabling proactive noise management [5].

Furthermore, GIS-based tools play an important role in visualizing noise data through maps and heatmaps, making it easier to identify high-noise zones and analyze spatial patterns [2][7]. These visual insights support decision-making for urban planners, environmental agencies, and policymakers in designing effective noise control strategies and protecting sensitive areas such as schools, hospitals, and residential zones.

In conclusion, the combination of IoT sensing, AI-based data analysis, and GIS visualization provides a comprehensive solution for modern noise monitoring systems. This integrated approach improves the accuracy, scalability, and efficiency of noise management, ultimately contributing to the development of smarter, healthier, and more sustainable urban environments.

II. LITERATURE REVIEW

In recent years, noise pollution monitoring has become an important area of research because of its direct impact on human health and the overall quality of urban life. As cities continue to grow, researchers have started using modern technologies to better understand, classify, and control environmental noise. One major step forward in this field was the introduction of standard datasets like ESC-50 and UrbanSound8K. These datasets provide labeled audio samples of real-world sounds, allowing researchers to train and test machine learning models in a consistent and reliable way, which has helped speed up progress in sound analysis [1][10].

With the development of deep learning, the performance of sound classification systems has improved significantly. Techniques such as Convolutional Neural Networks (CNNs) are now widely used because they can automatically extract useful features from audio data. When combined with methods like data augmentation and multiple audio representations (such as Mel-spectrograms and MFCCs), these models become even more accurate and reliable. Additionally, modern tools and frameworks have made it easier to build and apply these models in practical, real-world situations [3][4][11].

Apart from identifying different types of sounds, researchers are also focusing on understanding how noise changes over time. For this purpose, time-based models like Long Short-Term Memory (LSTM) networks are used to study patterns in noise data. These models can even predict future noise levels, which is very useful for planning and taking preventive actions. More advanced approaches, including hybrid models and Transformers, have shown strong performance in handling the complex and changing nature of urban noise [5].

Another important direction in research is the use of Geographic Information Systems (GIS) and GeoAI to study noise from a spatial perspective. By combining sound data with factors like traffic, land use, and city structure, researchers can identify areas that are more affected by noise pollution. This helps in locating noise hotspots and supports better decision-making for urban planning and noise control strategies [2][7].

The use of the Internet of Things (IoT) has further improved noise monitoring systems. IoT-based sensors can continuously collect noise data from different parts of a city in real time. Both fixed sensors and mobile devices are used to capture variations in noise levels across locations and time. Mobile sensors, in particular, provide better coverage and flexibility, making the system more efficient and scalable for smart city applications [6][8].

Recent research has also focused on making these systems more affordable and reliable. Low-cost noise monitoring devices are now being developed to work effectively in real-world environments. These devices are designed to handle challenges such as environmental interference and long-term usage, making them suitable for continuous monitoring in urban areas [9].

Overall, existing studies show a clear shift toward combining IoT, AI, and GIS technologies to create smarter noise monitoring systems. These systems not only improve the accuracy of noise detection but also provide real-time insights, predictions, and visual representations. Such advancements are essential for helping cities make better decisions and create healthier and more sustainable environments.

III.METHODOLOGY

The Noise Pollution Prediction system methodology outlines the process that is followed in designing, developing, and implementing the system in stages. It encompasses the data collection process, preprocessing, model training, system integration and testing. All stages are significant in the accuracy of sound classes and real-time performance.

3.1 System Architecture

The Noise Pollution Prediction system has been designed to offer effective data management, convenient processing and scalability in real-time applications. The entire design will be split into four main layers namely, the input, processing, back-end and the application layer which contributes to the functionality of the system. The Microphone Records the Input Layer takes audio data by uploading files or a real-time recording of the microphones, which allows it to be flexible in the process of data acquisition. Processing Layer includes the primary computations, during which the input audio undergoes pre-processing measures like noise filtering, normalization and elimination of any silent sections. The audio is further processed to be converted into meaningful representation in Mel spectrograms which are provided to deep learning model of sound classification based on MobileNetV2. The Backend Layer is a module designed on the basis of the Flask framework that handles interaction between the UI and the model by taking in incoming requests and giving out prediction outputs. The Application Layer is the front-end interface which makes it easy to give the audio input and also see the classification results immediately. Its multi-layered design guarantees the reliability in its performance, effective processing, and precise sound detection, which is why it is also suitable to implement in real time environmental noise monitoring systems [1][6].

