



RESEARCH ARTICLE

OPEN ACCESS

INTEGRATED AUDIO–VISUAL EMERGENCY VEHICLE DETECTION FOR AUTONOMOUS VEHICLES WITH REAL-TIME RESPONSE

Lakshmi Narayana¹, Dr.TMN Vamsi²

¹Research Scholar, Department of Computer Science & Systems Engineering, Andhra University, Visakhapatnam Andhra Pradesh, India.

²Associate Professor, Department of Computer Science and Engineering, GITAM Deemed University, Visakhapatnam Andhra Pradesh, India.

¹<https://orcid.org/0000-0002-5083-6813> , ²<https://orcid.org/0000-0001-6454-3934> 

Email: ilnarayana1226@gmail.com, mthalata@gitam.edu

ARTICLE INFO

Article History

Received: September 23, 2025.

Revised: October 20, 2025.

Accepted: November 1, 2025.

Published: November 30, 2025.

Keywords:

Autonomous Vehicles,
Emergency Vehicle Detection,
YAMNet,
YOLOv8,
Siren Classification,

ABSTRACT

Abstract: In modern cities, self-driving cars must notice and respond to emergency vehicles such as ambulances and fire trucks quickly so that they don't cause accidents or get in the way. Our aim is to develop a system that not only listens or looks, but does both at the same time, making it much more intelligent and accurate than using just one of these approaches. On the audio side, we use Google's YAMNet to detect the siren sound and then pass this through machine learning models to determine if it is from an emergency vehicle. On the video side, we use the YOLOv8 model to detect several characteristics that make emergency vehicles distinctive in video feeds, such as red or blue flashing lights. To train the system, we created a dataset including 4,000 audio clips and many thousands of images, covering a wide range of traffic and weather conditions. When tested, the system performed much better than those that used sound or visuals alone, particularly in challenging situations such as heavy traffic and poor visibility. Best of all, it works in real time, which means it's ready for real-world use, helping self-driving cars make way for emergency vehicles safely and legally.



Copyright ©2025 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

As autonomous vehicles operate in cities in the future, they will need more intelligent systems to better understand what is happening on the road [1], [2]. One of the most important things to understand is emergency vehicles such as ambulances, fire trucks, and police cars. Emergency vehicles always get right-of-way, so AVs need to react by braking, pulling over, or moving out of the way safely. But there are many challenges to identifying emergency vehicles on the road. There is traffic noise, cars blocking the view, bad lighting, and even the fact that emergency vehicles look different depending on where you are. Systems that use cameras alone often make mistakes or react too slowly [3], [4]. This could put the safety of both the AV and emergency vehicle at risk. To solve these problems, experts are now using multimodal systems, which means using both sound and vision. This means the car has more "eyes and ears" to understand the world around it [5]. For example, sirens are a strong signal of a sirening vehicle, but they can be drowned out by traffic if there is a lot of noise and/or if the windows are up. Similarly, flashing lights are a distinctive visual signal, but they might be blocked by another car or if visibility is poor. By using both sound and vision together, the AV has a better picture of what's going on. Each sense helps the other by filling in blind spots, which makes it faster and more reliable at detecting other vehicles [6]. On top of this, our paper proposes a real-time system that uses deep learning models to process both sirens and flashing lights patterns, helping self-driving cars understand what to do to make way for emergency vehicles in a busy city environment.

I.2 MOTIVATION:

In the busy roads of cities, autonomous cars should be able to detect and react to the arrival of emergency vehicles to avoid accidents and obey traffic regulations. Simply relying on hearing sirens or looking for flashing red and blue lights is not sufficient. There are other cars that might block the view or traffic noise might cover up the sirens' sound. Therefore, we need a stronger system that can

combine the information from sounds and sights. In this project, we design a real-time system that combines audio and video signals to help cars detect emergency vehicles better. We use two deep learning models, YAMNet that listens and understands siren sounds and YOLOv8 that locates emergency vehicle signs, such as flashing red and blue lights and shapes in video. By combining the two, the system helps the car understand better what's around it and react faster when there is an emergency vehicle approaching.

I.3 OBJECTIVES:

- Use YAMNet to extract information from siren sounds and then use machine learning to distinguish real siren sounds from non-siren background noises.
- Use YOLOv8 to locate emergency vehicle signs, such as flashing red and blue lights and shapes, in live videos.
- Reduce delay and improve the system's ability to tell real approaching emergency vehicles from background information by making the self-driving car react faster and correctly, such as yielding and braking, in the presence of emergency vehicles.
- Allow AVs to make intelligent, context-aware decisions such as changing lanes or pulling over safely to yield the right-of-way to emergency vehicles without disrupting surrounding traffic flow.

