Pakistan Journal of Engineering and Technology
ISSN (Print): 2664-2042; ISSN (Online): 2664-2050
Received: 2024-03-18
Accepted: 2024-04-24
https://doi.org/10.51846/vol7iss01pp22-33

PAKJET
Pakistan Journal of Engineering & Technology

Article

Automatic Identification of Emergency Sounds for Self-Driving Cars

Usman Afzal * and Lubna Farhi

Department of Electronic Engineering, Sir Syed University of Engineering and Technology, Karachi-75300, Pakistan
* Correspondence: Usman Afzal (MEE21S-003@ssuet.edu.pk, engrusmanafzal@hotmail.com)

Abstract: As the development of Self-Driving Cars (SDCs) advances, one important feature that requires attention is their ability to sense and respond to emergencies on the road. Drivers of emergency vehicles prioritize speed over safety during emergencies to save lives, potentially leading to accidents. However, manual cars sometimes may not understand the urgency of the situation, further increasing the risk of collisions. In such cases, self-driving cars offer a better solution to replace manual cars and minimize road accidents. These cars with emergency sound detections offer a level of responsiveness, accuracy, and consistency that surpasses manual cars, contributing to improved road safety and efficiency in emergencies. Therefore, the key objective of the research is to improve the listening capability of self-driving cars to detect emergency vehicles with the help of a hybrid feature extraction technique. The suggested technique leverages a combination of Complex Morlet Wavelet and Co-occurrence Matrix to obtain statistical features from the emergency sounds. The proposed technique can work with the input length of 1.2 seconds of raw waveforms. This research work investigates that self-driving cars can accurately identify emergency vehicles by examining the distinctive emergency sound patterns emitted by the emergency vehicle with the highest accuracy of 94%. At the same time, the proposed technique reduces the computational cost by 20 – 40 milliseconds when compared with other techniques. The result of this work not only provides better accuracy but also reduces detection time, which is a crucial requirement for real-time applications such as self-driving cars.

Keywords: Automatic Identification, Emergency Sounds, Machine Learning, Self-Driving Cars, Sound Recognition

1. Introduction

1.1. Background and Motivation

Self-driving cars (SDCs) are capable of driving without human input. These are often more efficient and safer than human drivers, leading to a significant reduction in accidents and associated costs. Many road accidents occur due to emergency automobiles because of their short duration of time of execution. The drivers of these vehicles often operate under intense time pressure to reach their destinations quickly, especially when responding to life-threatening situations. This pressure can cause drivers to prioritize speed over safety and take the risk of accidents.

Emergency Vehicles are recognized by their unique signals such as flashing lights and sirens. However, self-driving cars encounter difficulties in emergency vehicles recognition when operating at high speeds or adverse weather conditions. Moreover, failure of some sensors to detect emergency vehicles can potentially result in accidents with deadly consequences. This problem of self-driving cars to un-detect emergency vehicles represents a significant shortfall since SDCs were initially designed to be better at driving tasks. Therefore, it is important for SDCs to have the capability to recognize emergency vehicles in all potential road scenarios. This requires a robust and swift detection mechanism, such as detection through emergency sound, to ensure effective and accurate response of self-driving cars, regardless of the circumstances.

1.2. Literature Review

The automatic identification of emergency sounds is a critical aspect of enhancing the safety and efficiency of self-driving cars (SDCs). Many researchers have dedicated their efforts to studying this topic, contributing to our understanding of the challenges, methodologies utilized, and potential future advancements [1, 2]. The detailed literature survey of previous work along with methodology and shortcomings, is shown in Table 1.

Table 1. Literature review.

#	Author	Title	Methodology	Shortcoming
1.	Z. Islam and M. Abdel-Aty [3], 2022	"Real-time Emergency Vehicle Event Detection Using Audio Data"	The author partitioned audio data into fixed durations, extracting mel-frequency cepstral coefficients and zero crossing rate features, which were subsequently inputted into the Extreme Learning Machines (ELM) model for classification.	The research overlooks the impact of background noise on features extraction and performance of the classifier and its overall accuracy.

2.	H. Sun, X. Liu, K. Xu, J. Miao, and Q. Luo [4], 2021	"Emergency Vehicles Audio Detection and Localization in Autonomous Driving"	This paper utilizes a system that collects real-world siren data using cost-efficient microphones and extracted Mel-Frequency Cepstral Coefficients (MFCC) & Spectrogram features. Finally, provide these features to CNN for audio-based detection and localization of emergency vehicles.	In this research, although the accuracy is commendable, the model's reliance on lengthy input audio durations for effective emergency sound detection poses computational challenges, particularly for real-time applications such as self-driving cars.
3.	A. Garg, A. K. Gupta, D. Shrivastava, Y. Didwania, and P. J. Bora [5], 2019	"Emergency Vehicle Detection by Autonomous Vehicle"	The author applies camera and microphone for emergency sound detection. The images captured from camera are fed into a Deep Learning Convolutional Neural Network (CNN). And uses a pre-trained feature extractor to extract 128-dimensional audio features applies to support vector machine (SVM) model. The output from both the image detection and audio detection are combined to determine the presence of an ambulance.	This methodology utilizes high-dimensional features, demanding significant memory resources for training the classifier on a large dataset. However, this can potentially reduce the classifier's performance.
4.	M. Azad, F. Khaled, and M. R. H. Rumman [6], 2018	"An Efficient Way to Convert 1D Signal to 2D Digital Image Using Energy Values"	This research uses acoustic signals of faulty industrial motors which are considered as 1D time domain signals. Then transform 1D signals to 2D gray scale images utilizing their energy parameters Finally, extract 21 texture features from the gray scale images and uses SVM for classification.	The algorithm applied on a small dataset; it may limit the classifier's capacity to accurately classify new samples on large dataset.
5.	A. Sengür, S. Ekici, Y. Akbulut, and T. Kavas [7], 2017	"Time-Frequency Gray Level Co-occurrence Matrix Descriptors for Deception Detection"	The input speech signal is transformed into a spectrogram and then converted into 8-bit grayscale images. Some features of images are obtained from these grayscale images using GLCM. The effectiveness of this method is assessed using a real-life dataset, showing a classification accuracy higher than previously reported results.	This research overlooks the impact of background noise on the overall accuracy.

1.3. Contribution

Audio sensing technology has emerged new capabilities in self-driving cars. Therefore, the purpose of this research is to propose a "Hybrid Feature Extraction Technique" for self-driving cars to identify emergency sounds on different road situations by using machine learning classifiers. The proposed objectives aim to surpass the accuracy of existing methods, potentially offering better accuracy while being less consumption extensive. Additionally, it could effectively handle various types of road noises and accurately detect emergency sounds among them.

1.4. Paper Organization

The paper is structured as follows: Section 1 comprises an introduction and a review of related literature. Section 2 outlines the methodology employed in this research, with Section 3 presenting the obtained results. Section 4 concludes the research findings, while Section 5 proposes future actions in the field.

2. Proposed Methodology

This paper employs a hybrid feature extraction technique to classify emergency sounds, utilizing five machine learning classifiers - Decision Tree (DT), K-Nearest Neighbor (KNN), Naïve Bayes (NB), Support Vector Machines (SVM) and Artificial Neural Network (ANN). The proposed methodology workflow for emergency sound detection is illustrated in Fig. 1. A detailed explanation of how the method works is given below.

2.1. Audio Dataset

The research must prioritize the dataset, which consists of 1834 audio files in .wav format [8]. Among these, 932 files capture emergency vehicle signals, while 902 contain non-emergency ambient road noises. Each file varies in parameters such as min-max range, sample rate, duration and number of channels. Preprocessing is essential to address these variations for consistent analysis and results.