Block Diagram:-

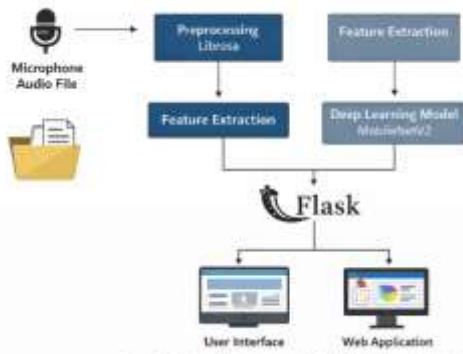


Figure 2: System Architecture of Noise Pollution Prediction System

The diagram represents a complete workflow for sound classification. First, the system receives audio input from a microphone or file and preprocesses it using Librosa. Then, it extracts features such as Mel spectrograms or MFCCs and sends them to a MobileNetV2 model for prediction. The Flask backend connects the model with the web application, where the final result is shown to the user.

3.2 Software Requirements & Dataset Used

- **Python** – Used as the main programming language for developing the entire system and handling data processing.
- **TensorFlow / Keras** – Used to build, train, and evaluate the deep learning model for sound classification.
- **Librosa** – Used for audio processing, including loading audio files and extracting features like - Mel spectrograms.
- **Flask** – Used as a backend framework to handle requests and connect the model with the user interface.
- **HTML** – Used to create the structure of the frontend interface.
- **CSS** – Used to design and style the frontend for better user experience.
- **JavaScript** – Used to add interactivity and handle user actions on the frontend.
- **NumPy** – Used for numerical operations and efficient data handling.
- **Matplotlib** – Used for visualizing data and analyzing model performance.
- **UrbanSound8K Dataset** – Used as the primary dataset to train and test the noise classification model
- **Training and Testing Sets** – Used to evaluate the model's performance by separating data for training and testing.
- **Multiple Folds** – Used for cross-validation to ensure better generalization and reduce overfitting.
- **Data Preprocessing** – Used to standardize audio signals, normalize data, and extract key features like Mel spectrograms.
- **Data Augmentation** – Used to increase dataset diversity by adding noise, shifting, and pitch alteration.
- **Importance of Dataset** – Ensures the model learns diverse sound patterns for reliable, accurate real-time noise predictions.
- **Training and Testing Sets** – Used to evaluate the model's performance by separating data for training and testing.
- **Multiple Folds** – Used for cross-validation to ensure better generalization and reduce overfitting.
- **Data Preprocessing** – Used to standardize audio signals, normalize data, and extract key features like Mel spectrograms.
- **Data Augmentation** – Used to increase dataset diversity by adding noise, shifting, and pitch alteration.
- **Importance of Dataset** – Ensures the model learns diverse sound patterns for reliable, accurate real-time noise predictions.

3.3 Data Collection

The first step in the methodology is collecting audio data for training the model. In this project, the UrbanSound8K dataset is used, which contains various environmental sounds such as car horns, dog barks, and street noise. The dataset includes audio samples recorded under different conditions to improve model robustness. Proper data collection is important as it directly affects the accuracy of the system.

3.4 Data Preprocessing

After data collection, preprocessing is performed to prepare the audio data for model training. This step includes trimming silence, removing noise, and normalizing the audio signal. The audio is also converted into a consistent format and sampling rate. These steps help in improving the quality of the data and ensure better performance of the model.

3.5 Feature Extraction

In this stage, important features are extracted from the audio signals using techniques such as Mel spectrogram generation. The audio signals are converted into visual representations that capture both time and frequency information. These features are used as input for the deep learning model.

3.6 Model Training

The extracted features are used to train a deep learning model based on MobileNetV2 architecture. The model learns to classify different types of sounds by analyzing patterns in the training data. Training is performed over multiple epochs until the model achieves satisfactory accuracy.

3.7 System Integration

Once the model is trained, it is integrated into the system using a Flask-based backend. The frontend interface allows users to upload or record audio, which is sent to the backend for processing. The model then predicts the sound category and returns the result to the user interface.