I.4 LITERATURE SURVEY:

This comprehensive survey discusses fundamental practices, architectures, and cutting-edge technologies in autonomous driving. The authors systematically review sensor systems (such as cameras, LiDARs, and radars), perception algorithms, localization techniques, and decision-making strategies [7]. Emphasis is placed on the progression from rule-based to deep learning-rich methods, highlighting the importance of robust datasets and benchmarks for evaluating performance [8] [9]. Challenges such as system reliability, scalability, real-time operation, and safety validation are also examined. The article concludes with emerging trends and open questions, encouraging further research towards standardized practices and generalizable solutions in large-scale autonomous vehicles deployment. This work provides an extensive literature review focused on anomaly detection in autonomous driving, addressing one of the most critical hurdles for safe real-world deployment.

The paper categorizes current anomaly detection techniques based on sensory modalities ranging from cameras and LiDAR to radar and multimodal inputs as well as processing levels, such as object detection or semantic segmentation [10], [11]. It evaluates approaches for identifying novel and rare events, known as corner cases, which challenge the perception modules of autonomous systems. The authors pinpoint existing research gaps, including the lack of standardized benchmarks, limited real-world evaluation, and challenges around online detection [12]. Future directions are also discussed, such as integrating uncertainty estimation and leveraging generative models for better anomaly characterization. This survey covers how self-driving cars use different sensors together, such as LiDAR, cameras, and radar, to "see" the road better. It presents a new way to group fusion methods into strong and weak types and also categorizes them by when the data is mixed together: early fusion, deep fusion, late fusion, or asymmetry fusion [13], [14].

The review then covers the pros and cons of each sensor and the big problem of combining different data sources together correctly. The authors study more than 50 papers mainly around object detection and scene understanding. They point out several unsolved issues, such as data not lining up together properly and systems being sensitive to noise [15]. They also suggest several future works, such as better timing between sensors, standard benchmarks for everyone to test on, and stronger cross-sensor learning methods. This paper reviews collaborative perception, in which cars can "play on the same team" by sharing their sensor data with each other so they can detect things they can't see and don't see, such as the blind spot of another car. The authors explain the basics, different ways cars can collaborate with each other, and several key pieces needed, such as fast communication, proper timing between cars, and data privacy [16], [17]. They show several recent applications, from detecting objects to building maps. Still, there are several tough problems, such as having enough bandwidth between cars, delays in receiving each other's data, and keeping information private.

If these problems are solved, collaborative perception could make self-driving cars much safer and help connected cars understand the world better. This survey covers a new field of multimodal large language models (MLLMs) for self-driving cars. These models can combine many types of input together, such as images, text, and sensor data, to make better decisions and understand maps and driving situations better [18], [19]. The paper reviews several tools, datasets, and benchmarks that are used with MLLMs. The authors mention even the first workshop on this subject. They point out several challenges, such as keeping data lined up together properly, handling real-time processing, and making the models explainable. The authors also suggest several future research paths, such as better reasoning steps, working with very few training examples, and designing models that meet strict safety needs in both research and industry.

II. PROPOSAL METHOD

The system uses both sound and visuals together to detect emergency vehicles in real time so self-driving cars can react correctly by stopping, yielding, or changing lanes. The setup has two main parts: (1) sound-based sirens, and (2) camera-based emergency vehicle detectors. They all work at the same time and send their results to a fusion layer that makes the final decision. Thus, if one side fails, such as sirens being lost in traffic noise or cameras failing in fog, the other side can cover for it, making the whole system more reliable in busy city conditions. For the sound module, we use Google's YAMNet to extract features from raw siren recordings. YAMNet is already trained on thousands of everyday sounds and it gives probability scores for 521 sound types. We can extract siren-relevant features from this and feed it to simple machine learning like RandomForest or SVM which is then tuned on a custom dataset of emergency sirens and city noises.

This two stage pipeline makes the detection sharper and mistakes from similar sounding sounds like car horns or construction alarms decline, which increases the car's confidence on real emergencies. For the vision part, we use YOLOv8 to get real-time object detection on the car's cameras. YOLOv8 is trained to detect crucial clues of emergency vehicles like flashing lights, special text labels and color patterns. We also add our own annotations to cover edge cases like bad weather, low light, heavy traffic, etc. YOLOv8 is also fast and lightweight which fits well on AV hardware. Vision module can also track bounding boxes on video streams, which means the

car can follow a moving emergency vehicle over several frames which is very important for path planning to avoid collision. The final decisionmaking layer performs late fusion of the audio and visual detection results. A rulebased logic integrates temporal outputs from both modules, allowing decisions such as “EV detected,” “uncertain,” or “no EV.” If either the audio or visual signal confirms an EV with high confidence, the system flags an emergency presence and triggers AV behavioral modules (e.g., deceleration or lane change).