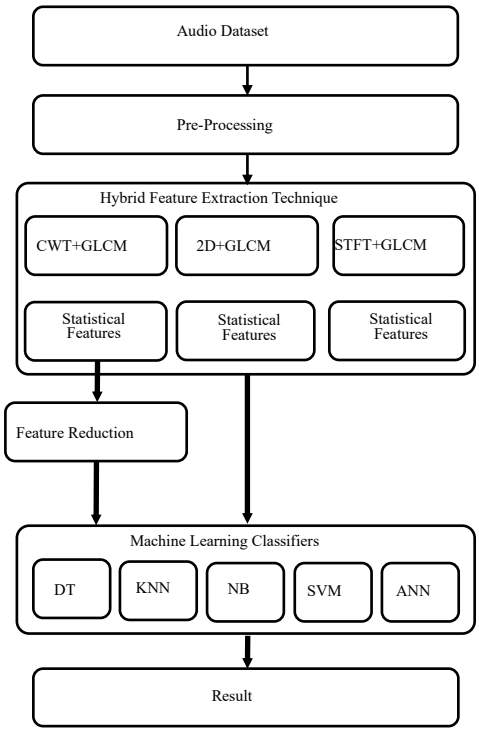

Figure 1. Proposed methodology flow diagram.

2.2. Pre-Processing

For comprehensive research endeavors, numerous parameters within the downloaded dataset demand attention. Key factors encompass rescaling, stereo to mono conversion, resampling, silence removal, trimming and finally, rescaling of trimmed audio of individual files.

Rescaling adjusts the range of a dataset [9]. Each audio file has a different min-max range based on the bit depth. For instance, 16-bit data is loaded within the range of ±33x10^3, 24-bit data within ±8.4x10^6, and 32-bit data within ±2.2x10^9. All files above 16-bit depth were rescaled to the range of ±33x10^3 using (1).

$$Y_{rescaled} = l + \left[\frac{Y - Y_{min}}{Y_{max} - Y_{min}}\right](u - l) \quad (1)$$

where, Y is original signal and $Y_{rescaled}$ is rescaled signal in $[l, u]$ range.

In the dataset, there are 1682 stereo recordings and 152 mono recordings. Equation (2) is utilized for conversion between stereo and mono sound.

$$Ch_m[n] = \frac{Ch_l[n] + Ch_r[n]}{2}, \quad n = 0, 1, 2, \dots, L-1 \quad (2)$$

where, Ch_l and Ch_r represent left and right channels of the stereo sound while Ch_m represents the single channel of mono sound. L is the length of signal.

Resampling adjusts the sample rate of an audio signal, varying from 8 to 96 KSPS within the dataset [10]. This process involves up-sampling followed by anti-aliasing low pass filter and down-sampling. Equations (3, 4) are used for up-sampling process.

$$x_u[n] = \begin{cases} x\left[\frac{n}{L}\right], & n = 0, \pm L, \pm 2L, \dots \\ 0, & otherwise \end{cases} \quad (3)$$

where, $x_u[n]$ denotes the up-sampled signal and L is the up-sampling rate.

$$y = \left(\frac{y_2 - y_1}{x_2 - x_1}\right).(x - x_1) + y_1 \quad (4)$$

where (x_1, y_1) is the previous known sample, (x_2, y_2) is the next known sample and (x, y) is unknown sample.

After up-sampling, an anti-aliasing low-pass filter is employed to mitigate aliasing caused by frequencies exceeding half the sample rate (the Nyquist frequency). This filter ensures accurate capture of frequencies within the desired bandwidth by setting the cutoff frequency slightly below the Nyquist frequency. Equations (5 - 7) govern the anti-aliasing low-pass filter.

$$h_{ideal}\left[n - \frac{\mathcal{M}}{2}\right] = \begin{cases} \frac{Sin(\omega_c(n - \frac{\mathcal{M}}{2})}{\pi\left(n - \frac{\mathcal{M}}{2}\right)}, & n \neq \frac{\mathcal{M}}{2} \\ \frac{\omega_c}{\pi}, & n = \frac{\mathcal{M}}{2} \end{cases} \quad (5)$$

$$w\left[n - \frac{\mathcal{M}}{2}\right] = 0.5 - 0.5Cos\left(\frac{2\pi n}{\mathcal{M}}\right) \quad (6)$$

$$h\left[n - \frac{\mathcal{M}}{2}\right] = h_{ideal}\left[n - \frac{\mathcal{M}}{2}\right] * w\left[n - \frac{\mathcal{M}}{2}\right], \quad n = 0, 1, 2, \dots, \mathcal{M} \quad (7)$$

where, h is impulse response of the filter, w is the window function, ω_c is the cutoff frequency, $\mathcal{M}$ is the filter order and n is the sample.

Down-sampling is the process of discarding samples from the original signal to achieve a lower sample rate. Equation (8) is used for down-sampling process.

$$y[n] = x[nM], \quad n = 0, 1, 2, \dots, L-1 \quad (8)$$

where, $y[n]$ is the output signal, $x[n]$ is the input signal, n is the sample, L is the length of the signal and M is the down-sampling factor.

Silence removal involves segmenting the audio signal into frames and applying a threshold below which the maximum value of any frame is considered silent [11]. Equations (9, 10) are used for silence removal process.

$$Total\ no.of\ Frames = \frac{Signal\ Length}{Frame\ Length} \quad (9)$$

$$\max Value\ of\ each\ Frame > Threshold \quad (10)$$

Trimming audio signals involves removing undesired segments to make them suitable for classification. Equation (11) is employed for this purpose.

$$Segment\ Length = Segment\ Duration * Sampling\ Frequency \quad (11)$$

Finally, theses 52920 samples of each audio file are rescaled to ±1 scale by using (12).

$$Y_{rescaled} = l + \left[\frac{Y - Y_{min}}{Y_{max} - Y_{min}}\right](u - l) \quad (12)$$

where, Y is original signal and $Y_{rescaled}$ is rescaled signal in $[l, u]$ range.

2.3. Hybrid Feature Extraction Technique

The proposed hybrid feature extraction technique combines multiple approaches to extract features from a signal. It aims to convert data into a more compact form while preserving relevant information. Initially, the 1-D time domain signal is transformed into a 2-D time-frequency domain, resembling a grayscale image. This image captures various patterns reflective of the underlying signal information. Leveraging the Gray Level Co-Occurrence Matrix (GLCM), relationships between pairs of pixels within the grayscale image are analyzed to compress the matrix into a concise representation. Ultimately, 22 distinct features are extracted from this process.

2.3.1. Continuous Wavelet Transform (CWT)

CWT utilizes a Complex Morlet Wavelet to transform a 1D-Signal into a 2D-Signal [12]. This waveform is characterized by its short-lived, wave-like oscillation, which is temporally localized. Equation (13) specifies the creation of the Complex Morlet Wavelet.

$$\Psi(x) = \frac{1}{\sqrt{\pi f_{td}}}.e^{i(2\pi f x)}.e^{-x^2/f_{td}} \quad (13)$$

where, f_{td} is the time-decay factor while f is the frequency of Complex Morlet Wavelet.

CWT overcomes the limitation of STFT by providing control over time and frequency resolution across different frequencies. Wavelets provide detailed temporal information while offering less frequency resolution, and vice versa. This property is beneficial for analyzing non-stationary signals, which undergo temporal variations [12]. Equation (14) is used for 1D-Signal to 2D-Signal transformation.

$$T(t,f) = \sum_{k=0}^{L-1} y(k).\Psi\left(\frac{n-k}{s}\right) \quad (14)$$

where $T(t,f)$ is the transformed 2D-Signal, $y(k)$ is sound signal, $\Psi(n)$ is the Complex Morlet Wavelet at specific signal frequency defined by scale s, k is the time point, n is the sample and L is the length of output signal.

