Journal of Computer Science

Research Article

Preventing Deforestation in the Indian Landscape Through Neural Network-Based Intelligence Using Sound Event Detection and Advanced Feature Extraction Techniques

Sallauddin Mohmmad[1,2] and Suresh Kumar Sanampudi[3]

[1]*Department of Computer Science and Engineering, JNTU Hyderabad, India*
[2]*School of Computer Science and Artificial Intelligence, SR University, Warangal, Telangana, India*
[3]*Department of Information Technology, JNTUH University College of Engineering, Nachupally, Kondagattu, Jagtial, Telangana, India*

Article history
Received: 17-03-2025
Revised: 07-08-2025
Accepted: 20-06-2025

Corresponding Author:
Sallauddin Mohmmad
Department of Computer Science and Engineering,
JNTU Hyderabad, India
Email: sallauddin.md@gmail.com

Abstract: Forests play a vital role in maintaining ecological balance, regulating the climate, and conserving biodiversity. However, India's forest landscape has witnessed significant changes between 1980 and 2024 due to deforestation, afforestation, and evolving conservation strategies. To address the challenges associated with forest monitoring, we proposed a model based on Sound Event Detection using a dataset comprising four classes: chainsaw sounds, handsaw sounds, axe-cutting sounds (synthetic), and negative environmental sounds (e.g., birds, animals, wind). The dataset was constructed from publicly available resources, except for the axe-cutting sound class, which was prepared synthetically. The model employed six feature extraction techniques Mel-Spectrogram, Mel-Frequency Cepstral Coefficients (MFCC), Chroma, Spectral Contrast, Tonnetz, and Spectral Bandwidth to capture critical audio characteristics. These features enabled the efficient representation of harmonic content, temporal patterns, and timbre, which were essential for distinguishing between classes. The proposed approach was executed using various deep learning models, including Customized 1D Convolutional Neural Networks (CNN), Bi-directional Convolutional Recurrent Neural Networks (Bi-CRNN), Bi-directional Gated Recurrent Unit-based CRNNs (Bi-GRU-CRNN), AlexNet, and ResNet. The Customized-CNN, implemented using Keras, demonstrated superior performance with an accuracy of 98%. The model's effectiveness was further validated as accuracy increased progressively from 95 to 98% when transitioning from two to six feature extraction clusters.

Keywords: Forest Monitoring, Sound Event Detection, CNN, Feature Extraction, Audio Classification, Deep Learning

Introduction

Forests play a crucial role in sustaining life on Earth by maintaining ecological balance, regulating the climate, conserving biodiversity, and providing resources for millions of people. The loss of forests due to deforestation has extreme consequences, including the exacerbation of climate change, loss of biodiversity, disruption of water cycles, and increased soil erosion (Mondal and Southworth, 2010). Deforestation also directly impacts the livelihoods of communities that depend on forests for their sustenance and cultural practices. However, forests in many parts of the world, particularly in India, face severe threats from deforestation driven by expanding agricultural activities, illegal logging, urbanization, and the growing demand for land and timber (Haq *et al*., 2022; Tewari *et al*., 2014). Technological advancement in the current generation, the forest authorities have implemented various monitoring systems. The Forest Survey of India (2023), under the Ministry of Environment, Forest, and Climate Change, conducted biennial assessments of forest cover using remote sensing technology, which is vital for informed decision making in forest conservation (Singha *et al*., 2024). On the other hand, by using drone technology implementation, governments focused on real-time localized monitoring, assessing forest damage, and tracking wildlife, while IoT devices provide continuous data on forest conditions, assisting in the detection of illegal activities and environmental changes (Buchelt *et al*., 2024; Zhao *et al*., 2019). Additional technologies such as forest fire alert systems, AI, and blockchain are used in forest management. The integration of research, like Machine Learning (ML) and neural networks, provides an innovative solution for forest conservation (Khan and Khan, 2022; Supriya and Gadekallu, 2023). Sound event detection with advanced feature extraction techniques are essential for proper sound classification. By analyzing audio data from forest environments leads to detect sounds of illegal activities such as tree cutting and logging vehicle movement. These feature extraction techniques enhance the precision of deforestation detection. Conducting the research based on this scenario requires a proper dataset and suitable feature extraction techniques (Luo *et al*., 2023; Purwins *et al*., 2019). Acoustic datasets related to forests, environments, and mountains provides resources for studying biodiversity, environmental health, and ecological dynamics. The datasets such as Rainforest Connection (RFCx) and Soundscapes to Landscapes (S2L) provide audio data captured from rainforests and various natural habitats (Latha *et al*., 2022; Quinn *et al*., 2022). These datasets are primarily used to monitor wildlife, identify species based on sound, and study the impacts of climate change. The Australian Acoustic Observatory also contributes recordings from ecosystems such as forests and wetlands, aiding in conservation and ecological studies (Roe *et al*., 2021).

Additionally, general acoustic datasets like DCASE and FreeSound contain various environmental sounds, including forests and outdoors, implemented with machine learning and sound classification practices (Serizel *et al*., 2020). The datasets like the AMMOD contain mountainous environmental sounds such as wind and water sounds in high-altitude regions (Wägele *et al*., 2022). These datasets are well-suited for research in various fields, including forest and landscape ecology. They also play an important role in supporting ecological research and environmental monitoring.

Sound event feature extraction is essential for converting the direct audio signals into a numerical representation that can be used for classification. One common approach involved time-domain features, which are directly derived from the raw waveform of the sound signal (Wang *et al*., 2024). Features like Zero-Crossing Rate (ZCR), Root Mean Square (RMS), and energy are widely used to capture basic characteristics such as intensity and periodicity (Ritts *et al*., 2024). However, many sound events have unique frequency components, which makes frequency-domain features particularly important. Using the STFT the audio signal is converted into its frequency representation, from which features like spectral centroid, spectral roll-off, and MFCCs can be derived. MFCCs, in particular, are extensively used in sound event and speech recognition due to their ability to represent the timbral aspects of sound (Folliot *et al*., 2022).

Most of the recent research combining the time and frequency information through time-frequency representations like spectrograms and wavelet transforms. Spectrograms provide a visual representation of the signal's frequency over time, and their compressed versions, such as log-mel spectrograms, are often used to project the relevant features. The wavelet transform is another powerful method, as it decomposes the signal into multiple resolutions, capturing details across different frequency bands (Serrurier *et al*., 2024; Ayoub Shaikh *et al*., 2022). In recent years, deep learning-based feature extraction methods have become popular. The CNNs and RNNs can automatically learn complex features from raw or pre-processed audio data spectrograms. Pre-trained models like VGG or OpenL3 are also used to generate robust audio features for sound event detections (Panwar *et al*., 2022; Thepade and Chaudhari, 2021). In particular, neural network models excel at recognizing complex sound patterns, making them ideal for forest sound analysis. For instance, the distinctive sound of chainsaws used in tree cutting or the engine noise of logging vehicles can be identified and classified by AI models trained on sound event datasets (Kentsch *et al*., 2020; Singh *et al*., 2023). These systems rely on feature extraction techniques to isolate relevant acoustic features such as frequency, amplitude, and duration, which are then processed through neural networks to differentiate between natural and human-caused events in the forest. Machine learning algorithms continuously improve detection accuracy by learning from labeled and unlabeled data, enabling the system to adapt to new patterns of illegal activity. Sound Event Detection becomes an important element in the fight against deforestation, providing forest rangers and conservationists with real-time data on illegal logging activities. These detection models are for quicker response times and more effective enforcement of forest protection. This research explores the use of ML, and neural networks for preventing deforestation in the Indian landscape.

By focusing on sound event detection, specifically tree-cutting sounds and logging vehicle detection, this study demonstrates how advanced techniques can be applied to monitor forest and human activities. Using real-time audio analysis, detection of tree-cutting sounds becomes best solution to protect India's forests, preserve biodiversity, and ensure the long-term sustainability of these ecosystems.

Motivation

India’s forest landscape has undergone significant changes from 1980 to 2024, driven by deforestation, afforestation efforts, and evolving conservation strategies. During the 1980s, forest cover in India was estimated at around 19% of the total geographical area, with minimal differentiation between dense, moderate, and open forest categories. This period was marked by large-scale deforestation, primarily due to agricultural expansion, infrastructure development, and industrial activities. As a result, dense forests were reduced, and open and degraded lands expanded. Scrublands, often representing degraded forests or sparsely vegetated areas, constituted a small portion of the forest landscape. In the 1990s and early 2000s, India began focusing on forest conservation and afforestation initiatives. The India State of Forest Report (ISFR) 1997 provided more granular data, identifying approximately 10% of India's land area as moderately dense forests, covering 337,600 sq. km. Very dense forests, though sparse, were recorded at about 1.7% of the geographical area. The increasing deforestation trends of the 1980s were countered by government programs such as Joint Forest Management (JFM), aimed at increasing tree cover in degraded areas. During this period, open forests covered 7-8% of India's land, while non-forest land remained high at around 75-80% (Forest Survey of India, 2023).

As India moved into the 2010s, forest conservation efforts began to show results. By 2011, moderately dense forests covered 307,000 sq. km, although this number saw a slight decline to 306,890 sq. km by 2021. Dense forest cover, however, showed a significant improvement, reaching 99,779 sq. km (about 3% of the land area). Open forests also saw an increase during this time, reaching 9.34% of the total land by 2021, reflecting the success of afforestation campaigns and conservation measures. The scrubland ratio remained fairly constant throughout this period, hovering between 1.4-1.5%, indicating a consistent but relatively small portion of degraded lands. Non-forest areas stabilized at approximately 76-77% during the same time. By 2022 and 2023, the forest landscape in India reflected a steady trend. Moderately dense forests continued to cover 9.33% of the country's geographical area, while very dense forests remained at 3% of the total area. Open forests accounted for 9.34% of the land, showing minimal change from 2021. Scrublands persisted at around 1.42%, with non-forest areas still constituting about 76.87% of the total land area. The results are showing that while forest cover has seen modest improvements, particularly in dense and open forests, challenges such as deforestation in northeastern and tribal regions persist, leading to localized declines in forest cover. Figure 1 shows the VDF, MDF, OF, Scrub and Non-Forest area of coverage in various years in the India. Table 1 illustrates the Year-wise MDF, OF, Scrub, VDF and non-forest coverage percentage in India from 2001 to 2021. Figure 2 presents the Very Dense Forest growth and reduction of Moderately Dense Forest reduction comparatively with Open Forest from 2001 to 2021. Figure 3 illustrates the Comparison of Non-Forest land area with Forest Coverage land area in India from 2001 to 2021. Table 1 presents the forest coverage in the India from 2001 to 2021 (Forest Survey of India, 2023).

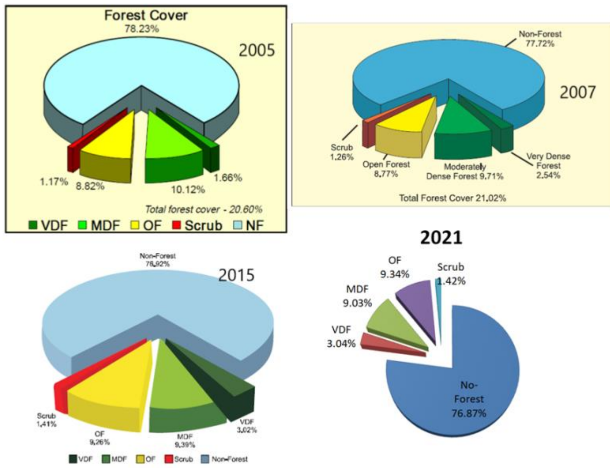


Fig. 1: VDF, MDF, OF, Scrub and Non-Forest area of coverage in various years in the India

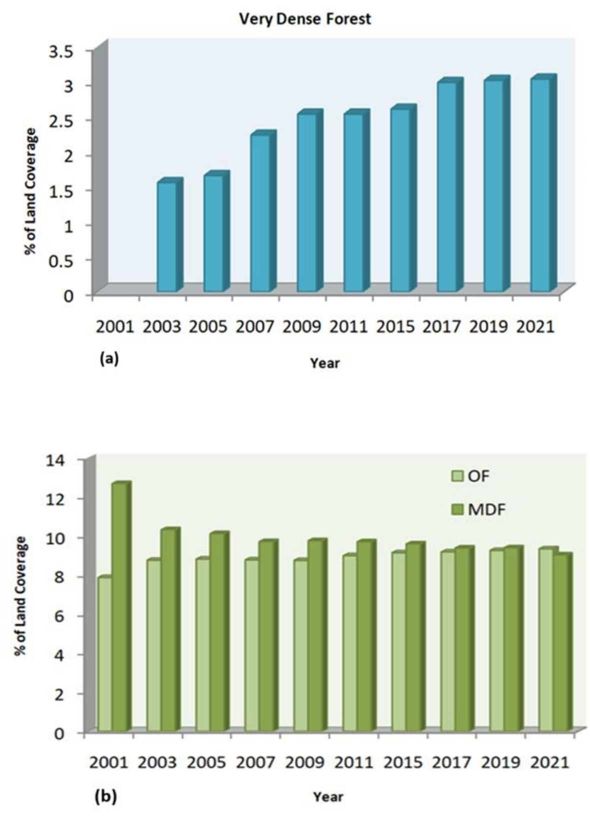


Fig. 2: (a) Very Dense Forest growth ratio from 2001 to 2021 and (b) Reduction of Moderately Dense Forest reduction comparatively with Open Forest from 2001 to 2021

DOI: 10.3844/jcssp.2025.2772.2801

Table 1: Forest Coverage area from 2001 to 2021 in India

Year	Non-Forest	MDF	OF	Scrub	VDF
2001	79.45	12.68	7.87	-	-
2003	78.13	10.32	8.76	1.23	1.56
2005	78.23	10.12	8.82	1.17	1.66
2007	77.72	9.71	8.77	1.26	2.24
2009	77.67	9.76	8.75	1.28	2.54
2011	77.51	9.7	8.99	1.28	2.54
2015	77.4	9.59	9.14	1.26	2.61
2017	77.06	9.38	9.18	1.4	2.99
2019	76.92	9.39	9.26	1.41	3.02
2021	76.87	9.03	9.34	1.42	3.04

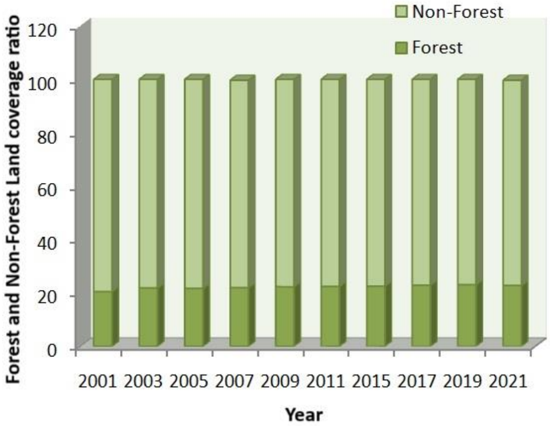


Fig. 3: Comparison of Non-Forest land area with Forest Coverage land area in India from 2001 to 2021

The Indian government has implemented several initiatives to protect and restore forest cover. Key among these is the Green India Mission (GIM), which focuses on afforestation and reforestation activities to enhance forest and tree cover across the country. Over the last five years, approximately 7,552.8 million has been allocated to support afforestation efforts under GIM. Additionally, the Compensatory Afforestation Fund (CAMPA) has been instrumental in financing compensatory afforestation projects, with 553,941.6 million released to state and union territory forest departments over the past five years. The government has also launched the Nagar Van Yojana, aiming to create 600 urban forests and 400 urban gardens by 2024-25 to improve green cover in urban areas. This initiative is funded by CAMPA and has seen the approval of 270 projects with a total cost of 2,386.4 million.