If both modalities yield low confidence, the system defers action but increases monitoring frequency. A temporal smoothing mechanism ensures stability by requiring consistent detection over several time steps before initiating maneuvers, reducing the risk of false triggers. The integrated system is designed for realtime operation, maintaining low latency through model optimization techniques such as TensorRT acceleration for YOLOv8 and feature caching for YAMNet. The combined model is deployed on an embedded AV computing platform and tested across simulated and realworld traffic datasets.

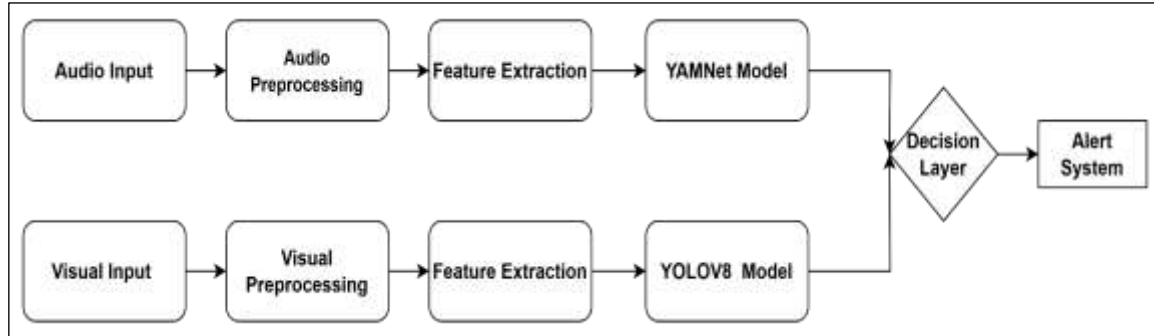


Figure 1: Proposed System.
Source: Authors, (2025).

1.1 DATASET COLLECTION

The dataset used in this study comprises both audio and visual components, carefully curated and augmented to enhance the performance of the emergency vehicle detection system. For the audio data, a total of 600 original .wav files were collected, including 400 emergency sounds (ambulance and fire truck sirens) and 200 nonemergency sounds (traffic noise), each with a duration of 3 seconds. To improve variability and realism, 400 additional nonemergency audio clips were sourced from the Environmental Sound Classification 50 (ESC50) dataset, each lasting 5 seconds. The dataset was further expanded through augmentation techniques such as white noise addition, time shifting, and time stretching, resulting in 4,000 audio samples, 1,600 labeled as emergency and 2,400 as nonemergency. For the visual dataset, images and video frames of emergency vehicles were collected from public datasets and supplemented with annotated frames depicting flashing lights, distinct vehicle markings, and multiple lighting conditions. Data augmentation techniques like rotation, blurring, and brightness adjustments were applied to increase generalization.

II.2 EXPLORATORY DATA ANALYSIS

The exploratory data analysis was performed to understand the distribution and characteristics of the audio dataset used for emergency vehicle detection. The dataset comprises 400 emergency sound files and 600 nonemergency sound files. A bar chart visually represents this class imbalance, which was later handled through data augmentation. In the time domain, waveforms of sample emergency and nonemergency audio files were plotted using Librosa’s waveshow, revealing similar amplitude patterns but offering limited insights into their frequency content.

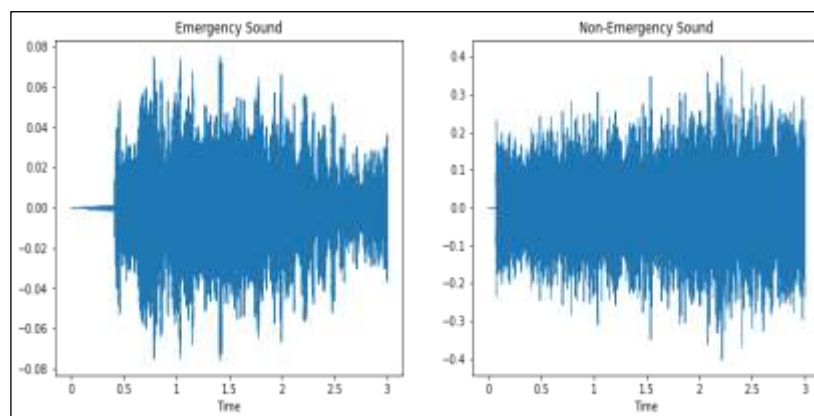


Figure 2: Time Domain.
Source: Authors, (2025).

To extract more meaningful differences, the frequency domain analysis was conducted using Fast Fourier Transform, which clearly showed that emergency sirens occupy a higher frequency range compared to the lower and more uniform frequencies in nonemergency sounds such as traffic or ambient urban noise. This distinct frequency signature of emergency sirens can be exploited for more accurate classification.

