ISSN NO: 9726-001X

Volume 13 Issue 02 2025

JOURNAL OF CURRENT SCIENCE

A COMPREHENSIVE DATASET FOR URBAN SOUND CLASSIFICATION AND ANALYSIS

Sudi Shylaja,
UG Student,
Department of CSE,
St. Martin's Engineering College,
Secunderabad, Telangana, India
sudishylaja1@gmail.com

Mrs. P. Devasudha,
Associate Professor,
Department of CSE,
St. Martin's Engineering College,
Secunderabad, Telangana, India
drjawaherlalcse@smec.ac.in

***Abstract:** The increasing complexity of urban surroundings and the impact of sound pollution on quality of life have made urban sound classification a critical field of study. Early sound analysis focused on identifying traffic noise and industrial activity using manual observations and simple acoustic monitoring techniques. Subjective sound classification by analysts was inconsistent and lacked scalability, making urban sound management difficult. This project aims to generate a comprehensive urban sound classification dataset to help identify and analyze urban noises and train machine learning models more accurately. This research A Comprehensive Dataset for Urban Sound Classification and Analysis is highlights the need for a strong data collection for sound categorization. Manual analysis and subjective interpretation limit urban sound classification approaches, limiting data collecting and policymaking. This research is driven by the need for objective sound pollution management approaches as urbanization accelerates. Researchers and policymakers will benefit from a comprehensive dataset on urban soundscapes. Using machine learning, the proposed method develops algorithms that accurately characterize urban sounds. These models can find patterns and features in sound recordings that traditional methods miss by examining big datasets. Real-time sound monitoring will improve urban planning and management to reduce noise pollution. To help researchers and city planners understand urban soundscapes, the study seeks nuance. Using machine learning to classify urban sounds is more efficient, scalable, and objective than traditional methods, improving urban living conditions and quality of life. The rapid expansion of urban areas has led to increased noise pollution, negatively affecting public health and overall quality of life. Effective urban sound classification is essential for monitoring and managing noise pollution, yet traditional methods rely on manual observations and subjective interpretations, limiting scalability and accuracy. By utilizing machine learning, the study ensures a scalable, efficient, and data-driven framework for sound classification, ultimately contributing to sustainable urban development and enhanced public well-being.*

***Keywords:** Urban sound classification, noise pollution, machine learning, sound analysis, urbanization.*

I. INTRODUCTION

Urban soundscapes form an essential aspect of city life, shaping both the daily experiences of residents and the functionality of urban spaces. The sounds of a city—ranging from traffic noise, construction activity, and public transportation to more subtle elements such as the chatter of pedestrians, the rustling of leaves in parks, or distant sirens—create a complex and dynamic acoustic environment. These soundscapes influence not only the quality of life but also urban planning, public health, and environmental sustainability. As cities continue to expand and evolve, understanding and managing urban sounds become increasingly critical for maintaining livable and efficient urban environments.
The classification and analysis of urban soundscapes are essential for multiple applications, including noise pollution monitoring, smart city development, urban planning, public safety, and even psychological well-being. For example, excessive noise pollution has been linked to increased stress levels, sleep disturbances, and cardiovascular issues, making its mitigation a public health priority. Additionally, urban sound analysis plays a crucial role in improving security systems, traffic management, and the overall design of public spaces to enhance comfort and usability. However, despite advancements in machine learning and audio processing, accurately classifying urban sounds remains a significant challenge. This difficulty arises due to the intricate and overlapping nature of city noises, which often exhibit variations in frequency,

amplitude, and temporal characteristics. Unlike controlled environments where sound classification is relatively straightforward, urban settings present an ever-changing and often unpredictable mix of sounds. Traditional methods of sound classification frequently struggle to cope with these complexities, leading to inconsistent or inaccurate results. Factors such as background noise, reverberation, and the transient nature of many urban sounds further complicate the process. To address these challenges, this study aims to develop a comprehensive and diverse dataset tailored specifically for urban sound classification and analysis. By leveraging advanced audio processing techniques and state-of-the-art machine learning models, our goal is to enhance the accuracy and reliability of urban sound recognition. The dataset will include a wide range of real-world urban sound samples, covering various environments, timeframes, and contextual variations. This will enable the development of robust algorithms capable of distinguishing between different urban sound sources with greater precision.