From 2010 to 2015, the Ministry of Environment, Forest and Climate Change (MoEFCC) primarily focused on forest conservation through schemes such as the National Afforestation Programme (NAP), CAMPA, and the Green India Mission. In the fiscal year 2014-15, the MoEFCC allocated 20,430 million for forestry and wildlife, with an emphasis on afforestation, forest management, and wildlife conservation. From 2015 to 2020, the government continued to enhance forest management efforts, with the Green India Mission receiving 3,670 million in 2017–18. CAMPA funds were systematically utilized for compensatory afforestation, and by 2020, government allocations increased to address climate change and manage forests, aligning with international commitments such as the Paris Agreement. For 2018–19, the MoEFCC budget stood at 26,750 million, supplemented by additional CAMPA funds. In the 2020–2025 period, despite the pandemic, environmental protection remained a priority. For 2020-21, the MoEFCC received 28,700 million, emphasizing afforestation and forest conservation. Projections for 2021–2025 show continued investment in forest management, with 25,200 million allocated in 2021–22, driven by commitments made at COP26 and the global push for sustainable development. Budget allocations for key schemes such as the National Afforestation and Eco-Development Board, and the Integrated Development of Wildlife Habitats, also saw incremental increases. Historically, the budget for forest development has grown significantly. From 2001 to 2010, allocations rose from 8,000 million in 2001 to 21,000 million by 2009–10. Between 2010 and 2020, funding increased steadily, except for a slight dip in 2014–15 due to budget restructuring. By 2019–20, the budget had reached 31,00 million, driven by climate change priorities and the utilization of CAMPA funds. Looking ahead to 2020–2025, budget projections estimate a steady rise, reaching around 27,000 million by 2024–25, in line with India's international climate commitments.

Key components influencing these budget allocations include the NAP, CAMPA, the Green India Mission, and international agreements like the Paris Agreement and COP26. Budget allocation for forest protection is spread across various schemes and initiatives, highlighting the government's commitment to this cause. However, the implementation and effectiveness of these initiatives remain a challenge, especially in the face of ongoing deforestation due to developmental activities. Figure 4 illustrates the Union Budget allocation in India from 2001 to 2024 for conservation of forest trees.

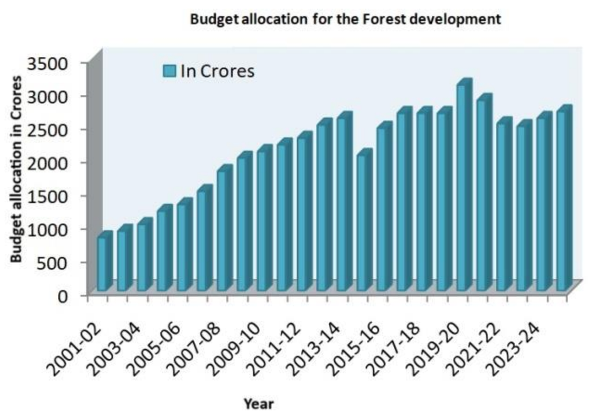


Fig. 4: Union Budget allocation in India from 2001 to 2024 for conservation of forest trees

Technological advancements are also being influenced to monitor and protect forests. The Forest, and Climate Change, conducts biennial assessments of forest cover using remote sensing technology, which is vital for informed decision-making in forest conservation (Chavhan *et al.*, 2024). India and several other countries are interested in using technologies to enhance forest conservation and monitor deforestation effectively (Sethuraman *et al.*, 2022). Key innovations include remote sensing and satellite imagery, which utilize satellites like Sentinel-2 and Landsat to provide high-resolution images that track changes in forest cover. The Forest Survey of India (2023) uses these images for its biennial State of Forest Report, while geo-spatial technology integrates GIS with satellite data to map forest health and identify threats. On the other hand, Drones complemented these efforts by offering real-time, localized monitoring, assessing forest damage, and tracking wildlife, while IoT devices provide continuous data on forest conditions, aiding in the detection of illegal activities and environmental changes. Additional technologies such as forest fire alert systems, AI, and blockchain further boosting for forest management. AI algorithms predict fire-prone areas and analyze large datasets for deforestation patterns, while block-chain ensures transparent supply chains by certifying the legal origins of timber. Cloud-based platforms like Global Forest Watch enable the real-time tracking of deforestation and forest fires, supporting collaborative conservation efforts. Together, these technologies address challenges such as illegal logging, forest fires, and climate change, contributing to the long-term sustainability and resilience of forests.

Contributions

This research addresses the challenge by proposing a robust sound event detection model for forest monitoring.

A comprehensive dataset was prepared, comprising four classes of sounds: chainsaw sounds, handsaw sounds, synthetic axe-cutting sounds, and negative environmental sounds. The dataset was analyzed using six advanced audio feature extraction techniques Mel-Spectrogram, MFCC, Chroma, Spectral Contrast, Tonnetz, and Spectral Bandwidth to capture crucial characteristics for sound classification. Multiple state-of-the-art deep learning models, including a Customized 1D CNN. This work demonstrates the potential of sound-based monitoring systems to detect deforestation activities effectively, providing a scalable solution to support forest conservation initiatives. Key contributions are:

- Prepared a dataset with four distinct sound classes, including synthetic axe-cutting sounds recorded in Mulugu district, Telangana, India
- Applied six audio feature extraction techniques (Mel-Spectrogram, MFCC, Chroma, Spectral Contrast, Tonnetz, and Spectral Bandwidth) to capture key audio characteristics
- Evaluated the dataset using various deep learning models using with six different feature combinations. Apart from that Custom-CNN model achieved the 98% accuracy

Literature Review

Akbal (2020) classified environmental sounds using suitable feature extraction techniques in three stages that are feature generation, selection, and classification. One-dimensional Local Binary Pattern (1DLBP), Ternary Pattern (1D-TP), and statistical methods were used for feature extraction, while Neighborhood Component Analysis (NCA) selects features. They implemented SVM and gained 90.25% accuracy on the ESC-10 dataset. (Permana *et al.*, 2022) developed a bio-inspired early warning system for forest fires using bird sounds. Five hundred eighty bird samples were collected through microphones, sounds were preprocessed with STFT, and black-and-white spectrograms were used for faster classification. A CNN classified the sounds into normal and threatened conditions, securing 96.45% precision with data from Indonesian birds.

Khare *et al.* (2020) proposed a hybrid model using Optimum Allocation Sampling (OAS) to select samples, which are converted into spectrograms via STFT and fed into pre-trained AlexNet and VGG-16 networks. Deep features were extracted and classified using various techniques, including decision trees and support vector machines. Tested on the ESC-10 dataset, the model achieved accuracies between 87.9 and 95.8%, outperforming existing methods and offered a robust solution for automatic environmental sound classification.

Jiang *et al.* (2023) high-resolution UAV imagery was employed to map six tree species, standing dead trees, and canopy gaps within a subtropical montane forest in eastern China. The researchers focused on a specific time when leaf color differences were prominent, utilizing four classification methods—KNN, CART, SVM, and RF to identify the tree species. The results revealed that UAV imagery captured during distinct leaf periods can effectively map tree species in complex mountainous terrains. In this research, KNN achieved the highest accuracy at 83%.

Vinod *et al.* (2023) employed a convolutional neural network for TOF mapping in Bengaluru, India, using HRS images. A semi-automated process was developed for generating labeled training samples via Object-Based Image Analysis (OBIA), reducing data preparation time. A U-Net deep learning model was then used for TOF classification, achieving 89.65% accuracy and a 93.03% F1 score, outperforming OBIA (80.73% accuracy, 86.44% F1 score). This methodology for assessing TOF in urban areas can also be applied to agriculture-dominated regions.

Lohit (2021) focused on using drones for reforestation, developing a working prototype to address various challenges. Drone reforestation is 9 times faster than manual planting, covering larger areas efficiently by staying airborne. The future scope includes integrating deep learning techniques for drones to track deforested land and accurately sow seeds by hovering over specific locations or dropping seeds mid-air. Behera *et al*. (2023) presented LW-AerialSegNet, a lightweight CNN design for segmenting images by adding more layers to capture important features while using techniques to reduce the number of parameters, making it suitable for devices that operate on the Internet of Things (IoT). Tested on the NITRDrone and Urban Drone datasets, it achieved 82% and 71% intersection over union (IoU), outperforming other methods. LW-AerialSegNet can be used on drones to identify objects like plants and road lines, helping with mapping urban and agricultural areas.

Rathod *et al*. (2023) presented a UAV designed for rescue and safety in forests, featuring a durable F450 quadcopter frame, four 1000 KV brushless motors, and a KK2.1 Flight Control Board for 90 minutes of flight. They included a Raspberry Pi camera for real-time video, a GSM module for contactless communication, and a motorized lid for quick aid delivery from a first aid kit. The Neo-6 M GPS module provides accurate positioning (2.5 m accuracy) and collects temperature and humidity data with a DHT 11 sensor. Using deep learning models like ANN and GANs, the UAV predicts forest fires with 90.7% accuracy. Kasyap *et al*. (2022) proposed cost-effective deep learning techniques for predicting forest fires using a mixed approach that combines YOLOv4 tiny and LiDAR methods. Unmanned Aerial Vehicles (UAVs) are used to patrol forest areas. The model deployed on the UAV achieved a classification time of 1.24 seconds, with 91% accuracy and an F1 score of 0.91.

Das *et al*. (2022) presented an edge-enabled drone network integrated with mobile edge computing and machine learning models to predict bird species. Experiments conducted in two geographic regions achieved 98.2 and 96.9% accuracy using a random forest classifier, with log loss values of 0.07 and 0.4. The edge device utilized only 1.4% of CPU and 329.14 MB of buffer memory, with an execution time of 45 milliseconds. Anees *et al*. (2024) investigated the relationship between Landsat-9 remote sensing data and topographical features for monitoring Above-Ground Biomass (AGB). It employed machine learning algorithms, including Random Forest (RF), XGBoost, and Support Vector Regression (SVR), to identify optimal predictor combinations. The RF model, using Landsat-9 OLI and Shuttle Radar Topography Mission Digital Elevation Model (SRTM DEM) predictors, achieved a relative mean absolute error (RMAE) of 14.33%, relative Root Mean Square Error (RRMSE) of 22.23%, and an R^2 of 0.81, making it the most effective model.

Singha *et al*. (2024) combined GIS, remote sensing, and machine learning to find and assess areas at risk of forest fires in the STR and their vulnerability to climate change. They used a dataset of 44 factors, such as topography and climate data, with ten machine learning models, including neural networks and Random Forest, along with optimization methods. The study found high fire risk in the northern and southern parts, with the neural net and RF-PSO models showing risk percentages of 12.44 and 12.89%. Low-risk zones had scores around 23.41 and 18.57%.

Qadeer *et al*. (2024) looked at using machine learning to model Above-Ground Biomass (AGB) in Pakistan's Diamir district using free Sentinel-1 and Sentinel-2 satellite data, along with 171 field-measured points. Several algorithms, including Random Forest and XGBoost, were tested and improved. While Sentinel-2 data performed better than Sentinel-1, combining both gave the best results (R^2 greater than 0.7, RMSE = 105.64 Mg/ha, MAE = 85.34 Mg/ha). da Silva *et al*. (2023) used images from UAVs and machine learning algorithms to identify the invasive species Hovenia dulcis in a conservation area in southern Brazil. Field data were collected through a floristic survey, and UAVs captured RGB images, which were processed to create orthomosaics. The classification involved four categories: H. dulcis, similar species, shade, and other species, using Pixel-Based (PB) and Object-Based Image Analysis (OBIA) with Random Forest and SVM. The RF algorithm in the PB approach performed best, achieving a Kappa index of 0.87 and Overall Accuracy (OA) of 91.5% in training, with 90.91% success in validation.