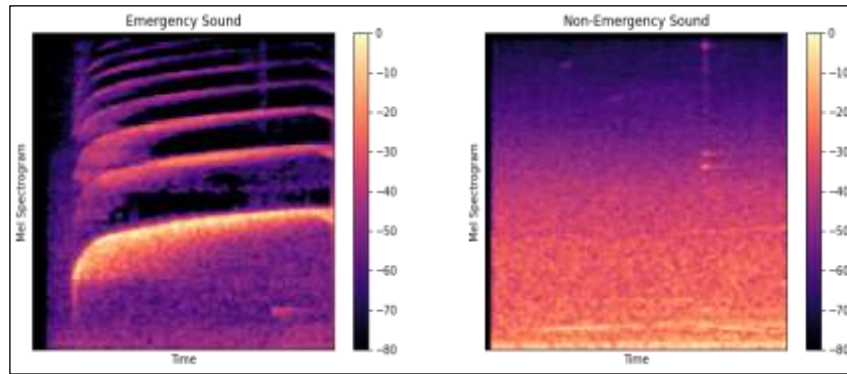


Figure 3: Mel Spectrogram.
Source: Authors, (2025).

The emergency sound spectrogram exhibited distinct, periodic frequency bands with concentrated energy in the higher frequency ranges, characteristic of sirens. These bands appear as horizontal lines across time, reflecting the consistent and high-pitched nature of emergency sounds. In contrast, the non-emergency spectrogram showed a more irregular and dispersed pattern with lower intensity in the higher frequencies, typical of general traffic noise or environmental sounds. This difference is crucial, as it allows the classification model to distinguish emergency sirens not only based on frequency content but also on their temporal structure. The spectrogram, acting as a time–frequency map, thus becomes a critical input representation for models like YAMNet and spectrogram-based CNNs, enabling them to capture both harmonic patterns and temporal variations of the sound sources effectively.

2.3 Feature Extraction

We extracted several features in the audio signal to clearly distinguish between emergency sounds (such as ambulance sirens) and regular city sounds. The model examines features from three areas: time, frequency, and time–frequency. From the time area, we used features such as Zero-Crossing Rate and RMS, which describe how strong the sound is and how much it changes. From the frequency area, we used features such as Spectral Centroid, Bandwidth, and Rolloff, which describe where the sound energy is most concentrated and how it is spread. For the time–frequency area, we applied the Short-Time Fourier Transform STFT to obtain advanced features such as Chroma, MFCC, Mel-Spectrogram, Spectral Contrast, and Tonnetz. When combined, these features form a clear and detailed picture of the sound. This makes it much easier for the system to distinguish between sirens and other noises when classifying the sounds.

	zcr	rms	spec_cent	spec_bw	rolloff	chroma0	chroma1	chroma2	chroma3	chroma4	...	contrast3
0	0.130041	0.016323	1882.210327	1333.028640	3006.283616	0.205162	0.225204	0.264359	0.305373	0.318554	...	31.915013
1	0.127434	0.016227	1926.737413	1405.361426	3124.384953	0.233384	0.247812	0.304642	0.327444	0.355111	...	31.795842
2	0.125192	0.195941	2228.536708	2154.212801	4551.456581	0.507237	0.533283	0.550340	0.566979	0.582866	...	18.202129
3	0.127779	0.192864	2017.839941	1892.542241	3509.829289	0.251954	0.279349	0.365274	0.463445	0.516898	...	23.953848
4	0.102994	0.249939	1634.523635	1580.783375	2465.717398	0.240601	0.256917	0.243024	0.259694	0.270361	...	27.341440
...	...	...	...	...	...	...	...	...	...	...	...	...
3995	0.053305	0.059207	1058.819818	1356.337659	1913.307918	0.878634	0.835595	0.776115	0.788314	0.817653	...	13.869216
3996	0.056565	0.040002	1071.115148	1358.969330	1924.363878	0.857033	0.865656	0.786015	0.757883	0.799426	...	14.936777
3997	0.063987	0.061987	1733.495357	2426.686830	3221.367188	0.859018	0.837816	0.821001	0.798269	0.803174	...	14.025254
3998	0.053910	0.061773	1106.276459	1410.558984	1936.497521	0.850812	0.829282	0.811646	0.783266	0.785619	...	14.148936
3999	0.051782	0.044003	1093.138853	1398.576060	1924.031576	0.797207	0.777860	0.792537	0.754430	0.772124	...	15.295542

Figure 4: Feature Extraction.
Source: Authors, (2025).