Furthermore, this study will explore innovative approaches such as deep learning architectures, spectrogram-based analysis, and hybrid models that combine traditional signal processing with modern artificial intelligence techniques. By refining classification methods and improving dataset diversity, we aim to contribute valuable insights to urban noise management strategies, ultimately fostering healthier and more sustainable cities.

Through this research, we hope to bridge the gap between theoretical advancements in sound classification and practical implementations in urban environments, paving the way for more efficient and intelligent urban sound management solutions.

II. RELATED WORK

Urban sound classification and analysis have emerged as critical components of smart city development, aiding in noise pollution monitoring, public safety, and urban planning. The field has evolved from rudimentary noise level assessments to advanced machine learning-driven models capable of classifying diverse sound sources in complex urban environments. Early studies primarily relied on handcrafted feature extraction and traditional classification methods, such as support vector machines (SVM) and hidden Markov models (HMM). However, these approaches often struggled with the overlapping and dynamic nature of urban soundscapes.

The advent of deep learning, particularly convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid architectures like CNN-LSTM, has revolutionized urban sound classification. CNNs excel at capturing spatial patterns in spectrogram representations of audio signals, while RNNs and LSTMs are effective at modeling temporal dependencies, making them well-suited for analyzing urban sound sequences. Research has demonstrated the effectiveness of these models in distinguishing various urban sounds, including vehicular traffic, human speech, construction activities, and natural ambient noises. Additionally, transformer-based models have recently gained attention for their ability to process long-range dependencies in audio sequences, further enhancing classification accuracy.

Despite these advancements, several challenges persist in real-world applications. Data availability and quality remain major concerns, as large-scale, well-annotated urban sound datasets are essential for training robust models. Noise variations due to environmental factors, such as weather conditions, time of day, and urban architecture, introduce generalization challenges, making it difficult to apply models trained on one city to another. Computational efficiency is another limitation, as deploying deep learning models in real-time applications requires high processing power and optimized algorithms. Furthermore, real-world urban environments contain overlapping and unpredictable sound events, making classification tasks more complex compared to controlled laboratory conditions.

To address these issues, future research should focus on developing diverse and representative urban sound datasets, incorporating data augmentation techniques to improve generalization, and exploring lightweight deep learning architectures for real-time deployment. Additionally, integrating urban sound classification with edge computing and Internet of Things (IoT) devices could enable decentralized, real-time noise monitoring systems for smart cities. By overcoming these limitations, urban sound classification can significantly contribute to enhanced noise management, improved urban living conditions, and more effective city planning strategies, ultimately leading to sustainable and intelligent urban environments.

III. PROPOSED WORK

Step 1: Dataset

ISSN NO: 9726-001X

Volume 13 Issue 02 2025

The research begins with the collection of a comprehensive urban sound dataset, organized into distinct categories representing various urban sound types, such as traffic, sirens, and human chatter. Each category consists of multiple audio files in WAV format, providing a diverse range of sounds to analyze. The dataset is stored in a directory structure that allows easy access and management of audio files.

Step 2: Dataset Preprocessing

The preprocessing phase involves several key steps to prepare the audio data for analysis. First, any null values in the dataset are checked and removed to ensure the integrity of the data. Then, the audio files are processed to remove background noise, enhancing the quality of the recordings. This step is crucial for obtaining clearer features from the audio, which will aid in the classification task. Additionally, features are extracted using Mel-frequency cepstral coefficients (MFCCs), which serve as a representation of the audio signal's characteristics.

Step 3: Label Encoding

To facilitate machine learning, the categorical labels associated with each sound file are transformed into numerical values through label encoding. This process involves mapping each category to a unique integer, allowing the algorithms to interpret the labels numerically. This step is essential for effectively training classification models, as machine learning algorithms require numerical input.

Step 4: Data Splitting

The dataset is then split into training and testing sets using a stratified approach to maintain the distribution of categories across both sets. This ensures that the model is trained and validated on representative samples, thereby enhancing its generalizability. The training set is used for model training, while the testing set is reserved for performance evaluation.