Ahmad and Singh (2022) proposed an approach that utilized MFCCs and Spectral Centroid for feature extraction for effective classification of environmental sounds. They used Machine learning techniques such as K-means clustering, GMM, and PCA to improve detection accuracy. Where PCA achieved 92% accuracy, and K-means clustering achieved 83% accuracy. Mporas *et al*. (2020) used chainsaw recordings and environmental noises from online repositories, down sampled to 8 kHz with 16-bit resolution. The MFCCs, harmonics-to-noise ratio, voicing probability and dominant frequency are used for extract the feature. On the other hand, the classification model, SVM with an RBF kernel improved accuracy by 2%, reaching 94.42% for an SNR of 20 dB. Qurthobi *et al*. (2025) examined the effectiveness of various pre-trained deep learning models, including MobileNet, GoogleNet, and ResNet, in classifying forest sound recordings from the FSC22 dataset, which comprises 2,025 audio samples across 27 categories. To enhance classification performance, a hybrid approach is proposed by integrating a CNN with a Bidirectional Long Short-Term Memory (BiLSTM) layer. They used MFCC through the Pareto-Mordukhovich method to improve audio feature representation. The proposed BiLSTM layer into GoogleNet, along with data augmentation, reduced the loss to 0.7209 and increased the classification accuracy to 0.7852.

Dataset

In this research, the Dataset was prepared with four distinct classes. Apart from that, three classes were collected from Google Audio, and one class named axe cutting, synthetically prepared in a nearby forest of Mulugu district, Telangana state, India. All sound samples of axe cutting were recorded using a Samsung Galaxy F34 smartphone, which features a built-in high-quality microphone. Recordings were saved in WAV format at a sampling rate of 44.1 kHz to ensure high-fidelity audio suitable for signal processing and analysis. The recordings were made by placing the mobile phone at distances ranging from 5 to 150 meters from the sound source. Recordings were conducted in a controlled outdoor forest environment with environmental condition, temperature ranged between 25 to 40°C. The axe-cutting sounds were recorded in a controlled manner and selected specific trees for recording, ensuring that no tree was felled. Each tree was struck for no more than 2 minutes, with the cut limited to 2-3 inches, and no plant was harmed and activity conducted under the observation of forest security. The chainsaw sound class, Handsaw sounds and negative class sound were collected from Google AudioSet. The audio samples collected for each class were around three hours long and were sampled down to 10-second samples. Initially, 1200 samples were prepared for each class. These samples were augmented with sample down the frequency and increase the frequency for the existing samples, representing the near and far-off effect from microphones when being recorded. The augmentation process increased the number of samples to greater than 2400 number of samples per each class. The chainsaw class provides a comprehensive assortment of recordings featuring various chainsaw types, intensities, and environments. The axe-cutting class encompasses a wide spectrum of axe-cutting techniques, including chopping, splitting, and shaping wood. The handsaw class covers a range of saw types, from fine-toothed models for intricate cuts to coarse-toothed saws for rougher work. The negative sound class presents an array of avian vocalizations as melodic birds, animals and other forest sounds. Figure 5 represents the number of audio files from each class after performing the above-described operations. According to Figure 5, the number of samples in the handsaw class is a bit higher than the rest of the classes because the frequency in this class not have a uniform distribution over the period of 10 seconds and also not reach the threshold set by the algorithm to augment the audio data. The audio waveform representation is the depiction of the audio format in terms of frequency and time domain. Figure 6 representing the patterns of sound waves marked when there is a significant spike which represents the unique trait of that class. For example, the chainsaw spikes are the longest when compared to the rest of the classes, and the axe-cutting spikes are the shortest ones when compared to all others. This research implemented k-fold cross-validation used to evaluate how well the model generalizes to unseen data by splitting the dataset into k parts (folds) here k = 5, training on k-1 parts, and testing on the remaining part. This process is repeated k times**,** and the results are averaged to get a more reliable estimate of the proposed CNN model performance**.** The k-fold cross-validation (k = 5) was performed to assess the proposed Custom CNN model generalization ability, yielding an average accuracy of 99.27% with a low standard deviation of 0.0012.

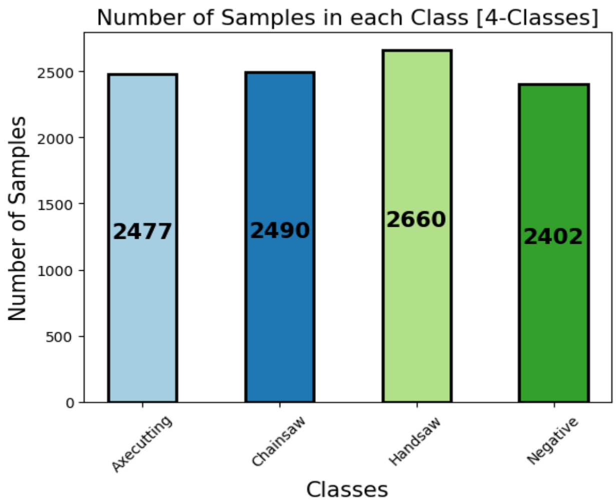


Fig. 5: Number of samples in four classes of dataset

Pre-Processing

The recorded Audio file data O consists of four classes determined as $O = \{O_1, O_2, O_3, O_4\}$. This dataset is first pre-processed, and the audio files are presented in sorted order according to the classes so that the audio files belonging to the same class are together. By using the Librosa library, audio samples are scaled down, and the frequency domain of the samples is analyzed by using various parameters such as sampling rate, channels, and length of the file. The recorded audio files were split into 10-second samples for each class. The split dataset $M = \{M_1, M_2, M_3, M_4\}$.The steps of the procedure explained in Algorithm 1.

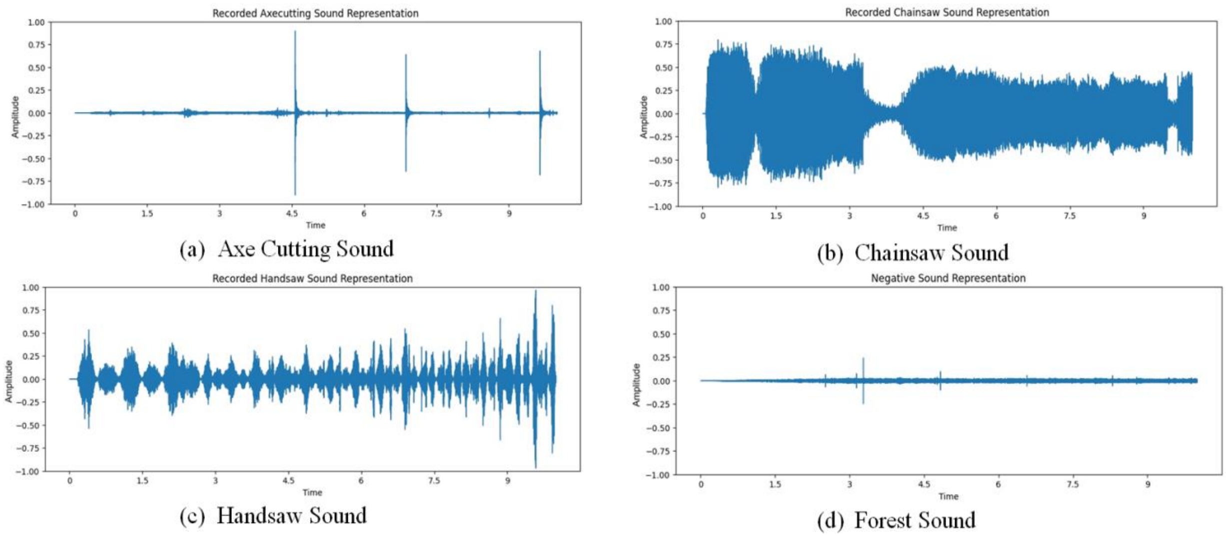


Fig. 6: Sample Sound wave of each dataset class

Algorithm 1: Audio file Pre − Processing

input: $O = \{O_1, O_2, O_3, O_4\}, i = 1, j = 0,$ audio file $f_j, t = 0$

$\boldsymbol{for}\ \boldsymbol{i} = 1\ to\ 4$ **do**

$\boldsymbol{for}\ j = 0\ to\ n$ **do**

Obtain audio file f_j from O_i

resample frequency: $y = 22050$

t= 10 Seconds

$chunks =$ Split$(f_j\ , t)$

for $k, chunk$ in enumerate$(chunks)$ **do**

$chunk\_Name = 'f_j^{'} + \{k\}.wav$

$chunk.export()$

end for

end for

end for

Output: Final dataset O, renamed the O to M

Augmentation

An amplitude-based data augmentation technique was applied to increase the diversity of the dataset. Each original audio file was processed to create two additional versions: one by reducing the volume (dividing the amplitude by two) and another by normalization (scaling based on the maximum amplitude if the scaling factor was less than or equal to 1.1). This method simulates variations in recording conditions such as lower volume and consistent loudness. It helps improve the model's robustness to real-world audio variations. The augmentation process was implemented using Librosa and Soundfile libraries. The output augmented dataset representing, $A = \{A_1, A_2, A_3, A_4\}$. Algorithm 2 illustrated the process of augmentation on the selected dataset.

Algorithm 2: Data Augmentation

Inputs: $M = \{M_1, M_2, M_3, M_4\}$

M: Root Directory

M_i: Sub Directory

F_j: Audio File within Sub Directoty

A: Augmented Directory

$y =$ Audio data

sr: Sampling Rate

$k =$ index for naming augmented files

for $i = 1$ to 4 **do**
if $F_j \in M_i$ **then**
Create directiry A_i/M_i
for each F_j in M_i **do**
$load(y), sr = 22050$
Export y to $A_i/M_i/M_i\_k.wav$
for $k = 1$ to 1200 **do**
//Volume Reduction Augmentation
$y1 = \frac{y}{2}$
Export $y1$ to $A_i/M_i/M_i\_k.wav$
$k = k + 1$
//Normalization Augmentation
Scaling factor $m = \frac{1}{\max(y)}$
if $m \leq 1.1$ **then**
$y2 = y \times m$
Export $y2$ to $A_i/M_i/M_i\_k.wav$
$k = k + 1$

DOI: 10.3844/jcssp.2025.2772.2801

end if
end for
end for
end if
end for
Output: Augmented Audio data A

Methodology

In this research, our main goal is to prevent deforestation. This research focused on finding tree-cutting sounds in the forest area of the Indian environment. As the initial process, we prepared the dataset of four classes. According to the Sound Event Detection to classify the sounds using ML and NN algorithms, the feature extraction process is more important. In this model feature extraction with multiple feature clusters, the compiled dataset is forwarded to a feature extraction algorithm, which extracts multiple features simultaneously. These extracted features are provided in numerical data format. The extracted feature data is inputted into multiple deep-learning models to test the accuracy of each class in real time. Figure 7 shows the flow of the classification system required to analyze the compiled dataset. The accuracy of each class is tested using all classification metrics, and the best feature cluster will be decided based on these metrics.

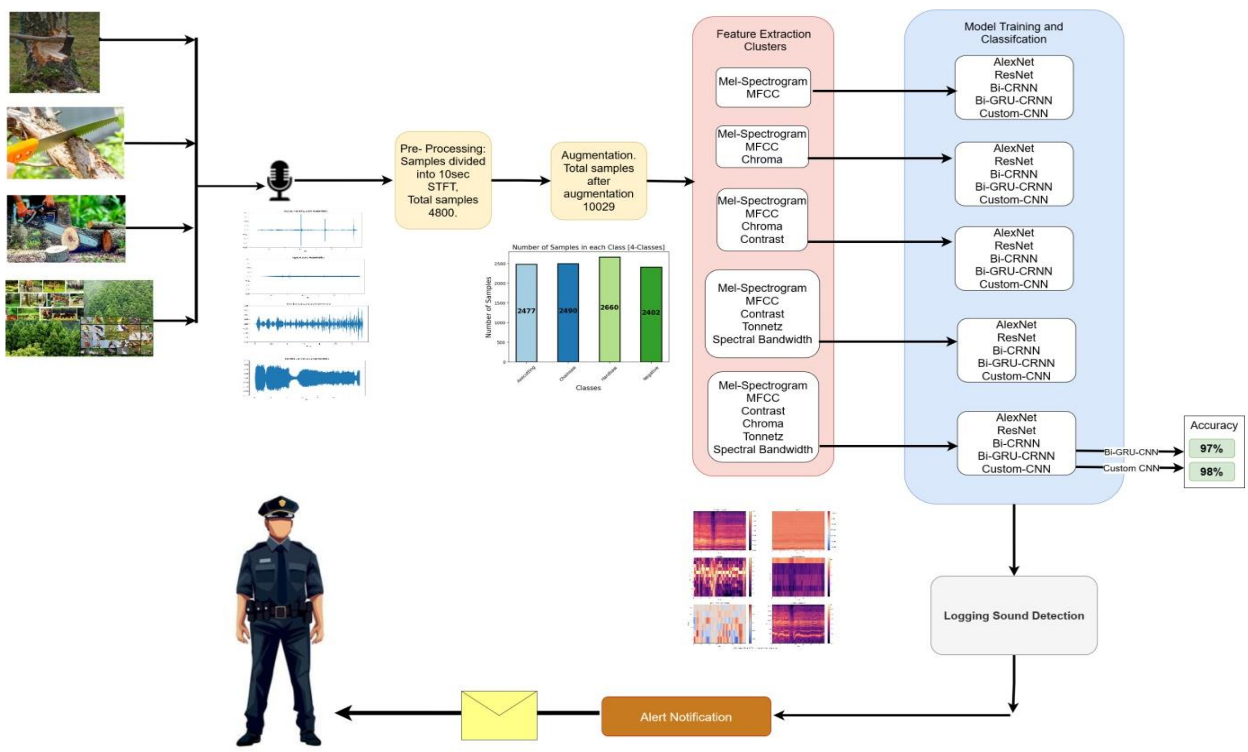


Fig. 7: Flow chart of the proposed model

The proposed feature cluster was selected based on the popularity of the feature extraction techniques used for audio classification and the most suitable for our implementation. The prepared 6 feature extraction techniques are Mel-Spectrogram, MFCCs, Chroma, Spectral Contrast, Tonnetz and Spectral Bandwidth. The Mel spectrogram is used for visual representation of the spectrum of frequencies in an audio signal obtained by converting the linear frequency scale to a logarithmic scale using the Mel scale, which closely resembles human auditory perception. MFCCs are a widely used feature extraction method in audio signal processing. They capture the short-term power spectrum of a sound signal. This process involved computing the Mel spectrogram, taking the logarithm of the powers, applying a discrete cosine transform, and retaining a subset of the resulting coefficients. Chroma representation of the energy distribution of pitch classes in an audio signal is particularly useful for harmonic content tasks, like music genre classification, chord recognition, and melody analysis. Spectral contrast measures the difference in amplitude between peaks and valleys in the spectral envelope of an audio signal. It helps capture the perceptual attribute of timbre, which is crucial in tasks like instrument recognition and genre classification. Tonnetz features

DOI: 10.3844/jcssp.2025.2772.2801

a representation of the harmonic content of an audio signal. Tonnetz is derived from the tonal centroids and is related to the harmonic relationship between different frequencies. Spectral bandwidth measures the width of the frequency range over which most of the signal's energy is concentrated. It provides information about the spread of frequencies in the spectrum, indicating whether a sound is narrow or broad in its frequency content. Figure 8 represents the clustering of features used to convert the audio data to numerical data. Figure 9 illustrates the flowchart of feature cluster selection followed by classification output. Figure 10 shows the Various Feature Extraction Spectrograms of Axe Cutting Sound sample. Figure 11 presents the Various Feature Extraction Spectrograms of Chainsaw Sound sample. Figure 12 presents the various Feature Extraction Spectrograms of Handsaw Sound sample. Followed by Figure 13 which represents the various Feature Extraction Spectrograms of Forest other Sound sample.