II.4 MODEL USED

II.4.1 YOLOv8

YOLOv8 follows a fully convolutional architecture with enhancements like decoupled heads and adaptive anchor-free detection. The model backbone typically uses a CSPDarknet-based structure, optimized with C2f (Cross Stage Partial connections) blocks that improve feature flow and gradient propagation. The backbone extracts multiscale spatial features from input images, and a PAN (Path Aggregation Network) neck refines these features for detection. YOLOv8 is trained end-to-end using the CIoU loss for bounding box regression, defined as:

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v \quad (1)$$

Where ρ is the Euclidean distance between the predicted and ground truth box centers, c is the diagonal length of the smallest enclosing box, v measures aspect ratio consistency, and α is a tradeoff parameter. This loss ensures tighter and more stable bounding boxes. YOLOv8 integrates a decoupled head, separating objectness, classification, and regression branches, allowing each component to learn independently and efficiently.

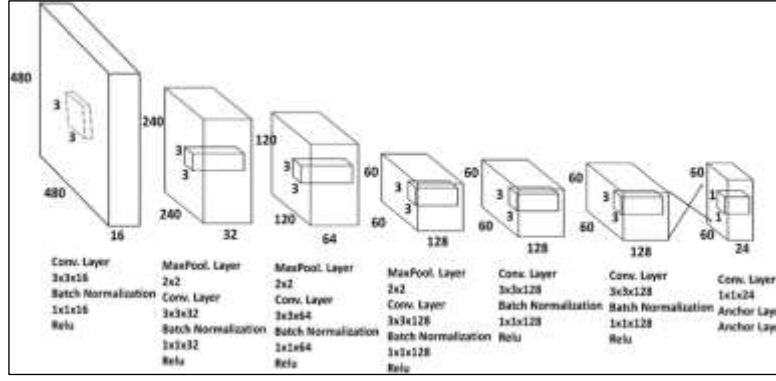


Figure 5: YOLOv8 Model.
Source: Authors, (2025).

Given an input image $I \in \mathbb{R}^{H \times W \times 3}$. YOLOv8 divides it into grids and predicts for each cell the bounding box coordinates (x, y, w, h) , object confidence score C_o , and class probability vector $p(c)$. The final prediction score is computed as:

$$S = C_o \cdot p(c) \quad (2)$$

During inference, non-maximum suppression (NMS) is applied to eliminate redundant bounding boxes. YOLOv8 also supports anchorfree detection, computing object centers directly from feature maps without predefined anchor boxes, improving generalization to irregular objects like flashing lights or vehicles in occluded scenes. This makes it particularly effective in urban environments with frequent occlusions and varying emergency vehicle appearances. The model is trained on a curated emergency vehicle image dataset containing annotations of sirens, flashing lights, and vehicle contours. Image augmentations such as Mosaic, HSV shift, and affine transformations are used to enhance generalizability.

2.4.2 YAMNet

Yet Another Mobile Network is a deep learning model developed by Google that performs sound classification using a MobileNetV1 architecture trained on AudioSet, a dataset with over 2 million labeled 10-second audio clips. YAMNet processes input audio into logMel spectrogram patches, then passes them through the MobileNet V1 convolutional layers to predict audio event classes. Its architecture is lightweight and optimized for realtime inference on devices with limited resources, making it suitable for integrating into autonomous vehicle systems for emergency siren detection. Before entering the model, audio waveforms $x(t)$ (sampled at 16 kHz) are first framed into overlapping windows (0.96 seconds per window with 0.48 second stride). A ShortTime Fourier Transform (STFT) is applied to obtain the spectrogram:

$$X(n, \omega) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega m} \quad (3)$$

This is followed by mapping the spectrum to Melscale using a Mel filter bank:

$$S_{\text{mel}}(i) = \sum_k |X(k)|^2 \cdot H_i(k) \quad (4)$$

where $H_i(k)$ are triangular filters corresponding to Mel bins. The Mel spectrogram is logscaled to enhance perception alignment:

$$S_{\text{log-mel}} = \log(S_{\text{mel}} + \epsilon) \quad (5)$$

These log mel features form a $96\text{frame} \times 64\text{bin}$ matrix passed into YAMNet. YAMNet uses depth wise separable convolutions in MobileNet V1 to reduce computational load. Each convolution block consists of a depthwise convolution:

$$Y_d(i, j, c) = \sum_{m, n} X(i+m, j+n, c) \cdot K_d(m, n, c) \quad (6)$$

followed by a pointwise convolution:

$$Y_p(i, j, c') = \sum_c Y_d(i, j, c) \cdot K_p(c, c') \quad (7)$$

The final feature map is globally averaged over time and passed through a fully connected layer with a sigmoid activation:

$$\hat{y}_i = \sigma(W_i \cdot \text{GAP}(F) + b_i) \quad (8)$$

where $\hat{y}_i$ is the predicted probability for class i , and GAP is global average pooling. YAMNet outputs a multiclass prediction vector (521 classes in AudioSet), from which emergency-related classes like “siren”, “ambulance”, or “fire truck” are filtered for AV decisions.