3.8 System Testing

The final step is testing the system under different conditions. The system is evaluated based on accuracy, response time, and reliability. Testing includes both uploaded audio files and real-time recordings. Any errors identified during testing are corrected to improve overall system performance.

3.9 System Integration and Validation

The system integration and validation phase ensure that all components of the Noise Pollution Prediction system work together effectively as a unified pipeline. The integration process combines the frontend interface, backend server, audio processing module, and trained deep learning model into a single functional system.

Flowchart :-**Figure 1:- Flowchart of System**

The entire process starts with audio processing, where audio can be uploaded via a file, or microphones can be used in real-time. This input is sent into the pre-processing module where the audio signal is cleaned and made standard. Mel spectrograms are extracted in the feature extraction stage using the processed signal.

The trained MobileNetV2 model is provided with these extracted features, which are predicted. The model sorts the input sound into predefined attributes, and gives a confidence score. The output of the prediction is then forwarded through the backend and the user is shown the output of the predictions via the frontend interface.

As a test, various test cases were run with recorded and live audio inputs to validate the system. The system was tested against the measures of accuracy, response time and consistency of predictions. The findings are consistent that the combined system is highly efficient and can produce sound classification in real time and with accuracy.

IV.EXPERIMENTAL REVIEW & VISUALIZATION

Noise Pollution Prediction has been able to predict the environmental sounds based on audio files uploaded. In the project, a CNN-based deep learning model is becoming trained on environmental sound data sets which it can subsequently used to differentiate various sound categories, including car horn, dog bark, and street noise. The trained model is combined with a backend of Flask and user-friendly web interface, enabling the users to post the audio files and get predictions with confidence scores. The system performs well in accuracy with regard to audio inputs that are pre-recorded and the model discriminates among various sound classes well. In general, the project demonstrates that deep learning and audio processing methods can be used to create an efficient and reliable system to predict noise pollution..