Step 5: Existing Algorithm

The existing algorithm utilized in this project is the Multi-Layer Perceptron (MLP) Classifier. MLP is a type of neural network that consists of multiple layers of neurons, including input, hidden, and output layers. It works by passing input data through these layers, where each neuron applies a weighted sum and an activation function to determine its output. While MLPs can model complex relationships in data, they may suffer from issues like overfitting and require careful tuning of hyperparameters.

Step 6: Proposed Algorithm

In contrast, the proposed algorithm is the LightGBM (LGBM) Classifier. LGBM is a gradient boosting framework that uses tree-based learning algorithms to build models. It operates by constructing multiple decision trees sequentially, where each tree corrects the errors of its predecessor. This approach significantly improves training speed and reduces memory consumption compared to traditional boosting methods. LGBM's architecture leverages a histogram-based algorithm, which efficiently handles large datasets and high-dimensional data.

Step 7: Performance Comparison

The performance of both algorithms is evaluated using several metrics, including accuracy, precision, recall, and F1-score. A confusion matrix is generated to visualize the classification results, providing insights into where the model performs well and where it may struggle. The results are compared to determine which algorithm better classifies urban sounds and to assess the improvements made by implementing LGBM over MLP.

Step 8: Prediction of Output from Test Data

Finally, the trained model is used to make predictions on new test data. The audio files are preprocessed in the same manner as the training data, ensuring consistency in feature extraction. The model's prediction capability is demonstrated using an example audio file, where the predicted category is printed out. This step showcases the practical application of the trained model in real-world scenarios, highlighting its utility for urban sound classification.

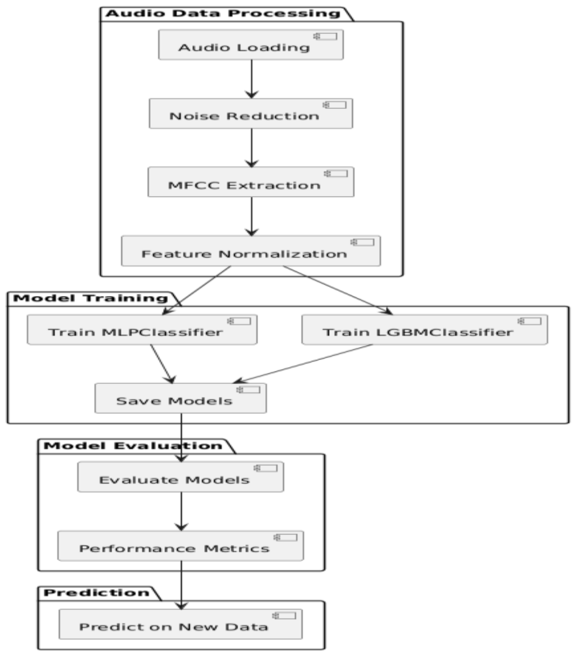

Figure 1: Architectural Block Diagram

ISSN NO: 9726-001X

Volume 13 Issue 02 2025

4.2 Workflow Data Preprocessing: Data preprocessing is a critical phase in any machine learning project, especially when working with real world datasets. It ensures that the data is clean, consistent, and ready for use in training machine learning models. In the context of the agricultural yield prediction project, data preprocessing involves several steps: Null Value Handling: The dataset is first examined for missing values (null values). Null values can occur for various reasons, such as data collection errors or incomplete records. These missing values are problematic as they can distort model predictions. The preprocessing step handles null values by either removing rows with missing data or imputing the missing values with appropriate replacements, such as the mean or median for numerical features or the most frequent value for categorical features.