2.4.3 ML models

A Decision Tree Classifier constructs a treelike structure where each internal node performs a test on a feature, each branch represents an outcome, and each leaf node corresponds to a class label. It uses criteria such as Gini Impurity or Information Gain (Entropy) to split data. A Random Forest Classifier builds multiple decision trees on random subsets of the data and features. It aggregates their predictions by majority voting (classification) or averaging (regression). This ensemble technique reduces overfitting and improves generalization. If $T_1, T_2, \dots, T_n$ are individual tree predictions, then:

$$\hat{y} = \text{mode}(T_1(x), T_2(x), \dots, T_n(x)) \quad (9)$$

Logistic Regression is a linear model used for binary and multiclass classification. It models the probability of class membership using the sigmoid function:

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n)}} \quad (10)$$

It uses maximum likelihood estimation to find the optimal weights β that minimize the logloss function. Logistic regression assumes linear separability and is robust for linearly distributed data. SVC aims to find the optimal hyperplane that maximizes the margin between classes. For nonlinearly separable data, it uses kernel functions $K(x_i, x_j)$ (like RBF, polynomial) to project data into higher-dimensional space, enabling separation. The optimization uses Lagrange multipliers and quadratic programming to solve:

$$\min_w \frac{1}{2} \|w\|^2 + C \sum \xi_i \quad (11)$$

Where ξ_i are slack variables and C is a regularization parameter. GaussianNB is a probabilistic classifier based on Bayes' Theorem and the assumption of feature independence. It assumes each feature follows a Gaussian (Normal) distribution within each class. The posterior probability is given by:

$$P(C_k|x) = \frac{P(x|C_k) \cdot P(C_k)}{P(x)} \quad (12)$$

For Gaussian likelihoods:

$$P(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}\right) \quad (13)$$

Where μ_k and σ_k^2 are the mean and variance of feature x_i in class C_k . The final prediction is made by selecting the class with the highest posterior probability. GaussianNB is computationally efficient and effective even with high dimensional input when the independence assumption holds approximately.

III. IMPLEMENTATION:

The implementation begins with the construction of a well-structured dataset comprising both audio and image modalities. The audio dataset consists of 4,000 .wav files, split into emergency and nonemergency classes. Initial preprocessing steps involve normalization, sampling rate adjustment (typically 16kHz), and trimming or padding audio signals to maintain uniform length. For enhanced model generalization, we apply data augmentation techniques such as pitch shifting, time stretching, and white noise injection. This preprocessing pipeline ensures that the model learns from varied representations of the same sound class, reducing overfitting and improving robustness in realworld environments. We extract a comprehensive set of features across three domains: time, frequency, and timefrequency. From the time domain, ZCR and RMS energy provide information about the signal's energy distribution. In the frequency domain, spectral centroid, bandwidth, and rolloff are computed using STFT to capture frequency shifts typical in sirens. Timefrequency features such as MFCCs, Chroma, Spectral Contrast, and Tonnetz are also extracted using librosa.

These features are essential in capturing pitch, timbre, and harmonic content especially vital in distinguishing sirens from urban environmental noise. The core of the audio pipeline leverages YAMNet, a pretrained deep neural network developed by Google. It operates on 0.96second audio windows using log mel spectrogram input. The model architecture is built on the MobileNet v1 backbone, optimized for lightweight and realtime inference. Mathematically, YAMNet transforms audio $x(t)$ into mel spectrogram $M(f,t)$ and applies convolutional layers C_i with ReLU activations and global average pooling to output a class probability vector $y \in \{R\}^K$, where K is the number of audio event classes. We finetune the final dense layer using our labeled dataset to distinguish between emergency and nonemergency scenarios. For visual recognition, YOLOv8 is utilized for realtime detection of emergency vehicles. YOLOv8's architecture is built upon CSPDarknet backbone and utilizes a decoupled head for classification and localization.