In this way, 1-D time domain signal with dimensions of 52920x1 will be converted to 2-D time frequency domain signal with dimensions of 21x52920.

2.3.2. Direct Conversion of 1D-Signal to 2D-Signal

Frame-based analysis facilitates the direct conversion of a 1D-Signal into a 2D-Signal. This technique involves dividing the 1D-Signal into non-overlapping frames, treating each frame as an independent entity. The result is a 2D matrix where one dimension represents time, and the other represents squared amplitude. Equation (15) mathematically describes this process.

$$Total\ no.of\ Frames = \frac{Signal\ Length}{Frame\ Length} \quad (15)$$

2.3.3. Short Time Fourier Transform (STFT)

STFT is a technique used to investigate the frequency components of a signal as it evolves over time. It employs overlapping windows of fixed duration, typically lasting tens to hundreds of milliseconds, shifted along the signal with a specified hop size. Each windowed segment undergoes a Fourier transform to compute its frequency content. The resulting frequency contents are combined to create a time-frequency representation of the signal, often visualized as a spectrogram. Equations (15 – 18) are used for the transformation.

$$m = \frac{framesize}{2} + 1 \quad (16)$$

$$k = \frac{samples - framesize}{hopsize} + 1 \quad (17)$$

$$S(m,k) = \sum_{n=0}^{N-1} x(n+mH).w(n).e^{-i2\pi n\frac{k}{N}} \quad (18)$$

where, $S(m,k)$ is the 2-D time-frequency matrix containing m frames and k frequency bins, $x(n+mH)$ is the individual frame, $w(n)$ is the window function.

2.3.4. Gray Level Co-occurrence Matrix (GLCM)

Converting a 1D-Signal into a 2D-Signal represents an image, typically grayscale, where pixel values range from 0 to 1, encompassing various intensities. Each pixel contributes to the overall visual representation, and the spatial relationships between pixel intensities are further analyzed using GLCM. GLCM quantifies the frequency of pairs of pixel values occurring at specific spatial relationships within the image. This matrix offers insights into patterns and structures within the image, with each element denoting the probability of encountering a specific pixel pair at a given spatial association. By analyzing GLCM, intricate spatial patterns within the grayscale image can be captured and quantified.

2.3.5. Statistical Features

Once normalized GLCM is formed then it is used to calculate 22 distinct statistical features using (19 - 40). These features indicate the existence of all co-occurrences of gray-level values within the image. Table 2 lists all 22 extracted features [13-15].

Table 2. Statistical features.

Features	Formula	
Autocorrelation	$\sum_{i=0}^{N}\sum_{j=0}^{M}(ij)P(i,j)$	(19)
Contrast	$\sum_{i}^{N}\sum_{j}^{M}\lvert i-j\rvert^2\,\mathrm{P}(i,j)$	(20)
Correlation 1	$\sum_{i}^{N}\sum_{j}^{M}\frac{(i-\mu_i)(j-\mu_j)\mathrm{P}(i,j)}{\sigma_i\sigma_j}$	(21)
Correlation 2	$\frac{\sum_i^N\sum_j^M(ij)P(i,j)-\mu_x\mu_y}{\sigma_x\sigma_y}$	(22)
Cluster Prominence	$\sum_{i}^{N}\sum_{j}^{M}(i+j-\mu_x-\mu_y)^4P(i,j)$	(23)
Cluster Shade	$\sum_{i}^{N}\sum_{j}^{M}(i+j-\mu_x-\mu_y)^3\mathrm{P}(i,j)$	(24)
Dissimilarity	$\sum_{i}^{N}\sum_{j}^{M}\lvert i-j\rvert\,\mathrm{P}(i,j)$	(25)
Energy	$\sum_{i}^{N}\sum_{j}^{M}\mathrm{P}(i,j)^2$	(26)
Entropy	$-\sum_{i}^{N}\sum_{j}^{M}P(i,j)\log(\mathrm{P}(i,j))$	(27)
Inverse Difference	$\sum_{i}^{N}\sum_{j}^{M}\frac{\mathrm{P}(i,j)}{1+\lvert i-j\rvert}$	(28)
Homogeneity	$\sum_{i}^{N}\sum_{j}^{M}\frac{\mathrm{P}(i,j)}{1+(i-j)^2}$	(29)
Maximum Probability	$\max_{i,j}\mathrm{P}(i,j)$	(30)
Variance	$\sum_{i}^{N}\sum_{j}^{M}(i-\mu)^2P(i,j)$	(31)
Sum Average	$\sum_{i=2}^{2N_g}iP_{x+y}(i)$	(32)
Sum Variance	$\sum_{i=2}^{2N_g}\left(i-\left(\sum_{i}^{N}\sum_{j}^{M}\mathrm{P}(i,j)^2\right)\right)^2P_{x+y}(i)$	(33)
Sum Entropy	$-\sum_{i=2}^{2N_g}\mathrm{P}_{x+y}(i)\log(\mathrm{P}_{x+y}(i))$	(34)
Difference Variance	$\sum_{i=0}^{N_g-1}i^2P_{x-y}(i)$	(35)
Difference Entropy	$-\sum_{i=0}^{N_g-1}\mathrm{P}_{x-y}(i)\log(\mathrm{P}_{x-y}(i))$	(36)
Information Measures of Correlation 1	$\frac{HXY-HXY1}{\max(HX,HY)}$	(37)
Information Measures of Correlation 2	$(1-e^{-2.0(HXY2-HXY)})^{1/2}$	(38)
Inverse Difference Normalized	$\sum_{i,j=1}^{G}\frac{C_{ij}}{1+\lvert i-j\rvert^2/G}$	(39)
Inverse Difference Moment Normalized	$\sum_{i,j=1}^{G}\frac{C_{ij}}{1+(i-j)^2/G^2}$	(40)

2.4. Feature Reduction through PCA

PCA is a method that extracts the most relevant information from high-dimensional data and stores it in a lower-dimensional space. PCA involves several steps to reduce features from 22 to 7.

The first step is important because it ensures that the data is centralized around the origin of the coordinate system, as in (41).

$$X_c = \frac{X-\bar{X}}{\sigma} \tag{41}$$

where, X_c is the centered data, X is the original data, $\bar{X}$ is the mean vector calculated across each feature and σ is the standard deviation.

After centering the dataset, PCA computes the covariance matrix of the centered data, as in (42).

$$C = \frac{1}{n-1}X_c^TX_c \tag{42}$$

where, C is the covariance matrix, n is the number of observations, X_c^T is the transpose of the centered data matrix and X_c is the centered data matrix.

By analyzing this covariance matrix, PCA identifies eigenvectors and eigenvalues of the covariance matrix. The eigenvectors, sorted by their corresponding eigenvalues, are chosen as the principal components as defined in (43).

$$Cv_i = \lambda_i v_i \tag{43}$$

where, v_i is the ith eigenvector, λ_i is the corresponding eigenvalue and C is the covariance matrix.

Once the principal components have been selected, PCA transforms the original data onto these components, as in (44).

$$X_{proj} = X_cV_k \tag{44}$$

where, X_{proj} is the projected data onto the k principal components, V_k is the matrix containing the top k eigenvectors.

2.5. Machine Learning Classifiers

Machine learning classification is a type of algorithm that categorizes input data into predefined classes based on patterns and relationships found in training data. The aim of classification is to build a predictive model that can accurately assign labels to new, unseen data.