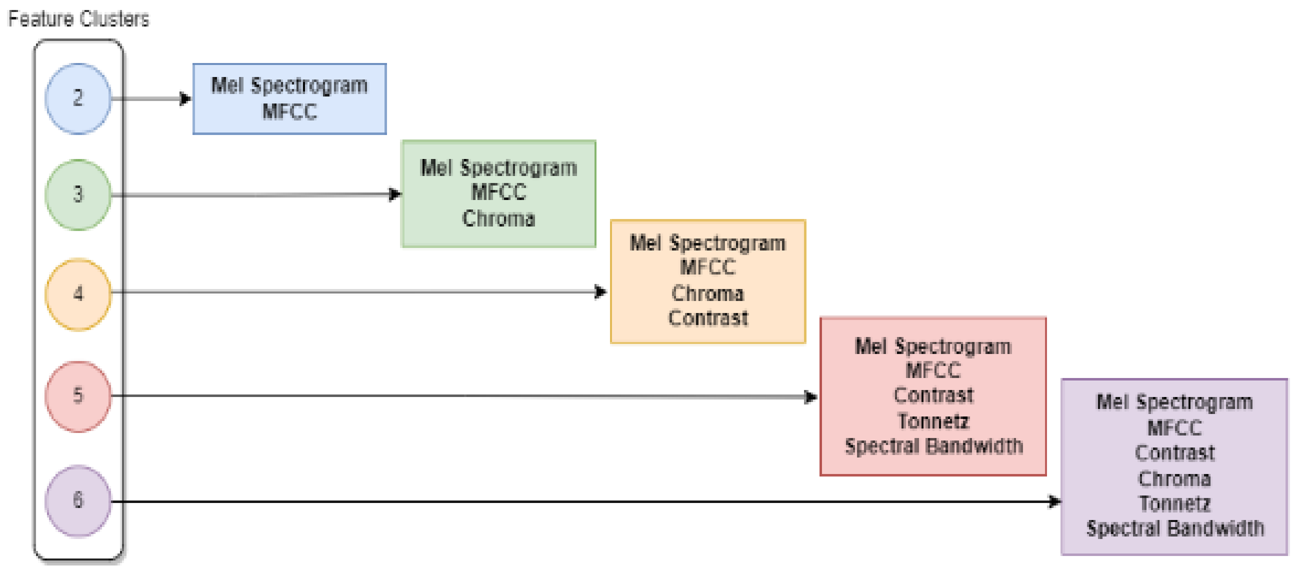


Fig. 8: Feature extraction clusters of the proposed model

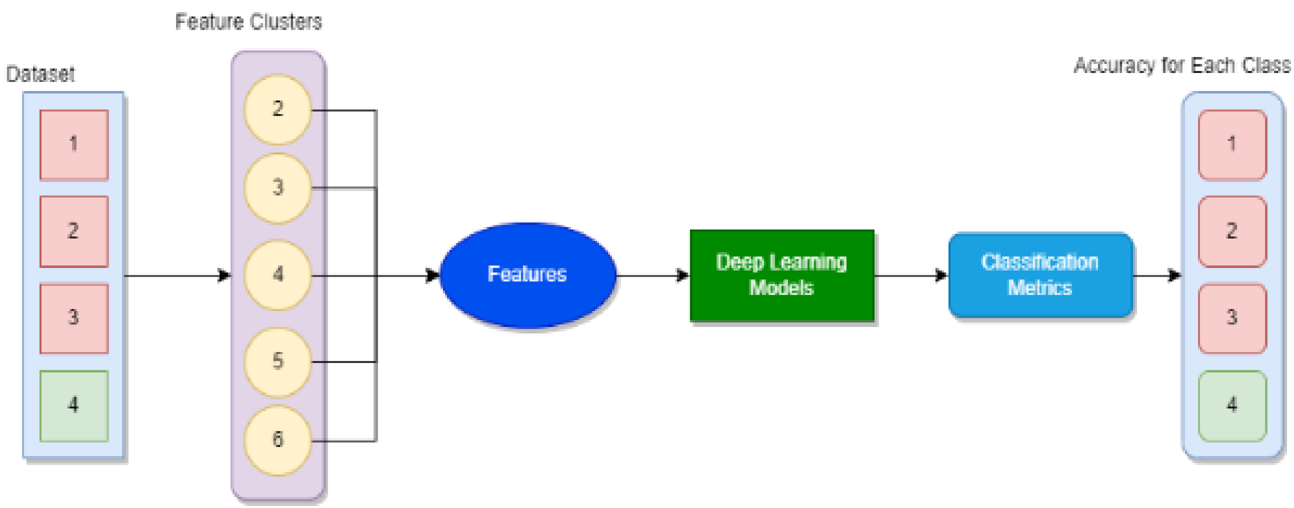


Fig. 9: Flow chart of sound classification based on the features clusters

DOI: 10.3844/jcssp.2025.2772.2801

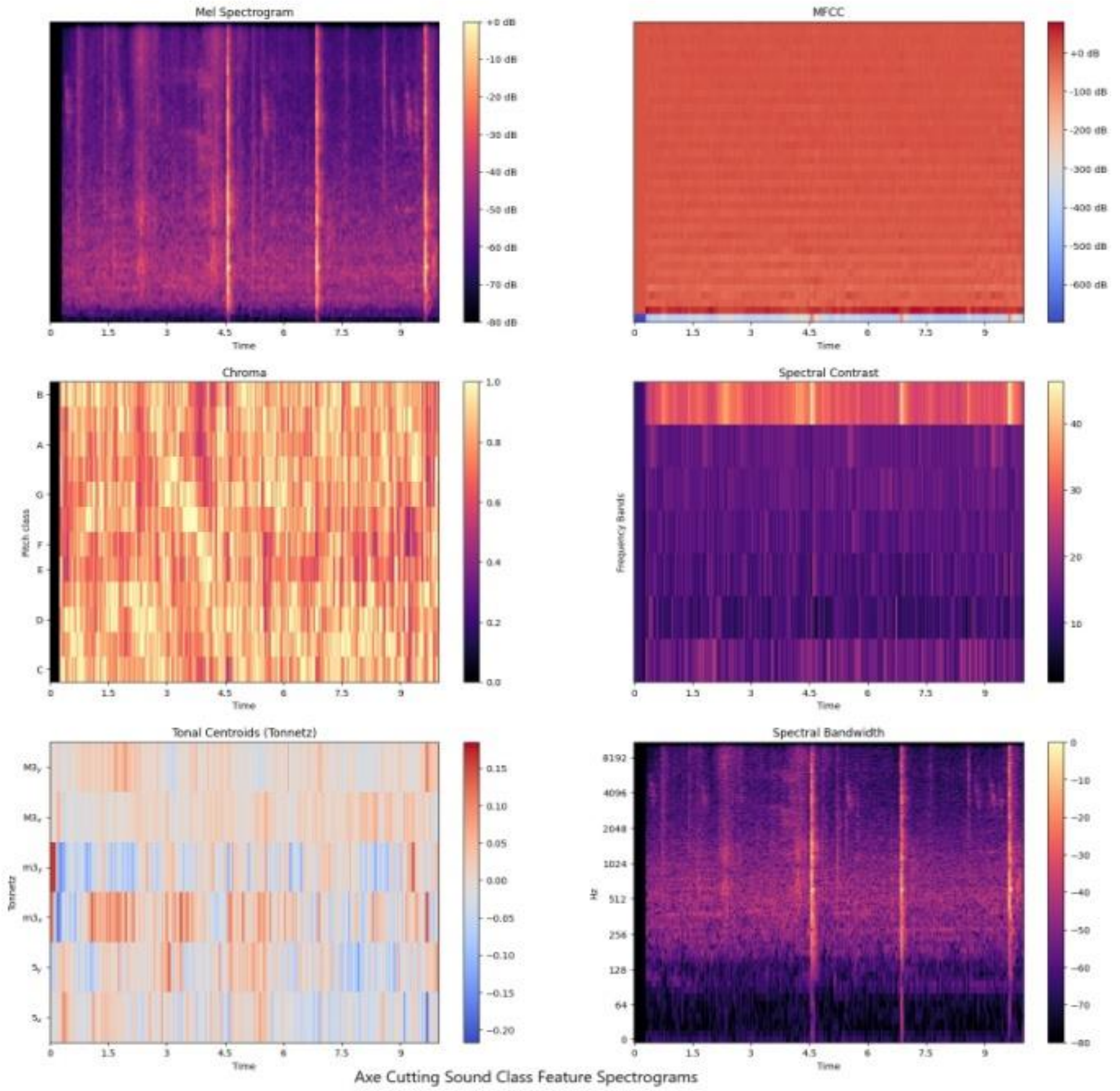
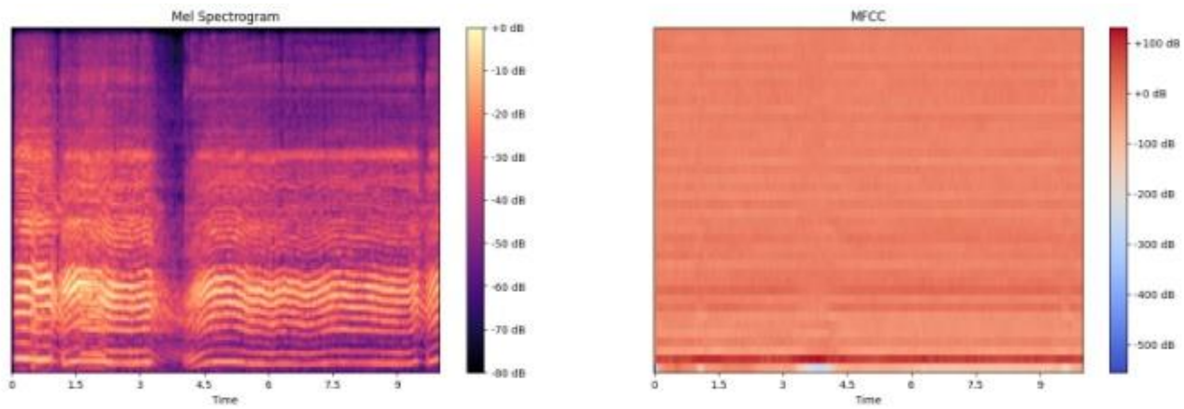