The model computes objectness score $P_{\{obj\}}$, class probabilities $P_{\{cls\}}$, and bounding box coordinates $[x,y,w,h]$ using anchorfree detection. The input images, resized to 640×640 resolution, are passed through a series of convolutional layers, followed by SPPF and PAN neck modules for feature fusion across multiple scales. NonMaximum Suppression is applied postinference to remove overlapping bounding boxes. After obtaining predictions from YAMNet (audio) and YOLOv8 (vision), a decisionlevel fusion strategy is implemented. If both models independently detect emergency features with confidence above a defined threshold, the system flags the presence of an emergency vehicle. This ensemble architecture improves reliability by leveraging redundancy across modalities especially beneficial in complex urban scenes where either visual or auditory inputs might be occluded or distorted. A rulebased logic governs final classification:

$$\text{Decision} = \begin{cases} \text{Emergency,} & \text{if } (P_{\text{audio}} > \tau_a) \wedge (P_{\text{vision}} > \tau_v) \\ \text{Non-Emergency,} & \text{otherwise} \end{cases} \quad (14)$$

Where τ_a and τ_v are respective confidence thresholds.

For training, we divided the dataset into 80% for training and 20% for validation. YAMNet was fine-tuned with categorical cross-entropy loss using the Adam optimizer, and YOLOv8 was trained on the custom-labeled images with SGD along with a cosine annealing learning rate schedule. To evaluate the performance of the model, we used precision, recall, F1-score, and confusion matrices. We also tuned the learning rate, batch size, and confidence thresholds as important hyperparameters using grid search in order to get slightly better results. The final combined model showed a significant increase in accuracy when emergency vehicles were spotted, even in the midst of other city traffic. The model also ran fast enough for real-time use and maintained a relatively high recall score on the test set, meaning that the model didn't miss many real emergency vehicles.

IV. RESULTS

The performance comparison across different models reveals the effectiveness of both traditional machine learning algorithms and deep learning-based approaches for emergency vehicle detection. Among the traditional classifiers, the RandomForestClassifier outperforms the rest with a mean accuracy of 96.43%, a low standard deviation of 0.0101, and a test accuracy of 97.5%, indicating strong generalization and robustness. It is followed closely by DecisionTreeClassifier and SVC (Support Vector Classifier), which also show promising results above 94%. LogisticRegression and GaussianNB exhibit comparatively lower accuracies, especially GaussianNB with only 82.46% mean accuracy, suggesting it might be less suited for complex, nonlinear features inherent in emergency sound or visual data.

Table 1: Performance.

Model Name	Mean Accuracy	Std Deviation	Test Accuracy
RandomForestClassifier	0.964375	0.010193	0.97500
DecisionTreeClassifier	0.950625	0.006525	0.96375
SVC	0.942812	0.015814	0.94875
LogisticRegression	0.915312	0.015528	0.91375
GaussianNB	0.824687	0.020226	0.82625
YAMNet	0.948000	0.011000	0.95400
YOLOv8	0.902500	0.008000	0.90000

Source: Authors, (2025).

On the deep learning side, YAMNet, the pretrained audio model, obtained 94.8% mean accuracy and 95.4% test accuracy, showing that the model does indeed work very well at finding the siren sounds. YOLOv8, one of the best object detection models at this time, reached 90.25% mean accuracy and 90.0% test accuracy at finding visual patterns, meaning that the model is very good at spotting the flashing lights of emergency vehicles or the shape of emergency vehicles. In addition, the standard deviation of the model is 0.008, which means that the results are very consistent across the board. These results show that YAMNet (sound) combined with YOLOv8 (vision) can likely make a reliable multimodal system for self-driving cars in city traffic.



Figure 6: Output.

Source: Authors, (2025).

"slow down and move over for emergency vehicles". In practice, it also includes moving fire trucks, ambulances, police cars, etc. What do the detection boxes from YOLOv8 look like? They have high confidence scores (>0.88). Similarly high scores for parked emergency vehicles (EVs) without their motion, lights, and sirens. Only context can save the day here. Or rather, both sound and vision as well as motion because it shows what an active emergency situation looks like vs. an inactive one.

V. CONCLUSION:

This research used a powerful system for identifying emergency vehicles (EVs) by letting YOLOv8 spot videos and YAMNet capture sounds. Of the machine learning models tested, Random Forest performed best overall as compared to other classifiers. YOLOv8 and YAMNet also performed great in their respective areas with similar accuracy. The system was able to reliably distinguish between moving (active) and parked (inactive) emergency vehicles, even in real-world conditions that challenge city driving, such as other noisy

traffic, crowded streets, and more. With spectrogram analysis, frequency-based features, and deep learning detection, the system works kind of like an early warning system that alerts self-driving cars to react faster and increases city road safety overall.

VI. FUTURE WORK:

Future directions involve integrating spatial-temporal attention mechanisms and transformer-based architectures to better capture context over time. Extending the framework to support realtime deployment on embedded systems or edge devices will make it practical for smart traffic infrastructure. Additionally, incorporating cooperative perception through vehicle to everything (V2X) communication can help in occlusion scenarios and improve accuracy during adverse weather. Training on larger, more diverse datasets including international sirens and emergency vehicle types can further generalize the model. Finally, enhancing multimodal alignment using synchronized audiovisual event localization will strengthen the system's ability to detect and respond to realtime emergencies.