2.5.1. Decision Tree (DT)

For classifications, a supervised method known as a DT can be employed. A decision based on a feature is represented by an internal node, an outcome by a branch, and the predicted class label by a leaf node in this hierarchical structure that looks like an upside-down tree. To find the optimal feature and split point for data partitioning, the decision tree technique utilizes Gini impurity and information gain. To determine the Gini impurity, equation (45) is employed.

$$I_G(t) = 1 - \sum_{i=1}^{c} {p_i}^2 \quad (45)$$

where, c is the number of classes and p_i is the proportion of samples in class i at node t. Equation (46) is used for calculating the information gain.

$$IG(t, t_L, t_R) = I_G(t) - \frac{N_L}{N} I_G(t_L) - \frac{N_R}{N} I_G(t_R) \quad (46)$$

where, N_L and N_R are the number of samples in the left and right child nodes, respectively. N is the total number of samples in the parent node.

2.5.2. K-Nearest Neighbours

A supervised machine learning method known as KNN is employed for classification purposes. Because it is instance-based and non-parametric, the method learns the full training dataset without assuming anything about the distribution of the underlying data. The most common distance metric used is Euclidean distance, as in (47).

$$D(x, x') = \sqrt{\sum_{i=1}^{d} (x_i - x_i')^2} \quad (47)$$

where, D is the Euclidean distance between two points x and x' in a d-dimensional feature space.

The algorithm then selects the K nearest neighbors to the new instance based on the calculated distances using (48).

$$K = \sqrt{n} \quad (48)$$

where, n is the number of samples in the dataset.

For classification, the algorithm assigns the class label to the new instance based on the majority class label among its K nearest neighbors as in (49).

$$\hat{y} = argmax_{c \in \{c_1, c_2, \ldots, c_k\}} \sum_{i=1}^{k} 1(y_i = c) \quad (49)$$

where, $\hat{y}$ is the predicted class label for x and 1 is the indicator function that returns 1 if the condition inside is true while 0 otherwise.

2.5.3. Naïve Bayes (NB)

Classification tasks are handled by NB, a probabilistic machine learning method. It relies on Bayes' theorem and the "naive" belief that features are independent. During training, the algorithm takes the training dataset's class labels as input and uses them to learn the feature distribution and conditional probability. Based on the class label, the algorithm operates under the "naive" premise that all features are completely separate from one another. The algorithm calculates the posterior probability of each class given the features of the new instance using Bayes' theorem, as in (50).

$$P(y|x_1, x_2, \ldots, x_n) = \frac{P(y).P(x_1, x_2, \ldots, x_n|y)}{P(x_1, x_2, \ldots, x_n)} \quad (50)$$

where, $(x_1, x_2, \ldots, x_n)$ is the feature vector, y is the class label with two possible classes. Equation (51) is used to find class prior probability.

$$P(y) = \frac{number\ of\ samples\ with\ x_i\ and\ class\ y}{number\ of\ samples\ with\ class\ y} \quad (51)$$

Refer to (52) for likehood estimation for continuous features.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi {\sigma_y}^2}} e^{\left(-\frac{(x_i - \mu_y)^2}{2{\sigma_y}^2}\right)} \quad (52)$$

where, μ_y is the mean of feature x_i for class y and ${\sigma_y}^2$ is the variance of feature x_i for class y.
Refer to (53) for posterior probability.

$$P(y|x_1, x_2, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y) \quad (53)$$

where, n is the number of features.
And finally, (54) is used for decision rule.

$$\hat{y} = argmax_y P(y|x_1, x_2, \ldots, x_n) \quad (54)$$

where, $\hat{y}$ is the predicted class.

2.5.4. Support Vector Machines

The goal of SVM, a type of supervised machine learning method, is to maximize the margin—the distance between the hyperplane and the nearest data points—while classifying the data points into distinct groups. In the case of linearly separable data, SVM finds the hyperplane that maximizes the margin and correctly classifies all training data points, as in (55).

$$f(x) = w^T x + b \quad (55)$$

where, $f(x)$ is the decision function, w is the weight factor, x is the input feature vector and b is the bias term.
Classification rule is defined as in (56, 57).

$$min_{w,b} \frac{1}{2} \|w\|^2 \quad (56)$$

Subject to:

$$y_i(w^T x_i + b) \geq 1 \quad \text{for i = 1, 2, …, m} \quad (57)$$

where, m is the number of training examples, x_i is the i-th training feature vector, y_i is the corresponding class label (y_i=1 for positive class, y_i=-1 for negative class)

SVM aims to find the hyperplane using (58 – 60) that not only separates the data but also maximizes the margin between classes. This is achieved by solving an optimization problem that involves minimizing the classification error while maximizing the margin. SVM also includes a regularization parameter (C) that controls the trade-off between maximizing the margin and minimizing the classification error.

$$min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \xi_i \quad (58)$$

Subject to:

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad \text{for } i = 1, 2, \ldots, m \tag{59}$$

$$\xi_i \geq 0 \quad \text{for } i = 1, 2, \ldots, m \tag{60}$$

where, C is the regularization parameter and ξ_i slack variables that allow for misclassification.

2.5.5. Artificial Neural Networks (ANN)

ANN is a computer model that takes its cues from how the human brain's neural networks work. Artificial neurons, often called perceptrons, are the fundamental units of artificial neural networks (ANNs). Output of a single-layer perceptron is defined in (61).

$$y_j = f(w_{1j}x_1 + w_{2j}x_2 + w_{3j}x_3 + \cdots + w_{nj}x_n + b) \tag{61}$$

where, $(x_1, x_2, x_3, \ldots, x_n)$ are the input features, $(w_{1j}, w_{2j}, w_{3j}, \ldots, w_{nj})$ are the corresponding weights, b is the bias and f is the activation function.

2.6. Performance Evaluation of Classifiers

Classifier performance evaluation assesses the effectiveness of a classification model on a dataset, offering insights into its strengths and weaknesses. These metrics aid in model selection, parameter tuning, and optimization. Classifiers results are shown in a table which is known as confusion matrix, as shown in Fig. 2. Classifiers performance parameters such as accuracy, sensitivity, specificity and precision can be extracted from confusion matrix.

		Predicted Class
Actual Class	"TP" True Positive	"FN" False Negative
	"FP" False Positive	"TN" True Negative

Figure 2. Confusion matrix.

Accuracy, a common metric for classification model evaluation, measures the proportion of correctly classified instances out of the total dataset, as in (62).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{62}$$

The sensitivity of a classifier is defined as the % of true positives it can accurately detect within a dataset, as shown in (63).

$$Sensitivity = \frac{TP}{TP + FN} \tag{63}$$

For each instance in the negative class, specificity measures the fraction of true negative predictions, as shown in (64).

$$Specificity = \frac{TN}{FP + TN} \tag{64}$$

Precision measures the proportion of TF predictions among all positive predictions made by the classifier, as in (65).

$$Precision = \frac{TP}{TP + FP} \tag{65}$$

3. Results and Discussion

Intel® Core™ i5-8350U Processor and 24 GB RAM based hardware is used to prepare all the results in MATLAB environment. This section will cover the results and discussion on the proposed methodology.

3.1. Pre-Processing

Table 3 outlines key parameters of the original dataset, including min-max range, audio channel count, sample rate, time duration, and peak amplitude for both original and pre-processed files. The original values demonstrate variability across files, highlighting the need for standardization to enable further processing, as indicated by the pre-processed values.

Table 3. Original and pre-processed parameters of audio dataset.