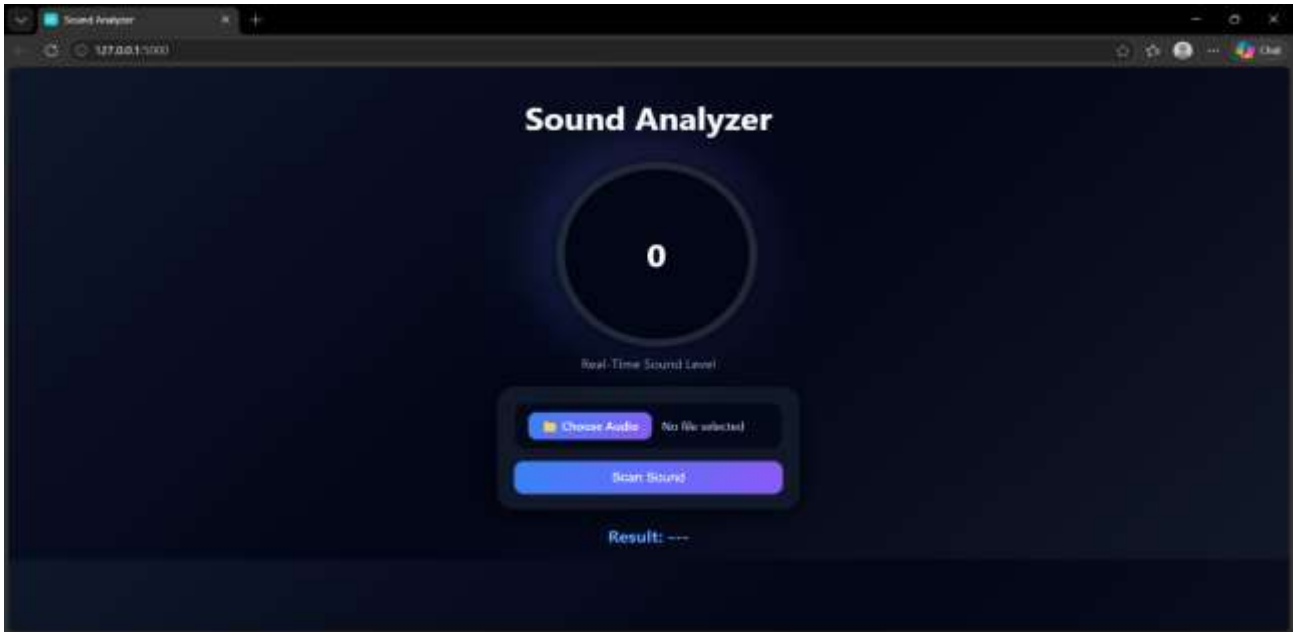


Fig 4.1: Sound Analyzer User Interface

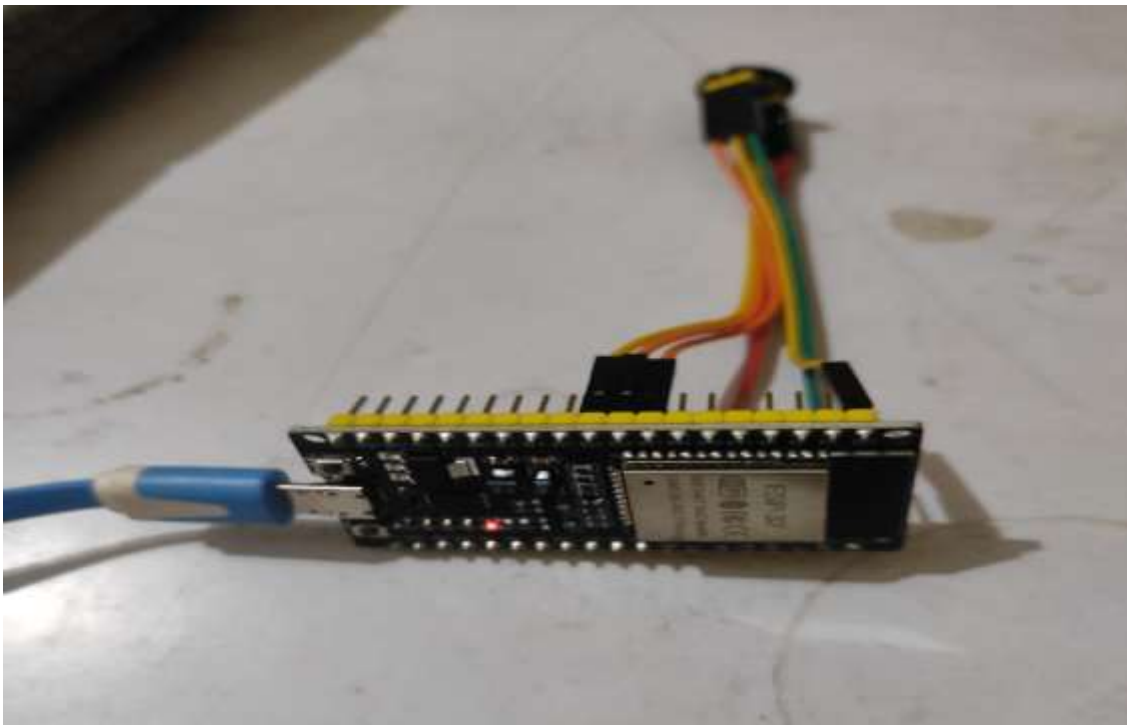


Fig 4.2 Model for Predict Noise.