Descriptive Statistics: Descriptive statistics are generated to understand the distribution of the dataset. This includes calculating the mean, median, standard deviation, minimum, and maximum values for each numerical feature. Descriptive statistics help identify outliers, anomalies, or any potential issues that need to be addressed before training the model. Identifying Unique Values: Categorical variables are examined to understand how many unique values they contain. This is important for encoding categorical data (if required), as it helps in determining the number of categories or labels that need to be transformed into a machine-readable format. Feature Scaling: Feature scaling is applied to numerical values to ensure that they are within a comparable range. For example, using techniques like MinMax Scaling, all values are transformed to a range between 0 and 1. This step is especially important for algorithms that rely on the distance between data points (such as neural networks) and helps the model converge faster and more reliably during training. 8 Data Splitting: Data splitting is another essential aspect of preparing data for machine learning, as it helps evaluate the model's performance on unseen data. In this project, the dataset is divided into two main subsets: the training set and the test set. Training Set: The training set is used to train the model. It contains a large portion of the dataset and allows the model to learn the relationships between the features and the target variable (in this case, agricultural yield). The model uses this data to adjust its parameters and fit the underlying patterns in the data. Test Set: The test set is used to evaluate the model's performance after training. It contains data that the model has not seen before, simulating real-world conditions where the model will encounter new, unseen data. By comparing the predicted values with the actual values in the test set, we can measure the accuracy and generalization ability of the model. Train-Test Split Ratio: In this project, the dataset is split into 80% training data and 20% test data using the train_test_split function. This ensures that the model has enough data to learn from, while also being evaluated on a sufficiently large set of test data to ensure it generalizes well. Resampling (Optional):In some cases, the dataset might be imbalanced, meaning that certain categories (e.g., low yield vs. high yield) are underrepresented. To address this, resampling techniques, oversampling the such as minority class or undersampling the majority class, can be used to ensure that the model is trained on a balanced dataset. This step is optional and depends on the nature of the data. Python Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a comprehensive standard library.

IV.RESULTS AND DISCUSSION

Implementation Description

- **Importing Libraries**
- The implementation begins by importing essential libraries for data handling, visualization, model training, evaluation, and serialization. Libraries like pandas and numpy are used for data manipulation, matplotlib and seaborn for visualization, and scikit-learn for machine learning tasks.
- **Dataset Loading and Exploration**
- The dataset is loaded from a CSV file named `urban_sound_data.csv` into a pandas DataFrame.
- Initial exploration of the dataset is done by checking its shape, structure, and the presence of any missing values. Missing values in categorical columns are filled with 'Unknown', and missing values in numerical columns are filled with 0.
- **Data Visualization**
- A count plot of the target variable `sound category` is generated to visualize the distribution of different

ISSN NO: 9726-001X

Volume 13 Issue 02 2025

sound classes. This helps in understanding the class balance in the dataset.

- **Label Encoding**
- Categorical variables in the dataset are encoded into numerical values using Label Encoder. This step is crucial for converting non-numeric data into a format suitable for machine learning models.
- **Data Resampling**
- The dataset is resampled to handle class imbalance and to ensure that the models have enough data to learn from. Resampling is done by generating a new dataset with 10,000 samples.
- **Train-Test Split**
- The dataset is split into training and testing sets using an 80-20 split. The training set is used to train the machine learning models, while the test set is used to evaluate their performance.
- **Model Building and Evaluation**
- **K-Nearest Neighbours (KNN):**
- If a pre-trained KNN model exists, it is loaded; otherwise, a new KNN model is trained with specific hyperparameters (p=5, algorithm='ball_tree', n_neighbors=4, weights='distance', leaf_size=17).
- The trained model is saved using joblib for future use.
- Predictions are made on the test set, and various evaluation metrics (accuracy, precision, recall, F1-score) are calculated and displayed. A confusion matrix is also generated to visualize the model's performance.
- **Multi-Layer Perceptron (MLP):**
- Similar to the KNN model, the MLP model is either loaded if it exists or trained from scratch. The MLP model has one hidden layer with 100 neurons and is trained for 300 iterations.
- The model is saved, and predictions are made on the test set. Evaluation metrics and a confusion matrix are generated to assess the MLP model's performance.
- **Comparison of Models**
- The performance metrics of both models (KNN and MLP) are compared in a tabular format. This comparison helps determine which model performs better in classifying urban sound categories.
- **Prediction on New Data**
- A new dataset (`new_audio_data.csv`) is loaded for testing the trained models. The data undergoes the same preprocessing steps, including filling missing values and label encoding.
- Predictions are made for each row in the new test data, and the results are printed, indicating the predicted sound category for each sample.

Dataset Description

1. Demographic Information:

- `fileID`: Unique identifier for each audio file.
- `duration`: Duration of the audio clip in seconds.