Key Parameters	Original	Methodology	Pre-Processed
Min-Max Range	$\pm 33 \times 10^3$ to $\pm 2.2 \times 10^9$	Rescaling	$\pm 33 \times 10^3$
Audio Channels	1 - 2	Averaging	1
Sample Rate (Hz)	8000 - 96000	Resampling	44100
Silence Regions	Present	Thresholding	Absent
Sound Duration (sec)	0.05 - 28.54	Trimming	1.2
Peak Amplitude (V)	Variable	Rescaling	±1

The pre-processing details of two emergency sounds and two non-emergency sounds is shown in Fig. 3. It addresses various parameters, including rescaling, stereo-to-mono conversion, sample rate adjustment, silence removal, trimming and rescaling of trimmed signals. The min-max rescaling, shows the amplitude of the signal with respect to the bit depth, is rescaled to $\pm 33 \times 10^3$ which represents 16 bits per sample bit depth. Stereo sound, characterized by two audio channels (as shown in Fig. 3(b) and Fig. 3(c)) is converted to mono sound. Resample each audio signal to 44100 samples per second (SPS). Silence regions, identified by a line of zero amplitude in original audio, are removed from the signal and replaced with updated samples by thresholding. Finally, the audio duration is trimmed to 1.2 seconds, and the overall signal within this duration is normalized to ±1 amplitude.

Fig. 4 displays original and pre-processed 1D-Sigals. Prior to pre-processing, the amplitude and duration of each audio sample vary. However, after completing all pre-processing procedures the amplitude range is constrained within ±1V and the time duration is standardized to 1.2 seconds. The variability of emergency sound data within the overall audio dataset is evident in its signal-to-noise ratio (SNR). Fig. 4(c) shows that emergency sounds have below 0 dB SNR reading.

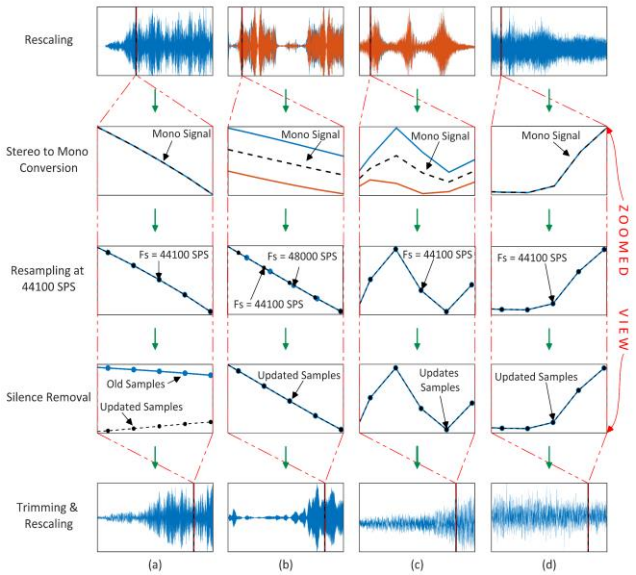

Figure 3. Pre-processing details of (a,b) emergency sounds; (c,d) non-emergency sounds.

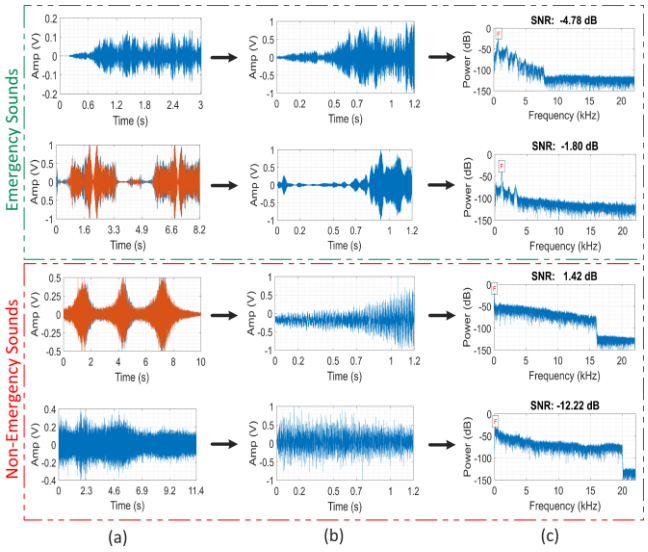

Figure 4. Original and pre-processed 1D-signals (a) original (b) pre-processed (c) SNR of pre-processed signal.

3.2. Hybrid Feature Extraction Technique

The proposed research uses a Complex Morlet Wavelet to convert 1-D signals into 2-D signals, followed by GLCM. Statistical features are then extracted from GLCM to distinguish emergency sounds from non-emergency sounds.

3.2.1. CWT+GLCM

Using Complex Morlet Wavelet, the audio signal is transformed from the time domain to the time-frequency domain, as illustrated in Fig. 5. Fig. 5(a) displays the 1-D representation of the signal in the time domain, while Fig. 5(b) represents the same signal in the time-frequency domain as a 2-D image. In the 2-D signal representation, the duration remains 1.2 seconds, but frequency components from 500 Hz to 2000 Hz are emphasized. The signal magnitude is squared and normalized to a range of 0 to 1 for further processing, as shown in Fig. 5(b).

The 2D-Signal of 21x52920 forms an image. The pixel values of this image represent squared magnitudes. Initially ranging from 0 to 1, the pixel values are normalized to values from 1 to 8. Finally, an 8x8 gray level co-occurrence matrix is generated, as presented in Fig. 6. Fig. 6(a) depicts the 2D signal, Fig. 6(b) illustrates the normalized 2D signal, and Fig. 6(c) represents the calculated gray level co-occurrence matrix derived from the 2D-Signal.

From the normalized GLCM, a set of 22 distinct features is extracted, encompassing statistical properties and spatial relationships inherent in the signal. These features serve as inputs for classification models, facilitating emergency sound categorization. The feature extraction process is depicted in Fig. 7, where Fig. 7(a) shows the 8x8 GLCM matrix, Fig. 7(b) illustrates the normalized GLCM matrix, and Fig. 7(c) signifies the resulting 1x22 feature set.

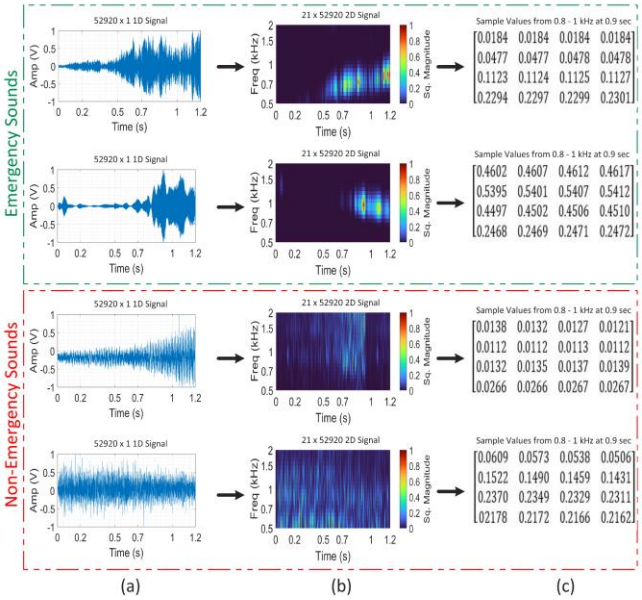

Figure 5. Transformation using Complex Morlet Wavelet (a) 1D-signal (b) Graphical form of 2D-signal (c) Matrix form of 2D-signal.

3.2.2. 2D+GLCM

The 2D+GLCM technique directly transforms the 1D signal into a 2D signal by segmenting it into frames, resulting in a 2D matrix where each column represents a frame. This matrix is then normalized to 1 to 8 levels for GLCM computation, followed by extraction of 22 statistical attributes from the computed GLCM. This process is depicted in Fig. 8 where Fig. 8(a) shows the 1D time domain signal, Fig. 8(b) illustrates the transformation of the 1D signal into a 2D signal using the 2D+GLCM method, Fig. 8(c) displays the 8x8 GLCM derived from the 2D signal, and Fig. 8(d) presents the resulting 22-feature set extracted from the GLCM.