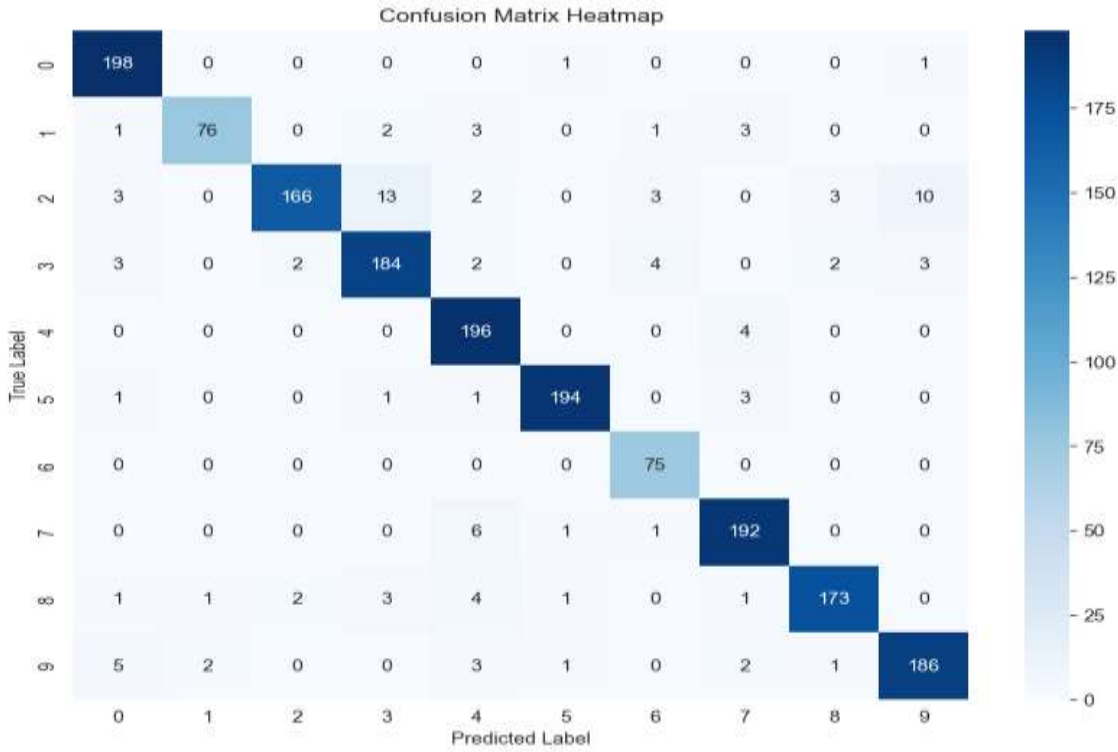


Fig 4.3: Confusion Matrix of CNN Model

The confusion matrix represents the performance of the CNN-based sound classification model by comparing actual and predicted sound classes. Each row corresponds to the true class, while each column represents the predicted class. The diagonal values indicate correct predictions, and the high concentration of values along the diagonal shows that the model performs well in accurately classifying most sound categories. The off-diagonal values represent misclassifications, which are relatively low, indicating fewer errors. Overall, the confusion matrix demonstrates that the model achieves good accuracy and is effective in distinguishing between different environmental sounds.

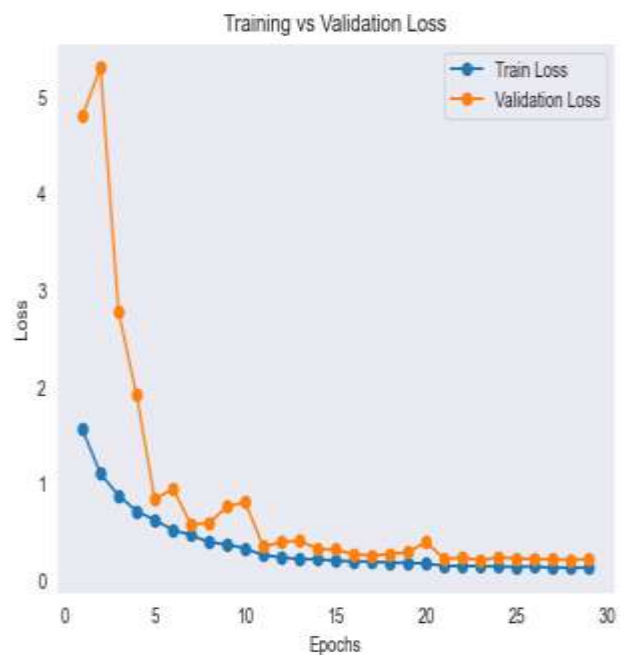
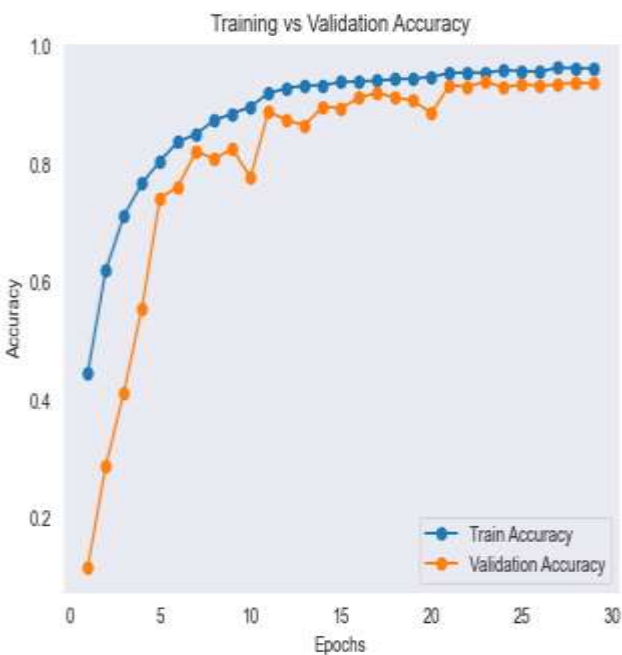