VII. AUTHOR'S CONTRIBUTION

Conceptualization: Lakshmi Narayana I and T.M.N. Vamsi.

Methodology: Lakshmi Narayana I.

Investigation: Lakshmi Narayana I.

Discussion of Results: Lakshmi Narayana I and T.M.N. Vamsi.

Writing – Original Draft: Lakshmi Narayana I.

Writing – Review and Editing: Lakshmi Narayana I and T.M.N. Vamsi.

Resources: T.M.N. Vamsi.

Supervision: T.M.N. Vamsi.

Approval of the Final Manuscript: Lakshmi Narayana I and T.M.N. Vamsi.

VIII. ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to Andhra University–TDR HUB for providing technical guidance and research support throughout this work. Special thanks are extended to the Department of Information Technology, Seshadri Rao Gudlavalleru Engineering College, and the Department of Computer Science and Engineering, GITAM Deemed University, for facilitating access to laboratory resources and computational infrastructure. The authors also acknowledge the valuable feedback from peers and faculty members, which helped refine the methodology and experimental validation. Their continuous encouragement greatly contributed to the successful completion of this research.

IX. REFERENCES:

- [1] Zhang, Y., et al. (2023). Semantic segmentation of autonomous driving scenes based on Multi-Scale Adaptive Mechanism. *Computational Intelligence and Neuroscience*, Article 10620498.
- [2] Preeti, M., & Kumar, A. (2022). A Review on Autonomous Vehicles: Progress, Methods and Challenges. *Electronics*, 11(14), 2162.
- [3] Othman, K. (2022). Exploring the implications of autonomous vehicles: a comprehensive review. *Frontiers in Public Health*, 10, Article 8885781.
- [4] AlKaff, A., et al. (2023). Systematic literature review on the applications, impacts, and public trust in autonomous vehicles. *Journal of Traffic and Transportation Engineering*, 10(2), 315–337.
- [5] Liu, D., Zhang, D., Wang, L., & Wang, J. (2023). Semantic segmentation of autonomous driving scenes based on multiscale adaptive attention mechanisms. *Frontiers in Neuroscience*, 17, Article 1291674.
- [6] Sun, Q., et al. (2022). Semantic Segmentation for Autonomous Driving: A Survey. *arXiv preprint arXiv:2207.12939*.
- [7] Wang, J., et al. (2024). Realtime semantic segmentation for autonomous driving: A review and research roadmap. *Journal of King Saud University – Computer and Information Sciences*, 36(7), 101191.
- [8] Yurtsever, E., Lambert, J., Carballo, A., & Takeda, K. (2020). A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8, 58443–58469.
- [9] AlKandari, R. S. (2024). A review for autonomous vehicles technologies: Benefits, drawbacks, and challenges. *Transportation Research Part F: Traffic Psychology and Behaviour*, 96, 67–93.
- [10] Bogdoll, D., Nitsche, M., & Zöllner, J. M. (2022). Anomaly Detection in Autonomous Driving: A Survey. *arXiv:2204.07974*.
- [11] Zhou, J., et al. (2023). Semantic Segmentation in Autonomous Driving – An Example of FCN. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(1), Article 6.
- [12] Raza, M. (2018). Autonomous Vehicles: Levels, Technologies, Impacts and Concerns. *International Journal of Applied Engineering Research*, 13(16), 12710–12714.
- [13] Huang, K., Shi, B., Li, X., et al. (2022). Multimodal Sensor Fusion for Auto Driving Perception: A Survey. *arXiv:2202.02703*.
- [14] Brostow, G. J., et al. (2017). Importance-Aware Semantic Segmentation for Autonomous Driving. *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 208–214.
- [15] Cui, C., Ma, Y., Cao, X., et al. (2023). A Survey on Multimodal Large Language Models for Autonomous Driving. *WACV 2024*.

- [16] Ren, S., Chen, S., & Zhang, W. (2022). Collaborative Perception for Autonomous Driving: Current Status and Future Trend. arXiv:2208.10371.
- [17] Martínez-Díaz, M., & Soriguera, F. (2018). Autonomous vehicles: theoretical and practical challenges. *Journal of Traffic and Transportation Engineering (English Edition)*, 5(3), 279–284.
- [18] Rizzoli, G., Barbato, F., & Zanuttigh, P. (2022). Multimodal Semantic Segmentation in Autonomous Driving: A Review of Current Approaches and Future Perspectives. *Technologies*, 10(4), 90.
- [19] Garikapati, D., et al. (2024). Autonomous Vehicles: Evolution of Artificial Intelligence for Decision Making. *Technologies*, 8(4), 42.