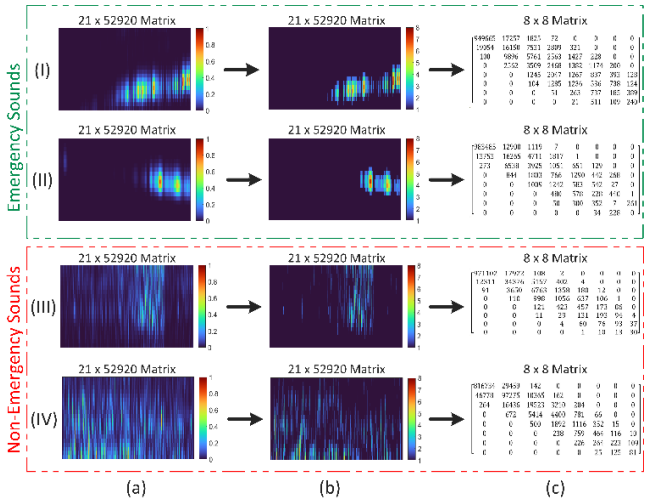

Figure 6. Convert 2D-signal into GLCM (a) 2D-signal (b) Normalized 2D-signal (c) GLCM matrix (8×8).

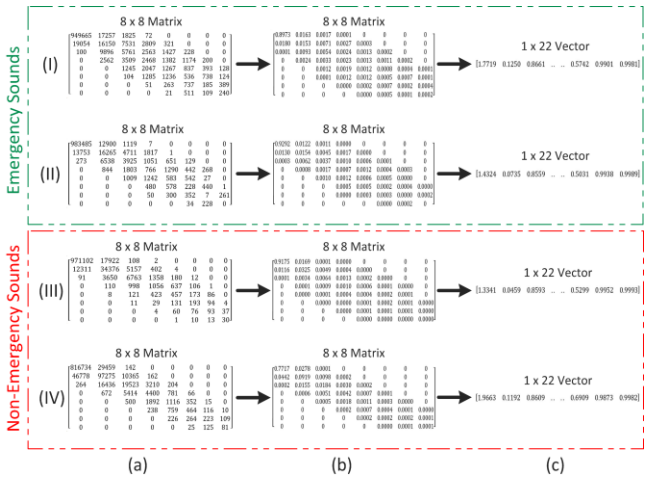

Figure 7. Feature extraction from co-occurrence matrix (a) GLCM (b) Normalized GLCM (c) Feature Set.

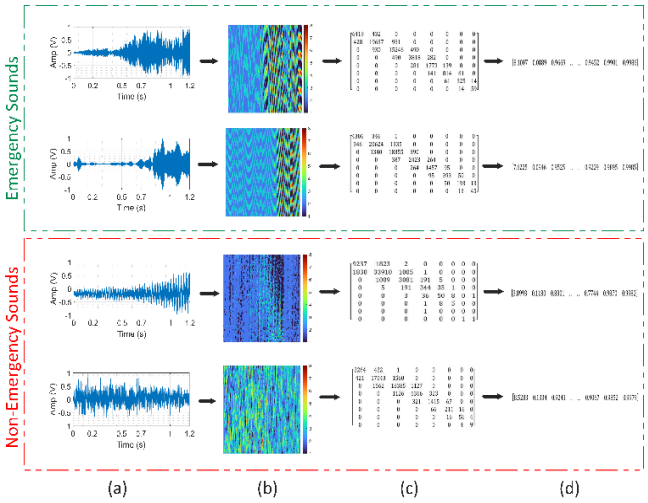

Figure 8. Feature extraction using 2D+GLCM (a) 1D-signal (b) 2D-signal (c) GLCM (d) Feature set.

3.2.3. STFT+GLCM

In another technique named STFT+GLCM, the 1D signal undergoes a transformation into a 2D Signal, as shown in Fig. 9. Fig. 9(a) displays the 1-D time domain signal. Fig. 9(b) displays the transformation of the 1D-Signal into a 2D-Signal through STFT+GLCM method. Fig. 9(c) displays the 8x8 GLCM derived from the 2D Signal and Fig. 9(d) displays the resultant 22 feature set extracted from the GLCM.

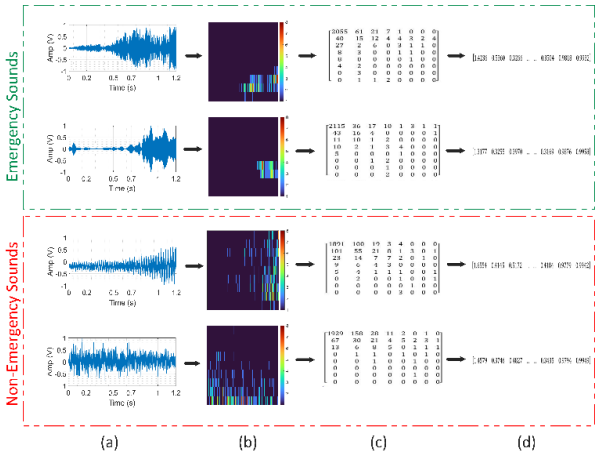

Figure 9. Feature extraction using STFT+GLCM (a) 1D-signal (b) 2D-signal (c) GLCM (d) Feature set.

3.3. Classifiers Performance based on Feature Extraction Technique

For classification, the dataset is divided into train and test sets, with 80% allocated to training and 20% to testing. Five classifiers are employed, and their overall performance is evaluated through 5-fold cross-validation. As 22 statistical features are derived from each audio, an input vector of size 1834x22 is obtained from the dataset. Out of this, 80% (1468x22) is used for training and validation, while the remaining 20% (366x22) is reserved for testing the classifiers' performance.

Table 4 demonstrates the output of classifiers on hybrid feature extraction techniques compared to prior techniques. In the confusion matrix, SVM correctly classifies 343 out of 366 audio samples, with 23 misclassifications. Among these, false positives account for 17 samples and false negatives for 6 samples.

Table 4. Confusion matrix comparison of classifiers on feature extraction techniques.

Classifiers / Techniques	Hybrid Technique (CWT+GLCM)		Prior Technique#1 (2D+GLCM)		Prior Technique#2 (STFT+GLCM)	
	"TP"	*"FP"*	*"TP"*	*"FP"*	*"TP"*	*"FP"*
	"FN"	*"TN"*	*"FN"*	*"TN"*	*"FN"*	*"TN"*
DT	174	19	141	40	173	20
	17	156	52	133	13	160
KNN	165	28	128	65	171	22
	9	164	40	133	10	163
NB	120	73	138	55	154	39
	30	143	45	128	20	153
SVM	176	17	136	57	173	20
	6	167	30	143	11	162
ANN	178	15	142	51	172	21
	9	164	30	143	8	165

Table 5 compares classifiers, revealing SVM's highest accuracy of 93.7% with the hybrid feature extraction technique. This indicates that emergency sound representation with just 22 hybrid features is effective when SVM employs the linear kernel.

Table 5. Accuracy comparison of classifiers on feature extraction techniques.

Classifiers / Techniques	Hybrid Technique (CWT+GLCM)	Prior Technique#1 (2D+GLCM)	Prior Technique#2 (STFT+GLCM)
DT	90.2	74.9	91.0
KNN	89.9	71.3	91.3
NB	71.9	72.7	83.9
SVM	**93.7**	76.2	91.5
ANN	93.4	77.9	92.1

Table 6 presents SVM classifier's performance on hybrid features with different kernel functions. The linear kernel achieves the highest accuracy among linear, polynomial, and RBF kernels.