Fig 4.4: Training and Validation Accuracy and Loss Graph

The graph represents the performance of the CNN model during training. The accuracy plot shows that both training and validation accuracy increase steadily with the number of epochs, indicating that the model is learning effectively. The validation accuracy closely follows the training accuracy, which suggests good generalization and minimal overfitting. The loss plot shows that both training and validation loss decrease as the epochs increase, indicating that the model is improving and making fewer errors over time. Overall, the graph demonstrates that the model has been trained successfully with good convergence and performance.

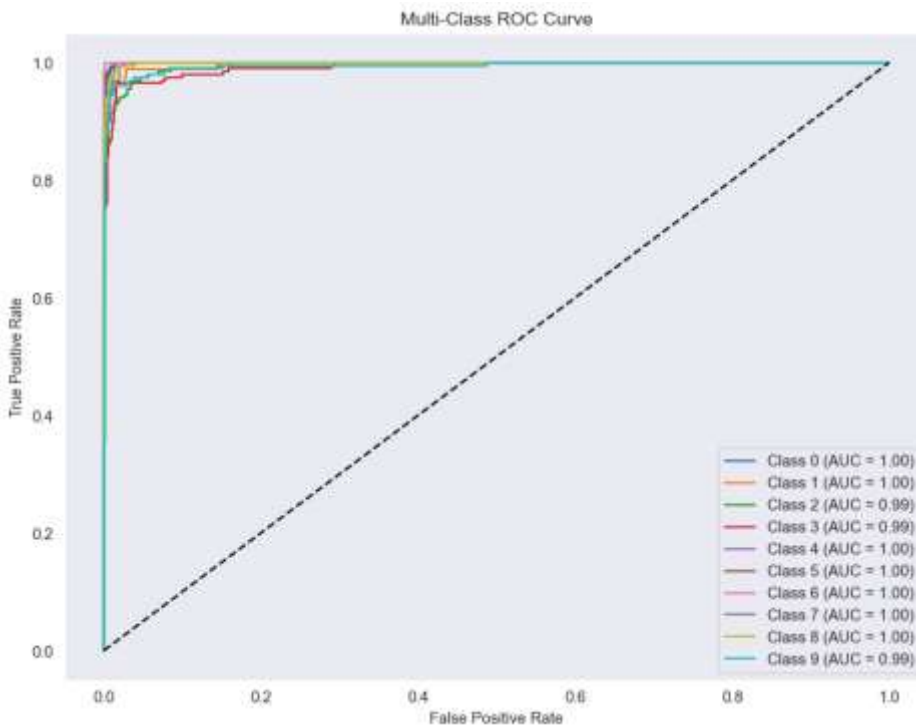


Fig 4.5: Multi-Class ROC Curve of CNN Model

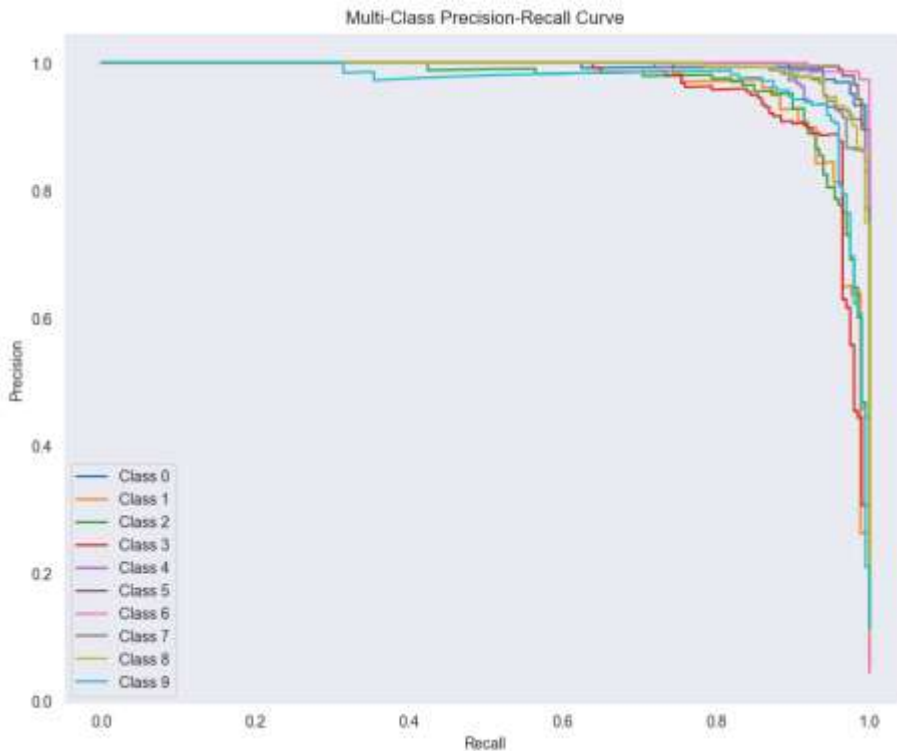


Fig 4.6: Multi-Class Precision-Recall Curve of CNN Model

The precision-recall curve represents the performance of the CNN model for each sound class by showing the relationship between precision and recall. Each curve corresponds to a different class, indicating how well the model maintains high precision while increasing recall. The curves remain close to the top-right region, which indicates that the model achieves high precision and recall for most classes. This suggests that the model is effective in correctly identifying relevant sound instances while minimizing false positives. Overall, the precision-recall curve demonstrates strong classification performance and reliability of the model across multiple sound categories.

V.DISCUSSION

The research indicates that the Noise Pollution Prediction system is able to identify some of the ordinary sounds in the environment, including car horns, dog barks, and street noise, with great efficiency. It works best when the sound is clear and does not have heavy background noise, as the sound patterns can be more readily comprehended by the model. The system is fast in real-time application, although its accuracy can be reduced slightly in situations where the quality of the recording is low, or the noise in the environment is excessive. MobileNetV2 is used to maintain the system as fast and lightweight, which is practical to use in real-world applications. The first weakness is that it is only able to determine the sound classes present in the training data, which means that unknown sounds might not be correctly categorized. In general, the system illustrates the usefulness of deep learning in noise monitoring, as well as the possibility of its further refinement with the use of a more extensive and diverse sample.