2. Audio Features:

- `chroma_stft`: Chroma short-time Fourier transform.
- `rmse`: Root mean square energy.
- `spectral_centroid`: Spectral centroid.
- `spectral_bandwidth`: Spectral bandwidth.
- `rolloff`: Spectral rolloff point.
- `zero_crossing_rate`: Rate of zero crossings in the audio signal.
- `mfcc1` to `mfcc20`: Mel-frequency cepstral coefficients (MFCCs), 20 in total.

3. Temporal Features:

- `tempo`: Tempo of the audio clip.
- `beat_frame`: Beat frame of the audio clip.

4. Metadata:

- `city`: City where the audio was recorded.
- `date_time`: Date and time of the recording.
- `weather`: Weather conditions during the recording.

5. Sound Category:

- `sound_category`: The category of the sound (e.g., car_horn, dog_bark, siren, etc.).

8.3 Results Description

To visualize the distribution of different sound categories in the dataset, a bar plot was created using the counts of each category. The count of sounds per category was calculated by iterating through the directory structure where the audio files are stored, and the results were compiled into a pandas DataFrame. The bar plot, with categories on the x-axis and their respective counts on the y-axis, displayed in a sky-blue color scheme, provides a clear overview of the number of audio samples available for each sound category. The plot includes labels for both axes, a title, and rotated category labels for better readability, ensuring a well-organized and informative visualization.

ISSN NO: 9726-001X

Volume 13 Issue 02 2025

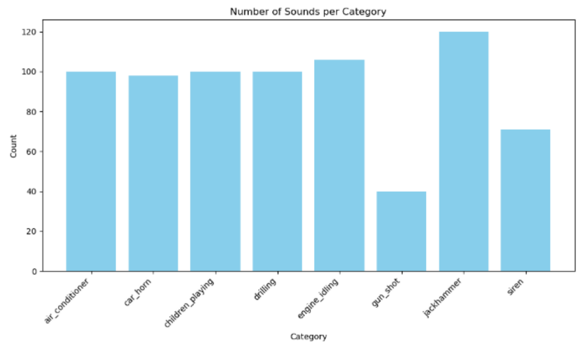

Fig 2: Count Plot for sound categories

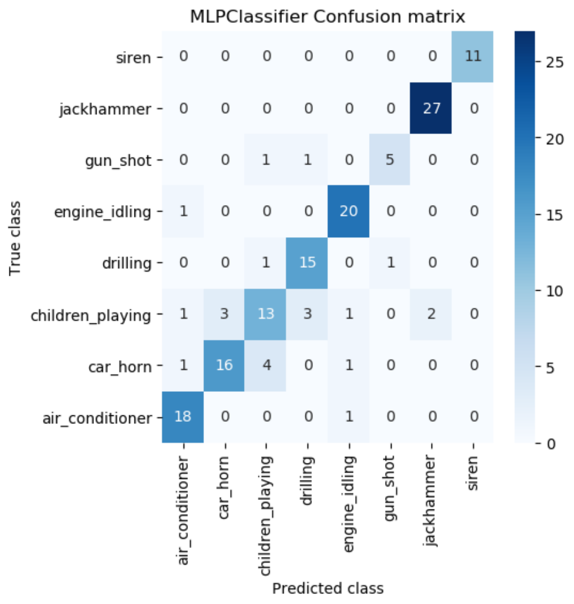

Fig 3: Multi-Layer Perceptron (MLP) model

The code begins by checking if a pre-trained Multi-Layer Perceptron (MLP) model exists at the specified path 'model/MLPClassifier'. If the model exists, it is loaded using joblib; otherwise, a new MLPClassifier instance is created. The model then makes predictions on the test set (`X_test`), and these predictions (`y_pred`) are evaluated against the true test labels (`y_test`) using the `performance_metrics` function to assess the MLPClassifier's performance.