Fig. 10: Various Feature Extraction Spectrograms of Axe Cutting Sound sample



DOI: 10.3844/jcssp.2025.2772.2801

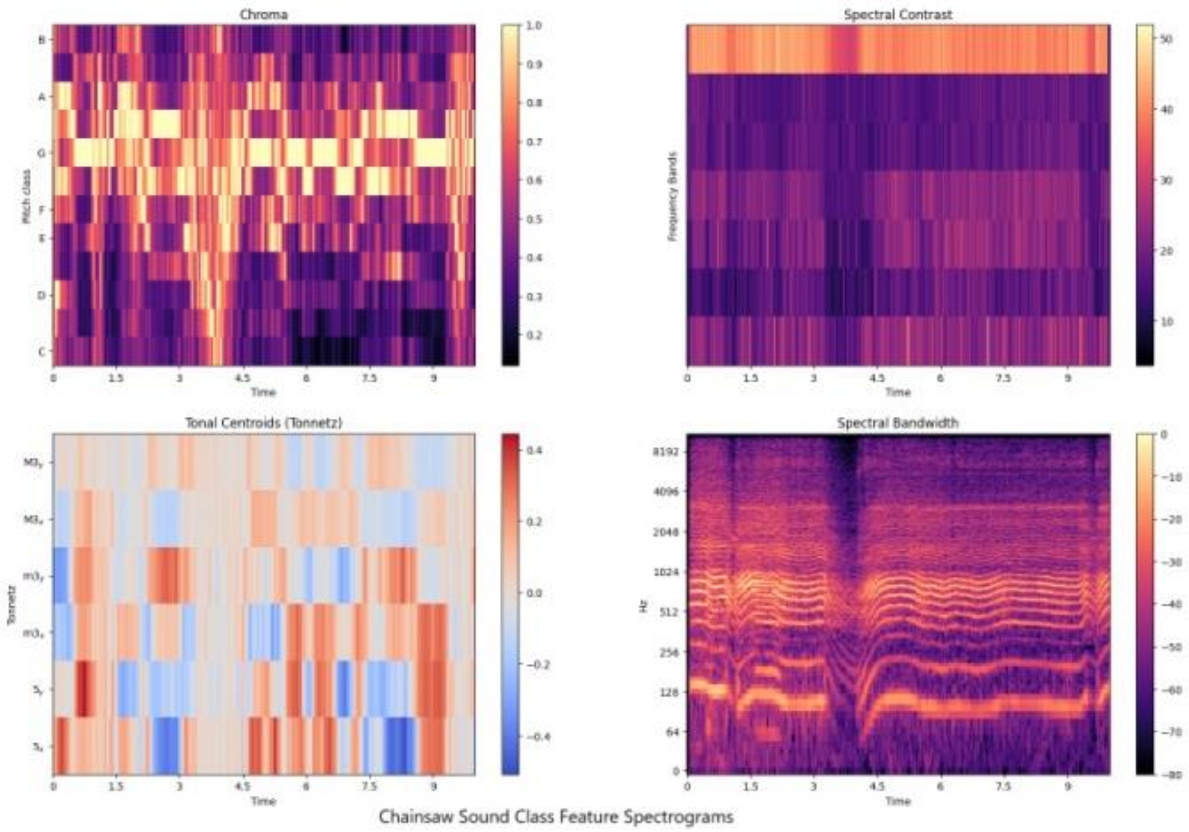
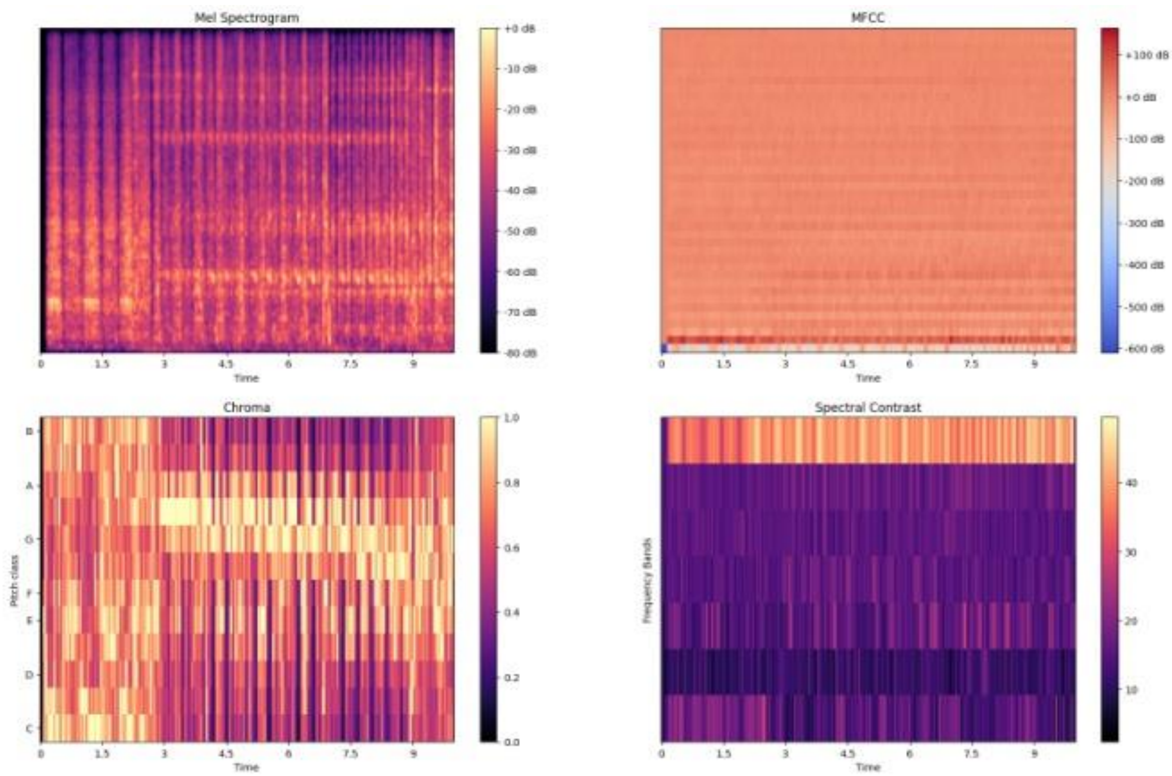


Fig. 11: Various Feature Extraction Spectrograms of Chainsaw Sound sample



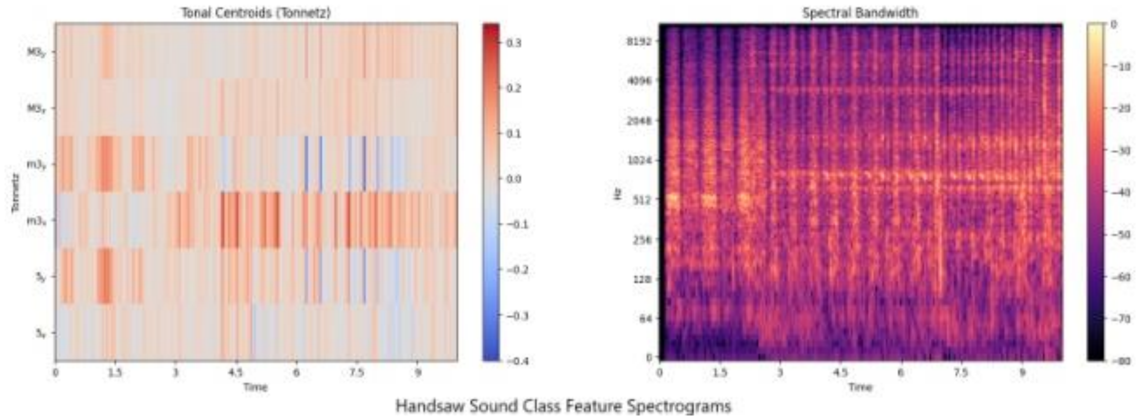


Fig. 12: Various Feature Extraction Spectrograms of Handsaw Sound sample

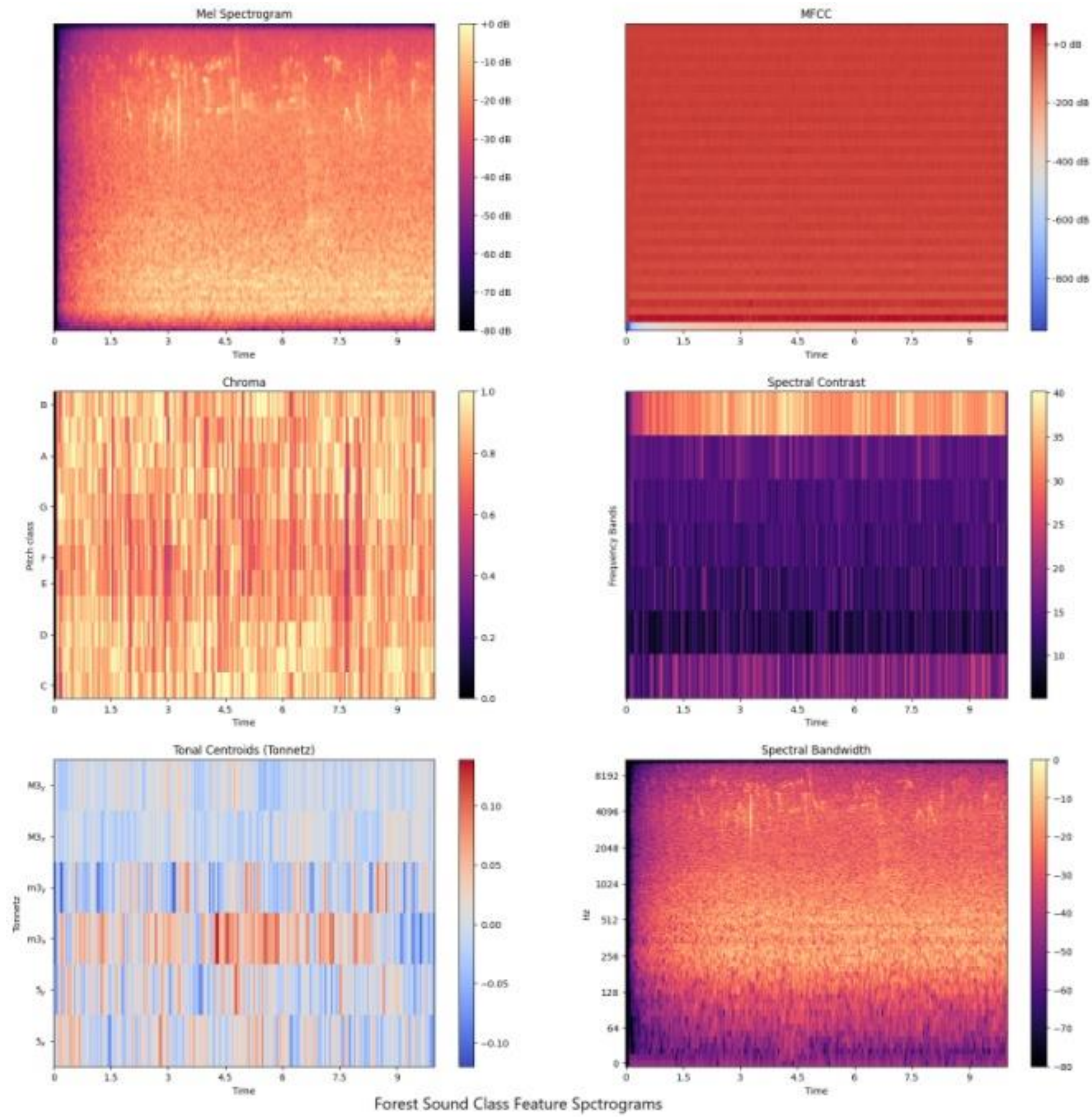


Fig. 13: Various Feature Extraction Spectrograms of Forest other Sound sample

DOI: 10.3844/jcssp.2025.2772.2801

The augmented dataset is used to train the model and evaluate the accuracy of the classes using classification metrics. To train the model, the data needs to be in numerical format, which can be achieved using feature extraction. Algorithm 3 provided the procedure of feature extraction from augmented data samples.

Algorithm 3: Feature Extraction

Input : $A = \{A_1, A_2, A_3, A_4\}$
repeat
repeat
repeat
Obtain audio file from A_i
Extract $\pm I_j$ Features
Calculate Mean of $\pm I_j$ Features
Store the mean Value of $\pm I_j$ Features
Until: $\pm I_j$ ($> 2\&\& \leq 6$)
Until: All Files in A_i
Until: All Classes $A = \{A_1, A_2, A_3, A_4\}$
Output: $\sim A = \pm I$ Feature Clusters
(Where $\pm I \geq 2\&\& \leq 6$)

The augmented dataset is used to train the model. Various sound features are extracted from the augmented data. The algorithm demonstrates the steps in feature extraction, and the process of these extracted features is further split into training and testing data for training the model. In the above algorithm, the augmented A data is converted to numerical $\sim A$ data by performing the feature extraction operations. The $\pm Ij$ Features means the number of features in a feature cluster. The I represents the feature cluster number, and j represents the number of feature cluster numbers. For instance, in the two Feature cluster, there are two features to be extracted I represent the number '2' whereas the j *represents* the remaining number of features that are needed to be extracted for this feature cluster, like '1' and '2'. The ($> 2\&\& \leq 6$) represents the number of feature clusters from '2' to '6'. The output of the above algorithm is the feature clusters, which are extracted from the audio files represented by $\sim A$.

Implementation

In our research, we divided the dataset for training and testing with ratio of 70:30. The deep learning models used for the classification are Customized Convolutional Neural Network (CNN), Bi-directional Convolutional Recurrent Neural Network (BiCRNN), Bidirectional Gated Recurrent Unit Convolutional Recurrent Neural Network (Bi-GRU-CRNN), AlexNet and ResNet.

The Customized 1D Convolutional Neural Network defined using Keras' Sequential API. The model begins with a Conv1D layer that accepts input data with 194 features (input shape of (194, 1)) and applies 64 filters with a kernel size of 3, using ReLU activation to introduce non-linearity. A second Conv1D layer follows, also with 64 filters and the same kernel size, again using ReLU activation to extract further patterns from the data. After these two convolutional layers, a MaxPooling1D layer with a pool size of 3 is introduced, reducing the spatial dimensions of the feature maps to simplify the representation while retaining important information. The output of the pooling layer is then flattened, converting the 3D tensor into a 1D vector so it can be passed through fully connected (dense) layers. The model included a sequence of four dense layers with 100, 50, 25, and 10 neurons, progressively reducing the dimensionality. Finally, the output layer has 4 neurons, corresponding to 4 output classes, and used SoftMax activation for multi-class classification. In our proposed model, we implemented the RMSprop optimizer to enhance training stability and accelerate convergence. Since our research involved with multi-class classification using the categorical cross-entropy loss function, which is crucial to optimize weight updates effectively. RMSprop plays a key role in this process by adaptively adjusting learning rates for each parameter, preventing drastic weight changes, and ensuring stable training. RMSprop mitigates large fluctuations in weight updates, reducing oscillations and improving overall model performance. RMSprop offers better handling of vanishing gradients and ensures faster convergence, making the model to suitable for training on noisy datasets. The batch size of the model training is 5 to process the five samples at a time before update the weights. The learning rate to controls the weights during the training is 0.001 for ensuring stable training and high accuracy. The model run for 10 epochs for training and testing. Figure 14 shows the Architecture of Proposed Custom-CNN model and Algorithm 4 illustrates procedure of model training on featured data.