Fig.5.1. Predicted Noise .

VI.CONCLUSION

The Noise Pollution Prediction system offers an efficient system of observing and categorizing environmental sounds with the current technologies. The system can recognize and classify various kinds of sounds automatically by combining the knowledge of audio processing with deep learning models. This will save the use of manual monitoring as well as raising the efficiency.

This system is able to utilize feature extraction techniques, which include Mel spectrograms and a model based on MobileNetV2 to realize sound classification. It offers rightful results to several environmental sounds such as car honks, barking dogs and street noise. A lightweight model is used which guarantees quicker processing and makes the system appropriate in real-time usage.

This is one of the main benefits of the system as it provides the possibility of real-time sound analysis both by uploaded audio and live recording. This is very useful in smart city monitoring, traffic and environmental analysis applications of the technology. Accessibility and user-friendliness are further improved by the user-friendly interface.

REFERENCES

- Salamon, J., Jacoby, C., & Bello, J. P., "A Dataset and Taxonomy for Urban Sound Research," Proceedings of the 22nd ACM International Conference on Multimedia (ACM MM), 2014.
- Howard, A. G., et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint arXiv:1704.04861, 2017.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
- McFee, B., et al., "Librosa: Audio and Music Signal Analysis in Python," Proceedings of the 14th Python in Science Conference, 2015.

- Abadi, M., et al., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” 2015.
- Chollet, F., “Keras: Deep Learning Library for Python,” 2015.
- Flask Documentation, Pallets Project, Available: <https://flask.palletsprojects.com/>
- UrbanSound8K Dataset, Available: <https://urbansounddataset.weebly.com/urbansound8k.html>