Table 6. SVM kernel comparison with hybrid feature extraction technique.

SVM Kernel	"Confusion Matrix" "TP" "FP" "FN" "TN"	Accuracy (%)
Linear	**176 17 6 167**	**93.7**
Polynomial	175 19 9 163	92.3
Radial Basis Function (RBF)	175 19 10 162	92.1

A comprehensive evaluation of these metrics reveals the classification performance across all three feature extraction techniques, as illustrated in Fig. 10. The most favorable results emerge when employing features extracted with hybrid technique, indicating an impressive accuracy rate of 93.7% through the SVM model. Prior techniques provides maximum accuracy rate of 76.2% and 91.5% with SVM classifier.

Fig. 11 illustrates the computational time of the SVM classifier with a linear kernel across three techniques. It is evident that prior techniques 1 & 2 require more computational time compared to the hybrid technique. Specifically, prior technique#1 takes 24.2 ms longer time and prior technique#2 requires 4.8 ms more time for detecting emergency sounds. This implies that the hybrid feature extraction technique represents emergency sounds in a more concise manner, thus improving the efficiency of the classifier.

The hybrid feature extraction technique generates 22 features for each sound, which are then subjected to dimensionality reduction through PCA. PCA reduces the features from 22 per sound to 7 per sound. Fig. 12 shows the graph of principal components versus the percentage of explained variances. It reveals that 7 principal components encapsulate 99.9% of the information from the hybrid features. Consequently, it is suggested that the characteristics inherent in emergency sounds can be effectively represented by these 7 principal components.

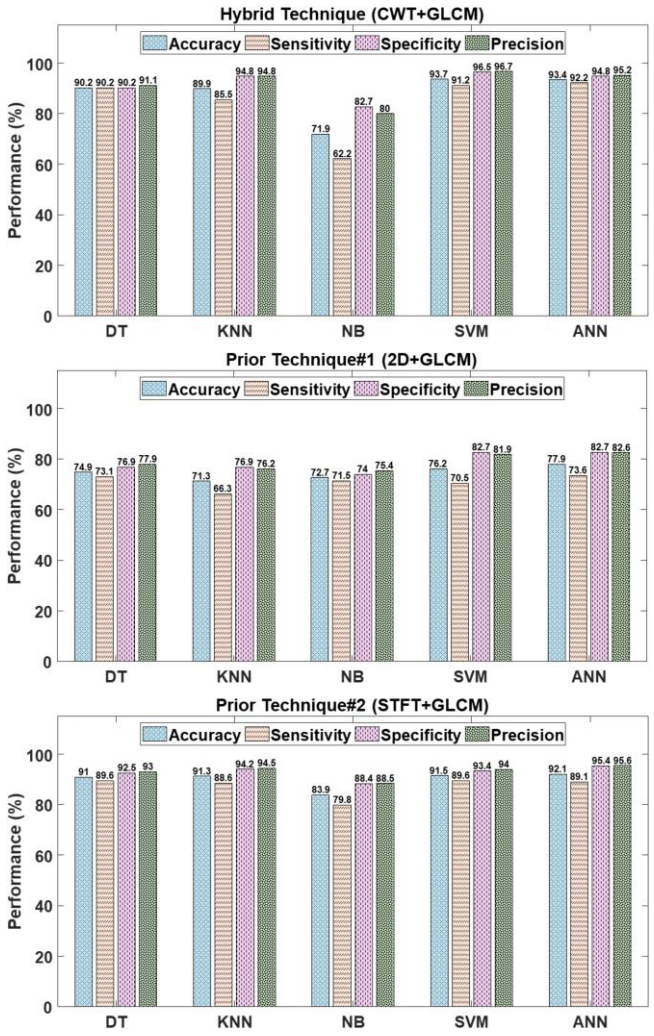

Figure 10. Classifiers performance evaluation comparison of hybrid feature extraction technique with prior techniques.

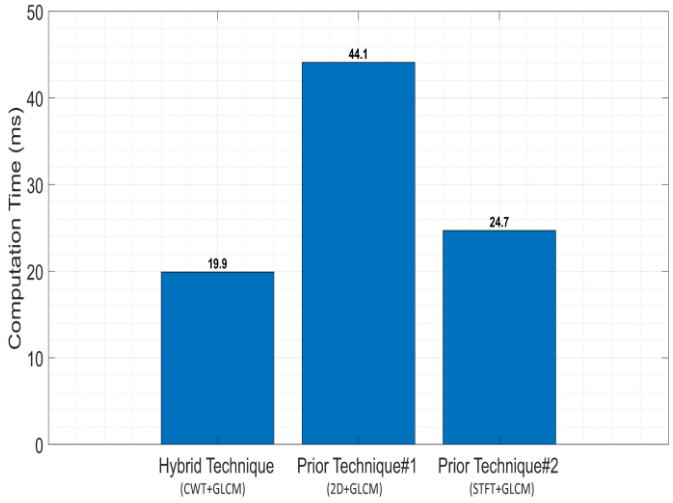

Figure 11. Computation time comparison of hybrid feature extraction technique with prior techniques.

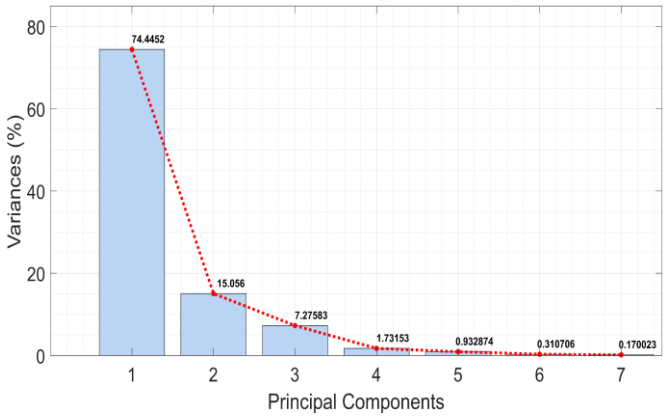

Figure 12. Principal components versus variances graph.

Table 7 presents the classifiers results on testing data with and without applying PCA. Without PCA, the confusion matrix of SVM indicates that 343 out of 366 audio samples are correctly classified, with 23 misclassifications. However, with PCA, 344 samples are correctly classified, and 22 samples are misclassified.

Table 7. Confusion matrix of classifiers on hybrid feature extraction technique with and without PCA.

Classifiers / Hybrid Technique	Without PCA (CWT+GLCM) "TP" "FP" "FN" "TN"	With PCA (CWT+GLCM+PCA) "TP" "FP" "FN" "TN"
DT	174 19 17 156	176 17 14 159
KNN	165 28 9 164	164 29 9 164
NB	120 73 30 143	164 29 11 162
SVM	176 17 6 167	177 16 6 167
ANN	178 15 9 164	182 11 12 161

Table 8 displays the accuracy scores attained by each classifier. Remarkably, the accuracy of the SVM model has increased from 93.7% to 94%. This enhancement implies that the hybrid feature extraction technique with PCA efficiently extracts pertinent features while retaining the majority of crucial information from the feature set.

Table 8. Classifiers accuracy comparison of hybrid feature extraction technique with and without PCA.

Classifiers / Hybrid Technique	Without PCA (CWT+GLCM)	With PCA (2D+GLCM+PCA)
DT	90.2	91.5
KNN	89.9	89.6
NB	71.9	89.1
SVM	93.7	**94.0**
ANN	93.4	93.7

Fig. 13 depicts the performance evaluation of all classifiers using hybrid feature extraction techniques, with and without applying PCA. The accuracy of the SVM classifier has increased from 93.7% to 94%, along with a rise in sensitivity from 91.2% to 91.7%. Additionally, the accuracies of other classifiers have also improved, except for KNN. This indicates that the proposed hybrid feature extraction technique with PCA enhances the classifiers' accuracy.