The extracted features in both the training and testing data are provided to a model to be trained. Augmented feature extracted data are considered $\sim A'$ for the training dataset and $\sim A''$ for the validation dataset. Then, the convolutional matrix will be considered W' and W'' as the weights initially. Then, the updated weights of the model will be considered GW' or GW'' for training and validation data, respectively. The algorithm for training the model using convolutional layers is given below. The weight matrices for the training and validation defined as below equations:

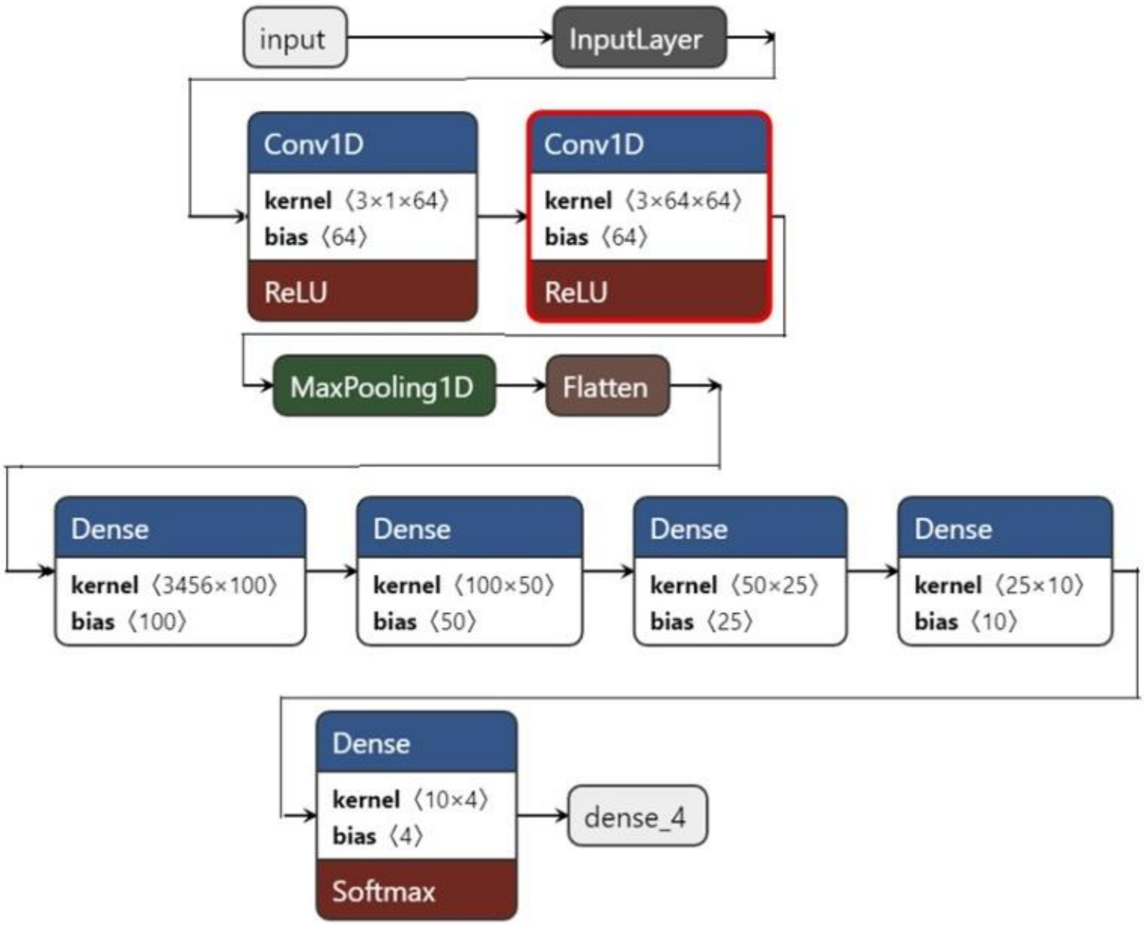


Fig. 14: Architecture of Proposed Custom-CNN model

Algorithm 4: Model Training

Input: $\sim A' = \{\sim A'_1, \sim A'_2, \sim A'_3, \sim A'_4\}$ or $\sim A''$
$= \{\sim A''_1, \sim A''_2, \sim A''_3, \sim A''_4\}$
A: Input Data, Y_{true}: True labels
$r = 10029$ //number of rows from the extracted feature
for each i in r **do**
Initialize the convolutional kernal matrices
W_1 with random values
$H_1 = f_{ReLU}(\text{Conv1D}(A, W_1, k_1))$
$H_2 = f_{ReLU}(\text{Conv1D}(H_1, W_2, k_1))$
$P = \text{MaxPooling}(H_2, p)$
$A_{flattened} = \text{Flatten}(A)$
Compute loss: $loss = \mathscr{L}(A_{flattened}, Y_{true})$
Initializing the weight matrices randomly for
the dense layers
Set up initial weights $W_i$and biases b_i for dense layers .
B=$W_i, b_i \leftarrow$ RandomInitialization()
$D_1 \leftarrow \sigma(X_{flattened}, W_1, b_1)$
for $i = 1$ downto 4 **do**
$D_{i+1} \leftarrow \sigma(D_i, B)$
end for
$\mathcal{L} = loss\ (D_i, Y_{true})$
for $i = 4$ downto 1 **do**
//Backward pass to calculate gradients:
$\Delta W_i \leftarrow \text{Backprop}(D_i, Y_{true})$
$\Delta b_i \leftarrow \text{Backprop}(D_i, Y_{true})$
$W_i \leftarrow \text{Update\_Weight}(W_i - \eta \Delta W_i,)$
$b_i \leftarrow \text{Update\_Biase}(b_i - \eta \Delta b_i)$
$B \leftarrow B - \eta \dfrac{\partial \mathcal{L}}{\partial B}$

DOI: 10.3844/jcssp.2025.2772.2801

// $\frac{\partial \mathcal{L}}{\partial B}$represents the collection of all gradients for weights and biases.

end for

until: $loss_t \approx loss_{t-1}$and $t \geq max\_iterations$=10

Ensure: Trained Model R and Updated weight B

Repeat:

Obtain a single row from the extracted feature of $\sim A^{'}$ or $\sim A^{''}$

Apply convolutional operation $(f^{'}, k^{'}), (f^{''}, k^{''})$

Apply Max pooling operation(p) Update the $W^{'}$and $W^{''}$mat

Until: All rows in $\sim A^{'}$ or $\sim A^{''}$

$$GW^{'}[m,n] = (\sim A^{'} \times k^{'})[m,n] = \sum_j \sum_k k^{'}[j,k] \sim A^{'}[m-j, n-k]$$

$$GW''[m,n] = (\sim A^{'} \times k^{''})[m,n] = \sum_j \sum_k k^{''}[j,k] \sim A^{''}[m-j, n-k]$$

In the above equations, GW' determines the weight matrices of the training data whereas GW'' determines the weight matrices of validation data that is provided to the dense network for classification. The other parameters such as $[\mathrm{m},\mathrm{n}]$ determine the numerical range in a feature cluster to which the convolutional and pooling operations are applied to get the feature map matrices where, $\sim\mathrm{A}'$, $\sim\mathrm{A}''$ are the feature rows and the k', k'' are the initial kernel matrices. After obtaining the weight matrices by applying the convolutional and pooling operations on each row and column the matrices are fed to the dense layer which performs the classification of cancer for lungs. In the next step, the extracted features are fed to the dense neural network which comprises of neurons which help in classifying the audio data according to the classes specified. Let us consider the initial input to the dense layers as the transformed version $\sim\mathrm{GW}'$ flattened values from the matrix which are considered as the input neurons of the dense layer, the dense layer consists of weights which are randomly initialized and updated accordingly by forward and backward propagation in numerous iterations, let us consider this randomly initialized weights as B'. These weights are forwarded to next dense layer and updated. Finally, the weights determine the probability of each class from where the maximum probability of class is chosen to classify that image. Algorithm 5 illustrates the procedure of Classification with Dense Layers.

Algorithm 5: Classification with dense layers

Input: Pre − trained weights GW' or GW'',Input Matrix : X, Y_{true}

$X_{flattened}$ = Flatten(X)

/ weights $W_i$and biases b_i for dense layers

B={W_i, b_i} ← RandomInitialization()

repeat

for $i = 1$ *to* 4 **do**

H_i ← Dense$_\mathrm{i}$(X, W_i, b_i)

end for

for $i = 4$ downto 1 **do**

//Backward pass to calculate gradients:

$\Delta W_i, \Delta b_i$ ← Backprop(H_i, Y_{true})

W_i ← Update_Weight($W_i - \eta\Delta W_i,$)

b_i ← Update_Biase($b_i - \eta\Delta b_i$)

B' ← Update(B)

end for

until: $loss_t \approx loss_{t-1}$ and $t \geq max\_iterations$=10

Output: Trained Model R and Updated weight B′

Algorithm 5 works for training data which gets transformed into input layer for the dense layers and the training process occurs accordingly, the validation set of values is used to validate the training of the model by classifying the values in that iteration. The output of the above algorithm will give the completely trained model R and the final updated weights B'. These weights determine the probability of each class when an input Audio is given which is forwarded to the model after feature extraction.

Results and Analysis

The results of the trained models are being discussed in this section for each feature cluster group. And this section also provides detailed explanation of how models are performing for 10 epochs in each of the feature cluster.

Two Features Cluster

In 2-Features cluster, the models while training have a significant rise in the accuracy from 2nd iteration. The model providing the best accuracy is the Bi-GRU-CRNN Custom model, the ResNet pre-defined model provided less accurate classification when compared to all the other models. Similarly, the ResNet whose accuracy and loss keeps fluctuating over the period of 10 iterations. The ROC curve and Precision vs Recall curve depicts the area created by all

DOI: 10.3844/jcssp.2025.2772.2801

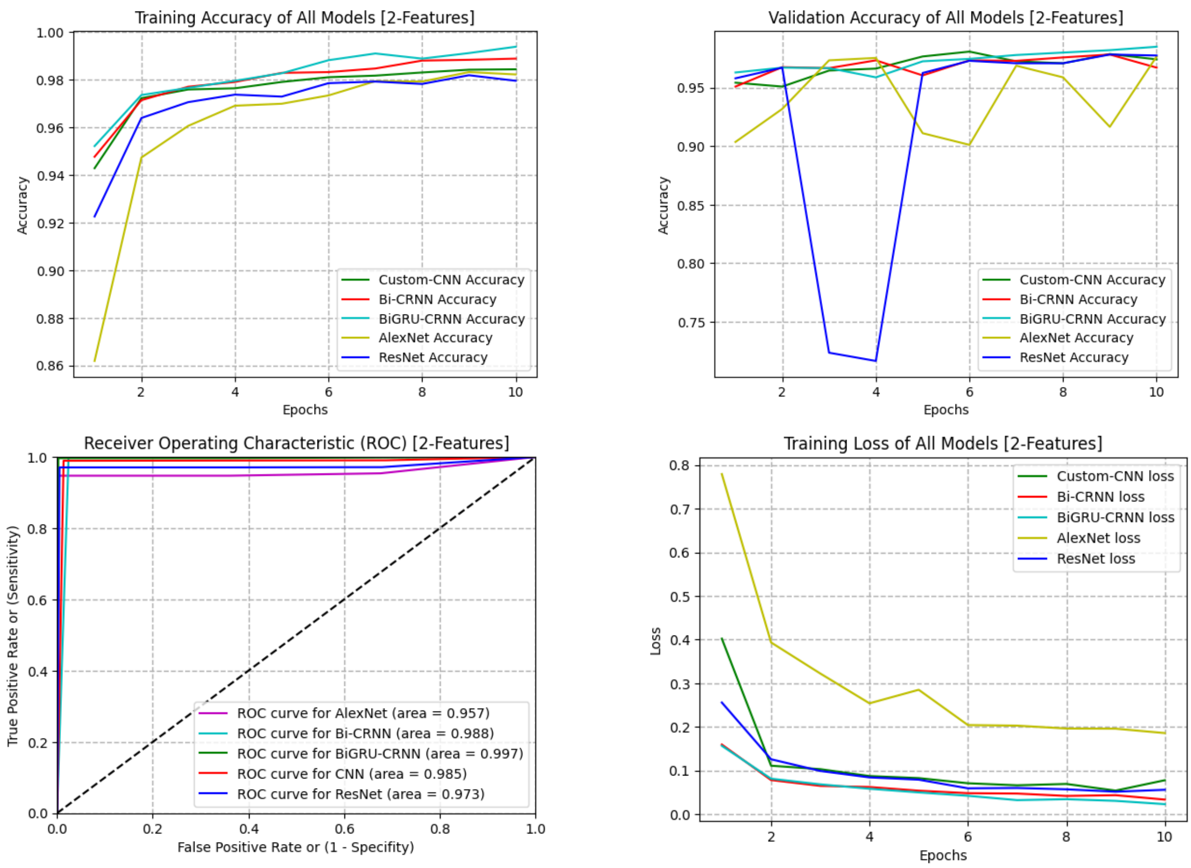
the models and shows the minute difference between them all. The 2-Feature cluster provided a valuable insight on how models work and train if these features are chosen for classifying the audio data. Table 2 presents the performance metric results for two features cluster. Figure 15 illustrates the Two Feature Cluster result analysis.

Three features Cluster

In 3-Features cluster, the models while training have a significant rise in the accuracy from 2nd iteration. The model providing the best accuracy for Custom model, the ResNet pre-defined model provided less accurate classification when compared to all the other models. Similarly, the validation accuracy and loss curves depict that the most unstable for ResNet and AlexNet whose accuracy and loss keeps fluctuating over the period of 10 iterations. The ROC curve and Precision vs Recall curve depicted the area created by all the models and shows the major difference between them all, the ResNet curve's area collapses in-terms of specificity and precision for classifying the data. Table 3 presents the performance metric results for three features cluster. Figure 16 presents the Three Features Cluster result analysis.