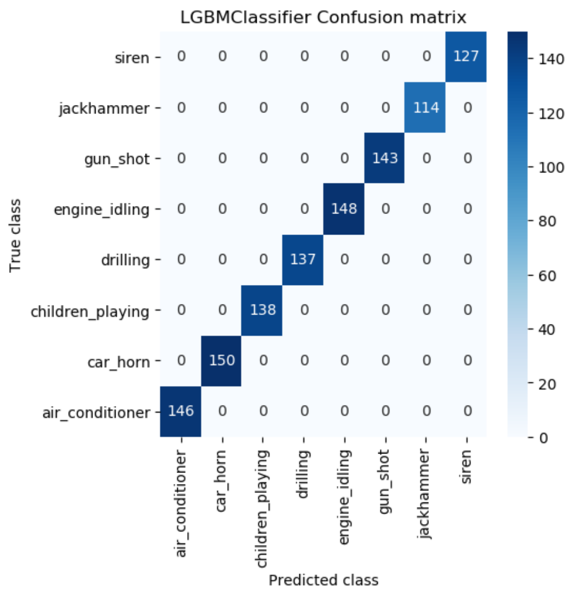

Fig 4: LGBMClassifier model

V CONCLUSION

The code first checks if a LightGBM classifier model (`LGBMClassifier`) already exists at the specified path (`'model/LGBMClassifier'`). If the model file is found, it is loaded using `joblib.load()`. If not, a new `LGBMClassifier` is instantiated, trained on the training data (`X_train`, `y_train`), and then saved to the specified path using `joblib.dump()`. After the model is either loaded or trained, it is used to make predictions on the test data (`X_test`). The predicted values (`y_pred`) are then evaluated against the actual test labels (`y_test`) using a function called `performance_metrics`, which computes and displays various performance metrics for the `LGBMClassifier`.

VI REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, et al. “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016: 779-788.

[2] J. Redmon, A. Farhadi. “YOLO9000: better, faster, stronger,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR).,2017: 7263 -7271.

[3] J. Redmon, A. Farhadi. “Yolov3: An incremental improvement,” 2018, arXiv:1804.02767

[4] Bochkovskiy, C.Y. Wang, H. Y. M. Liao. “Yolov4: Optimal speed and accuracy of object detection,”2020, arXiv:2004. 10934.

ISSN NO: 9726-001X

Volume 13 Issue 02 2025

[5] Cao, Z.; Yang, H.; Zhao, J.; Pan, X.; Zhang, L.; Liu, Z. A new region proposal network for far-infrared pedestrian detection. IEEE Access 2019, 7, 135023–135030.

[6] Park, J.; Chen, J.; Cho, Y.K.; Kang, D.Y.; Son, B.J.
CNN-based person detection using infrared images for night-time intrusion warning systems. Sensors 2020, 20, 34.

[7] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

[8] He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. 2015, 37, 1904–1916.

[9] Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. arXiv 2018, arXiv:1804.02767(1804).

[10] Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.

[11] Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.

[12] Li, X.; Zhao, L.; Wei, L.; Yang, M.H.; Wu, F.; Zhuang, Y.;
Ling, H.; Wang, J. Deepsaliency: Multi-task deep neural network model for salient object detection. IEEE Trans. Image Process. 2016, 25, 3919–3930.

[13]K. Lei, Z. Chen, S. Jia, and X. Zhang, "HVDetFusion: A Simple and Robust Camera-Radar Fusion Framework," in 2307.11323, 2023

[14]K. Shi, S. He, Z. Shi, A. Chen, Z. Xiong, J. Chen, and J. Luo, "Radar and Camera Fusion for Object Detection and Tracking: A Comprehensive Survey,"2410.19872, 2024

[15]S. Pang, D. Morris, and H. Radha, "TransCAR: Transformer-based Camera-And-Radar Fusion for 3D Object Detection," 2305.00397, 2023

[16] S. Yao, R. Guan, X. Huang, Z. Li, X. Sha, Y. Yue, E. G. Lim, H. Seo, K. L. Man, X. Zhu, and Y. Yue, "Radar-Camera Fusion for Object Detection and Semantic Segmentation in Autonomous Driving: A Comprehensive Review," in 2023

.[17] A. Nabati and H. Qi, "RRPN: Radar Region Proposal Network for Object Detection in Autonomous Vehicles," in 2020 IEEE International Conference on Image Processing (ICIP), 2020, pp. 3094-3098.