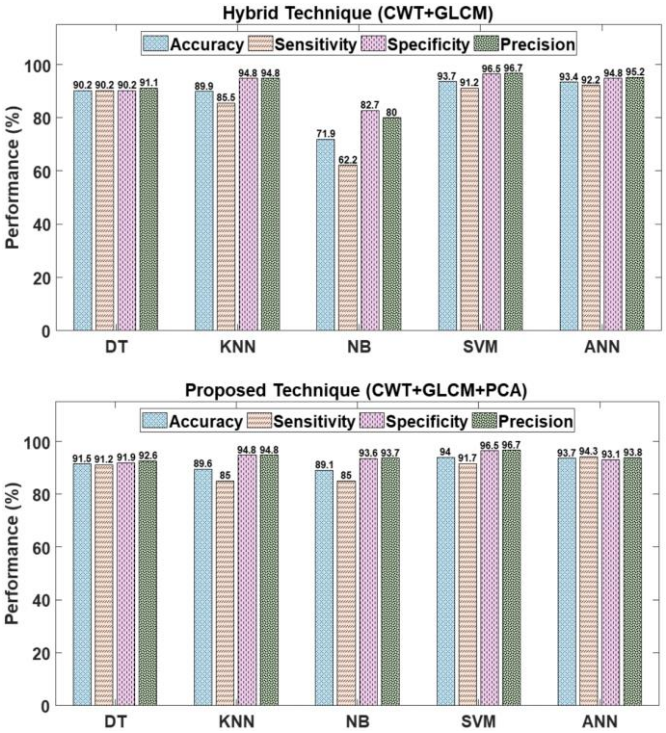

Figure 13. Classifiers performance evaluation comparison of hybrid feature extraction technique with and without PCA.

Fig. 14 illustrates the computation time comparison of the hybrid feature extraction technique with and without PCA. It shows a notable reduction of 16.4 ms in computation time when using PCA with the SVM classifier. This reduction indicates an improvement in the classifier's efficiency when operating on a reduced feature set. Focusing on the most relevant features enables classifiers to achieve better generalization and predictive performance.

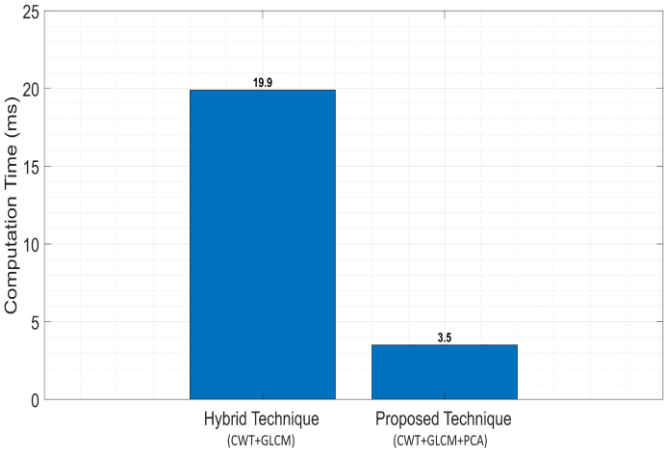

Figure 14. Computation time comparison of hybrid feature extraction technique with and without PCA.

4. Conclusion and Future Work

In this research, a hybrid feature extraction technique is proposed for self-driving cars to detect emergency sounds among various road noises. The proposed technique uses a combination of CWT and GLCM along with PCA to extract useful features from the emergency sounds dataset. The dataset has a vast variety of emergency sounds within different noise conditions. Five classifiers have been used to evaluate the performance of these classifiers on features extracted through the proposed technique. Among all the classifiers, on the input length of 1.2 seconds, the outcome of the proposed research has indicated that the SVM with linear kernel shows the highest accuracy rate of 94%. Moreover, the proposed technique also reduces the computational cost of about 20 – 40 milliseconds when compared with prior techniques, which is acceptable for real-time applications such as self-driving cars.

While the current approach yields satisfactory outcomes in self-driving car applications, there remains a need for future research to enhance detection performance while maintaining minimal processing time. Therefore, exploring the integration of deep learning architectures presents an avenue for developing sound classification models that are both robust and adaptive. This is especially pertinent for handling complex datasets in the context of self-driving cars' audio perception.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] J.-J. Liaw, W.-S. Wang, H.-C. Chu, M.-S. Huang, and C.-P. Lu, "Recognition of the ambulance siren sound in Taiwan by the longest common subsequence," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 2013: IEEE, pp. 3825-3828, doi: 10.1109/SMC.2013.653.

[2] J. Schröder, S. Goetze, V. Grützmacher, and J. Anemüller, "Automatic acoustic siren detection in traffic noise by part-based models," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013: IEEE, pp. 493-497, doi: 10.1109/ICASSP.2013.6637696.

[3] Z. Islam and M. Abdel-Aty, "Real-time Emergency Vehicle Event Detection Using Audio Data," *arXiv preprint arXiv:2202.01367,* 2022, doi: https://doi.org/10.48550/arXiv.2202.01367.

[4] H. Sun, X. Liu, K. Xu, J. Miao, and Q. Luo, "Emergency vehicles audio detection and localization in autonomous driving," *arXiv preprint arXiv:2109.14797,* 2021, doi: https://doi.org/10.48550/arXiv.2109.14797.

[5] A. Garg, A. K. Gupta, D. Shrivastava, Y. Didwania, and P. J. Bora, "Emergency Vehicle Detection by Autonomous Vehicle," *International Journal of Engineering Research and Technology (IJERT),* vol. 08*,* no. 05, 2019.

[6] M. Azad, F. Khaled, and M. R. H. Rumman, "An efficient way to convert 1D signal to 2D digital image using energy values," BRAC University, 2018.

[7] A. Sengür, S. Ekici, Y. Akbulut, and T. Kavas, "Time-Frequency Gray Level Co-occurrence Matrix Descriptors for Deception Detection," 2017.

[8] M. Asif, M. Usaid, M. Rashid, T. Rajab, S. Hussain, and S. Wasi, "Large-scale audio dataset for emergency vehicle sirens and road noises," *Scientific data,* vol. 9, no. 1, pp. 599, 2022.

[9] S. Patro and K. K. Sahu, "Normalization: A preprocessing stage," *arXiv preprint arXiv:1503.06462,* 2015.

[10] C. Feldbauer, M. Képesi, K. Witrisal, and E. Rank, "Multirate signal processing," *Graz University of Technology,* vol. 1, no. 3, pp. 1-10, 2005.

[11] M. Asadullah and S. Nisar, "A silence removal and endpoint detection approach for speech processing," *Sarhad University International Journal of Basic and Applied Sciences,* vol. 4, no. 1, pp. 10-15, 2016.

[12] A. Saxena, B. Wu, and G. Vachtsevanos, "A methodology for analyzing vibration data from planetary gear systems using complex Morlet wavelets," in *Proceedings of the 2005, American Control Conference, 2005.*, 2005: IEEE, pp. 4730-4735.

[13] R. M. Haralick, K. Shanmugam and I. Dinstein, "Textural Features for Image Classification," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610-621, Nov. 1973.

[14] L.-K. Soh and C. Tsatsoulis, "Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices," *IEEE Transactions on geoscience and remote sensing,* vol. 37, no. 2, pp. 780-795, 1999.

[15] D. A. Clausi, "An analysis of co-occurrence texture statistics as a function of grey level quantization," *Canadian Journal of remote sensing,* vol. 28, no. 1, pp. 45-62, 2002.