Table 2: Performance Metric results for two features cluster

Model	Precision	Recall	F1-Score	Accuracy	Loss
Training					
AlexNet	0.91	0.91	0.87	0.90	0.27
Bi-CRNN	0.93	0.92	0.93	0.93	0.12
BiGRU-CRNN	0.95	0.93	0.94	0.93	0.10
Custom CNN	0.95	0.94	0.94	0.95	0.09
ResNet	0.71	0.73	0.69	0.70	0.13
Validation					
AlexNet	0.89	0.90	0.90	0.90	0.75
Bi-CRNN	0.93	0.92	0.94	0.94	0.10
BiGRU-CRNN	0.92	0.91	0.96	0.96	0.09
Custom CNN	0.94	0.93	0.95	0.97	0.04
ResNet	0.72	0.72	0.69	0.70	0.16



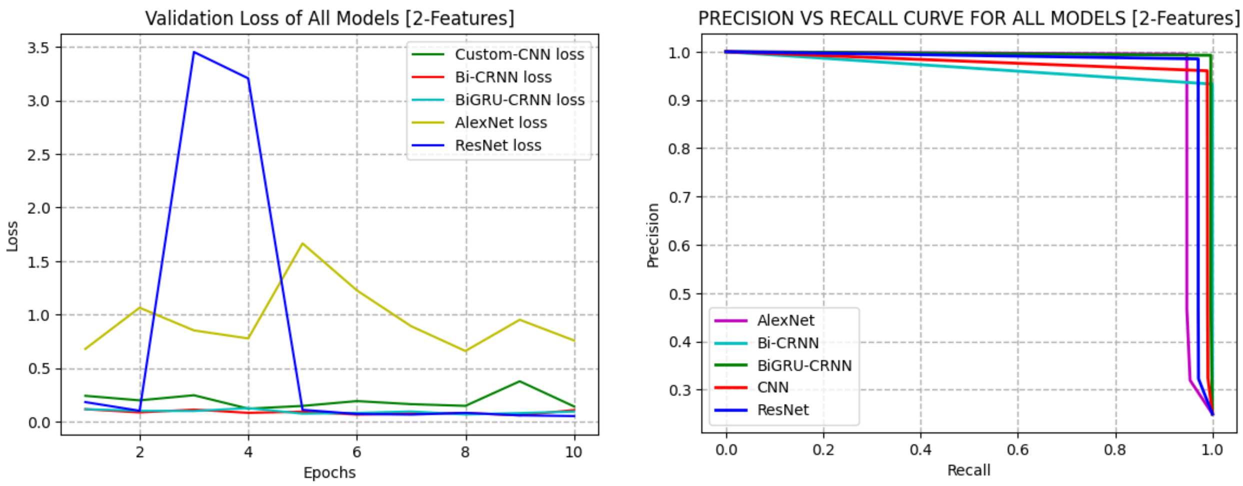
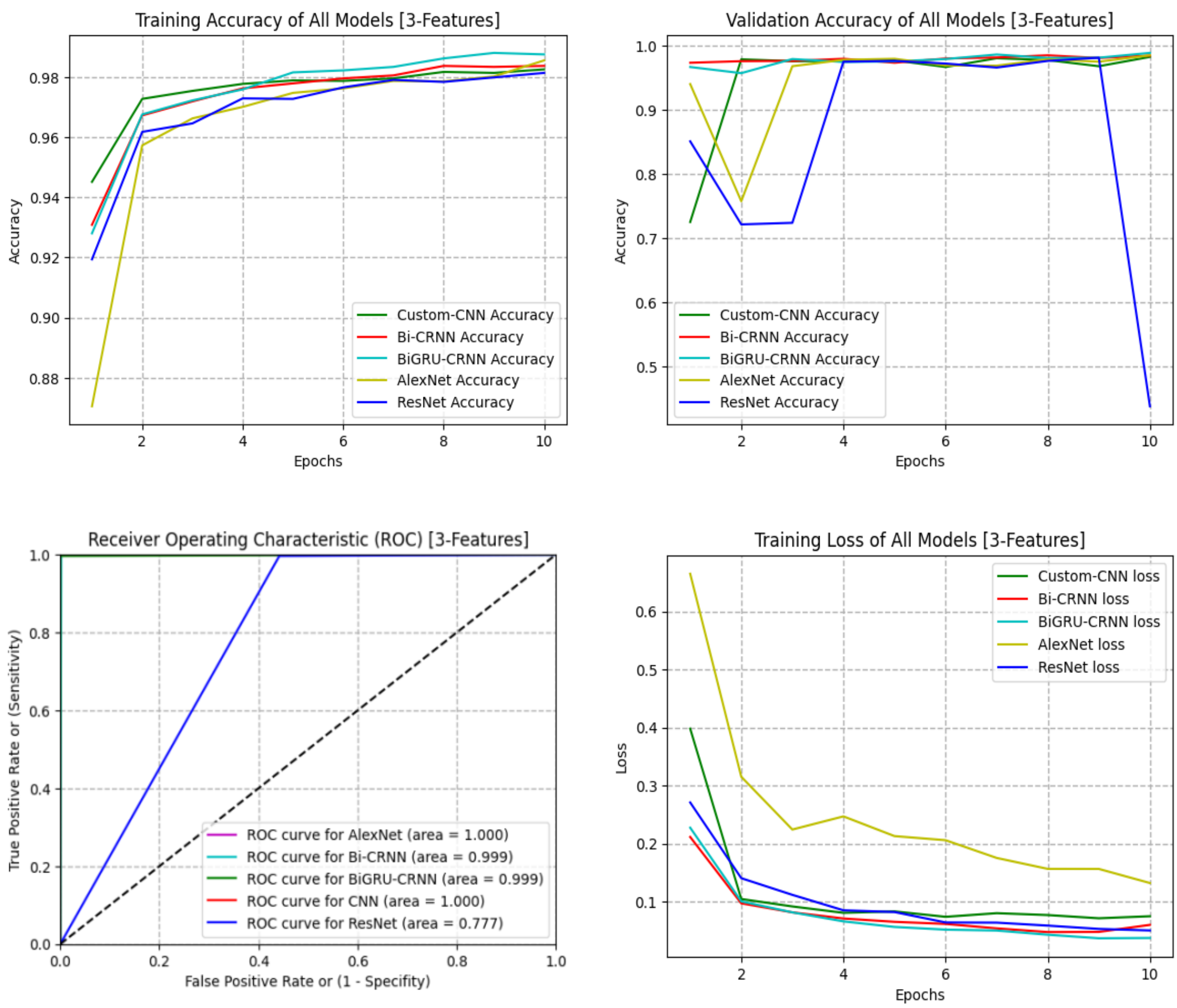


Fig. 15: Two Features Cluster result analysis



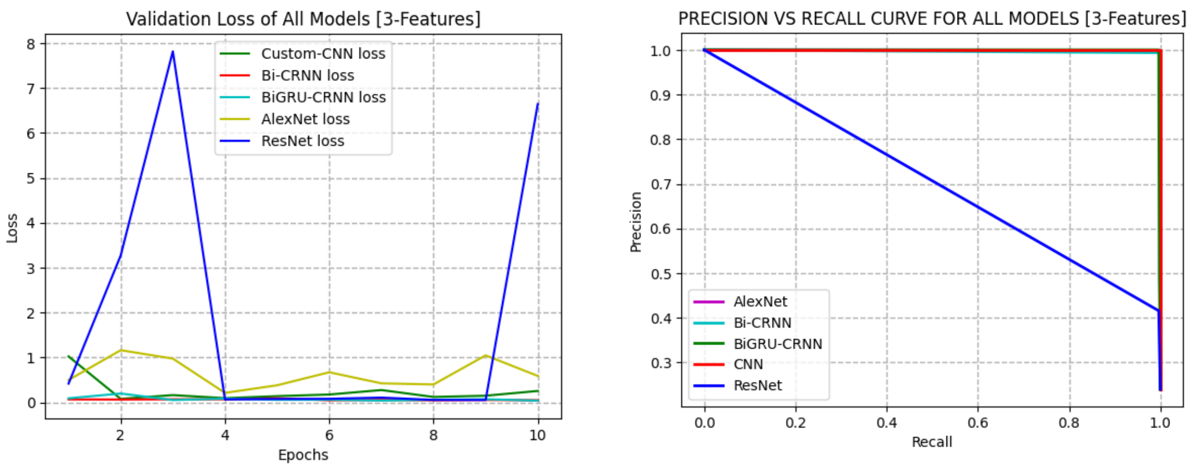


Fig. 16: Three Features Cluster result analysis

Table 3: Model Performance Metrics for Three Features Cluster

Model	Precision	Recall	F1-Score	Accuracy	Loss
Training					
AlexNet	0.89	0.89	0.91	0.90	0.12
Bi-CRNN	0.92	0.93	0.93	0.93	0.09
BiGRU-CRNN	0.93	0.95	0.95	0.95	0.07
Custom CNN	0.94	0.96	0.96	0.96	0.04
ResNet	0.74	0.76	0.68	0.75	1.39
Validation					
AlexNet	0.90	0.89	0.89	0.90	0.58
Bi-CRNN	0.91	0.92	0.93	0.92	0.19
BiGRU-CRNN	0.94	0.95	0.95	0.95	0.09
Custom CNN	0.95	0.95	0.96	0.96	0.15
ResNet	0.78	0.75	0.74	0.73	1.64

Table 4: Model Performance Metrics for Four Features Cluster

Model	Precision	Recall	F1-Score	Accuracy	Loss
Training					
AlexNet	0.91	0.91	0.91	0.89	0.12
Bi-CRNN	0.92	0.92	0.93	0.93	0.15
BiGRU-CRNN	0.96	0.95	0.95	0.96	0.09
Custom CNN	0.95	0.97	0.96	0.96	0.05
ResNet	0.81	0.82	0.80	0.82	3.09
Validation					
AlexNet	0.98	0.94	0.95	0.91	0.41
Bi-CRNN	0.92	0.90	0.95	0.95	0.17
BiGRU-CRNN	0.96	0.96	0.97	0.97	0.14
Custom CNN	0.93	0.99	0.96	0.97	0.11
ResNet	0.87	0.82	0.80	0.82	2.92

Four Features Cluster

In 4-Features cluster, the models while training have a significant rise in the accuracy from 3rd iteration. The model providing the best accuracy for Custom-CNN model and Bi-GRU-CRNN, the ResNet predefined model provides less accurate classification when compared to all the other models. Similarly, the validation accuracy and loss curves depict that the most unstable model is ResNet and AlexNet whose accuracy and loss keeps fluctuating over the period of 10 iterations. The ROC curve and Precision vs Recall curve depicts the area created by all the models and shows a great difference between them all, the ResNet curve's area collapses in-terms of specificity and precision for classifying the data and the area between the curves of all other models is quite observable which suggests that the specificity of this feature cluster is not stable. Table 4 shows the performance metric results for four features cluster. Figure 17 illustrates the four Feature Cluster result analysis.

DOI: 10.3844/jcssp.2025.2772.2801

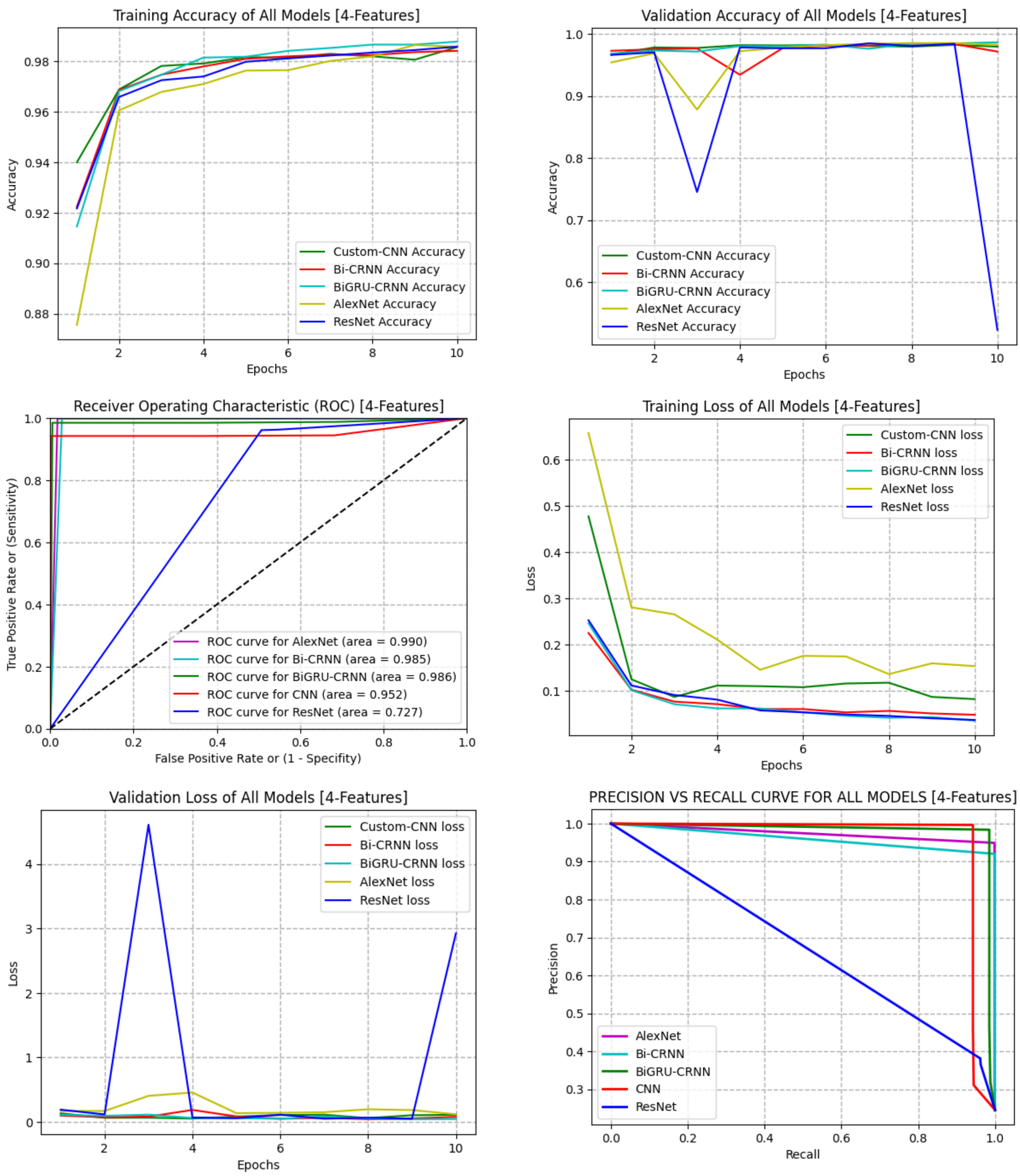


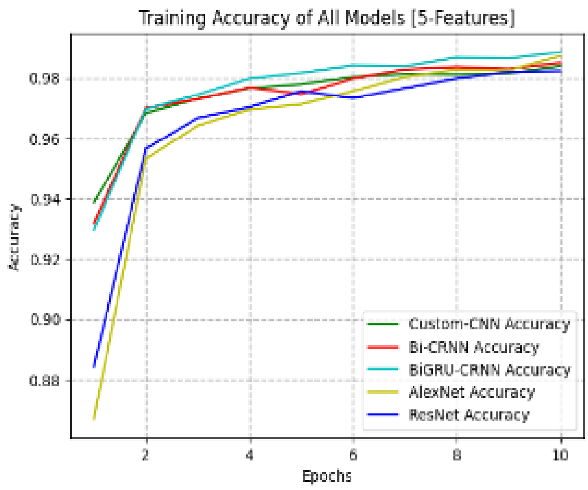
Fig. 17: Four Features Cluster result analysis

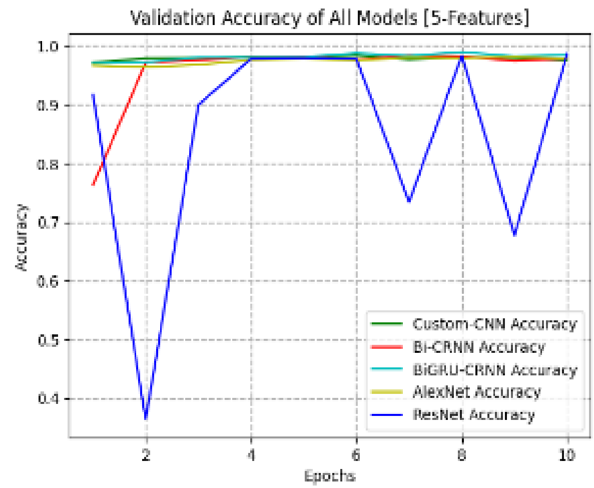
Five Features Cluster

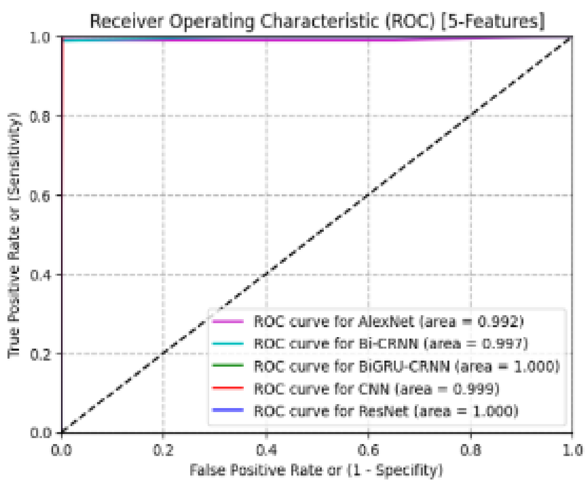
In 5-Features cluster, the models while training have a significant rise in the accuracy from 2nd iteration. The model providing the best accuracy for Custom model and Bi-GRU-CRNN, the ResNet pre-defined model provided less accurate classification when compared to all the other models. The ResNet and AlexNet whose accuracy and loss keeps fluctuating over the period of 10 iterations. The ROC curve and Precision vs Recall curve depicts the area created by all the models and shows a really minor difference between them all, there is hardly any instability in terms of precision and recall in these models. Table 5 shows the performance metric results for five features cluster. Figure 18 illustrates the five Feature Cluster result analysis.

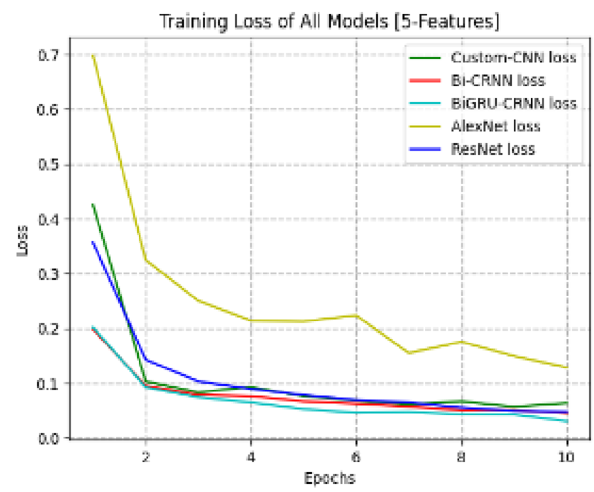
Table 5: Model Performance Metrics for Five Features Cluster

Model	precision	Recall	F1-Score	Accuracy	Loss
Training					
AlexNet	0.90	0.91	0.87	0.92	0.19
Bi-CRNN	0.95	0.95	0.94	0.96	0.08
BiGRU-CRNN	0.96	0.95	0.94	0.97	0.06
Custom CNN	0.95	0.97	0.97	0.98	0.05
ResNet	0.88	0.86	0.87	0.92	0.63
Validation					
AlexNet	0.92	0.93	0.92	0.94	0.43
Bi-CRNN	0.98	0.93	0.94	0.95	0.06
BiGRU-CRNN	0.95	0.98	0.97	0.97	0.04
Custom CNN	0.93	0.98	0.98	0.97	0.02
ResNet	0.89	0.85	0.87	0.91	1.03









DOI: 10.3844/jcssp.2025.2772.2801

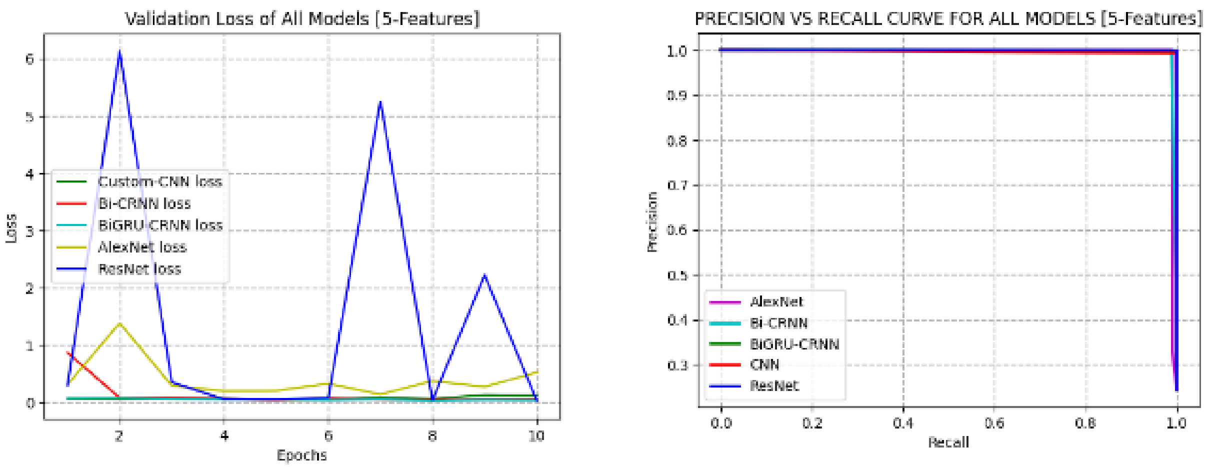


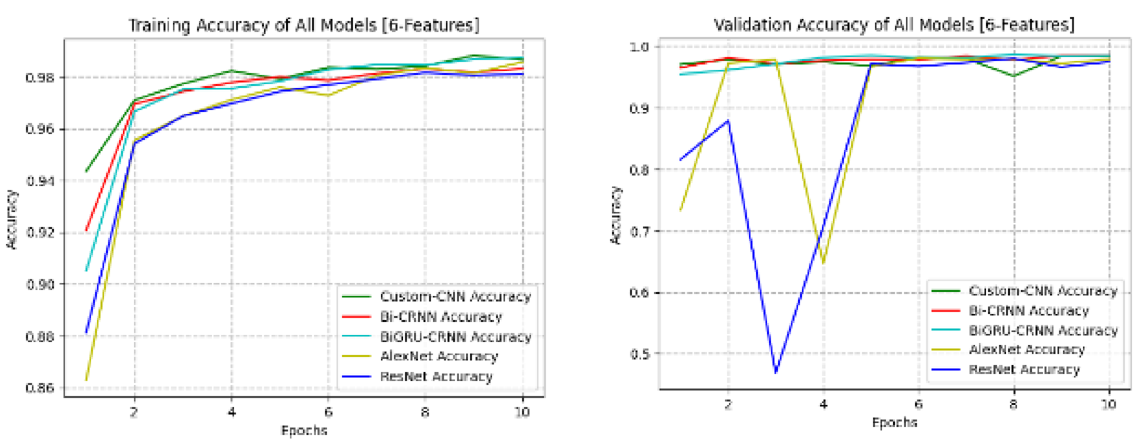
Fig. 18: Five Features Cluster result analysis

Table 6: Model Performance Metrics for six features cluster

Model	Precision	Recall	F1-Score	Accuracy	Loss
Training					
AlexNet	0.96	0.96	0.94	0.95	0.13
Bi-CRNN	0.94	0.94	0.98	0.96	0.06
BiGRU-CRNN	0.96	0.96	0.98	0.97	0.04
Custom CNN	0.98	0.98	0.98	0.99	0.02
ResNet	0.94	0.89	0.92	0.92	0.14
Validation					
AlexNet	0.94	0.96	0.92	0.94	0.22
Bi-CRNN	0.96	0.96	0.97	0.96	0.07
BiGRU-CRNN	0.98	0.96	0.97	0.97	0.05
Custom CNN	0.97	0.98	0.97	0.98	0.04
ResNet	0.92	0.92	0.92	0.92	0.26

Six Features Cluster

The 6-Feature Cluster models have a significant rise in the accuracy from 2nd iteration while training. The model providing the best accuracy is Custom-CNN model, the ResNet pre-defined model provides less accurate classification when compared to all the other models. Similarly, the validation accuracy and loss curves depict that the most unstable model is ResNet and AlexNet whose accuracy and loss keeps fluctuating over the period of 10 iterations. The best model is Custom-CNN when compared to all other models in all aspects. The ROC curve and Precision vs Recall curve depicts the area created by all the models and shows a really no difference between them all, there is hardly any instability in terms of precision and recall in these models. Table 6 shows the performance metric results for six features cluster. Figure 19 illustrates the six Feature Cluster result analysis.



DOI: 10.3844/jcssp.2025.2772.2801

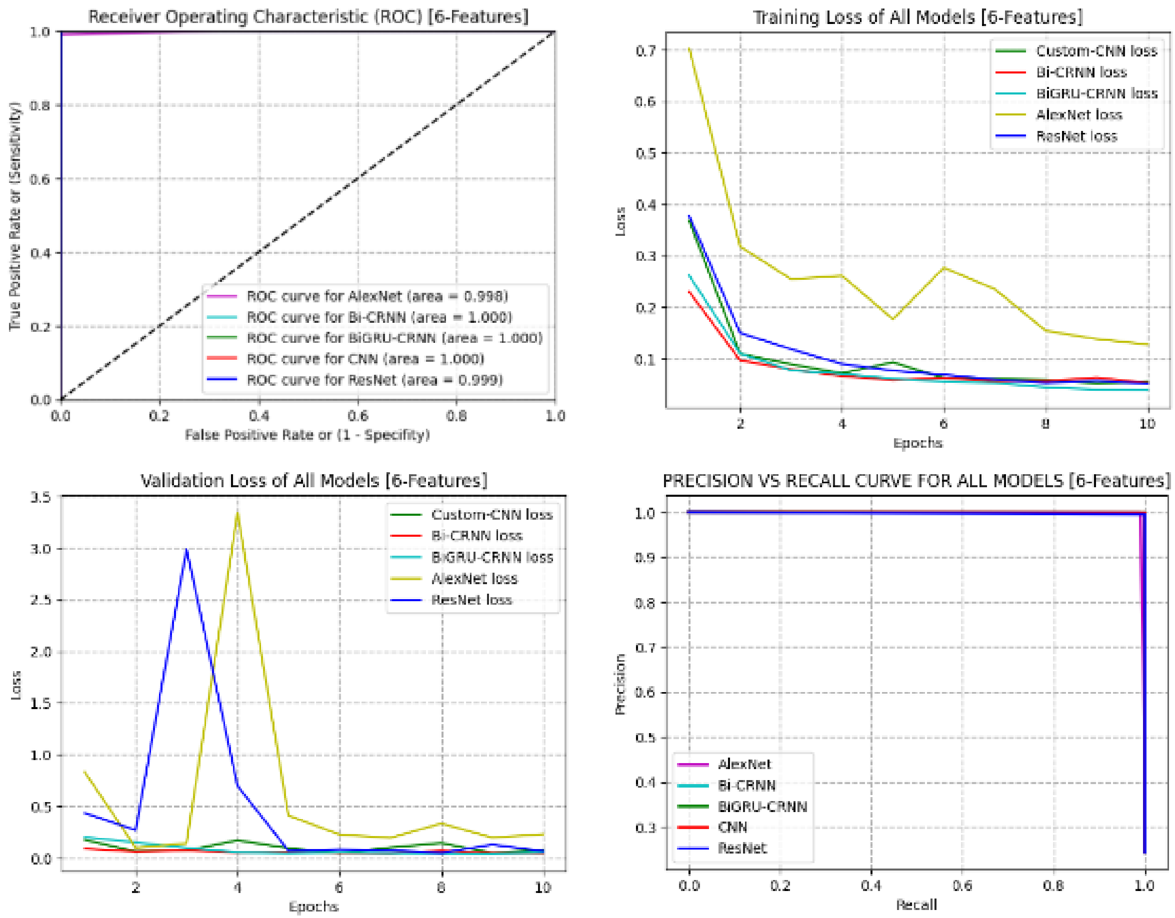


Fig. 19: Six Features Cluster result analysis

By comparing all the above metrics of different feature clusters that are for 2-Feature, 3-Feature, 4-Feature, 5-Feature, and 6-Feature concluded that the audio data when converted to numerical data can be classified effectively with multiple features such that it becomes more effective while classifying the audio files in real-time. While investigating the above result analysis, our research concluded that the most dependent feature clusters are the 5-Feature clusters and 6-Feature clusters since, they have good stability in most of the models when compared to rest of the feature clusters. The pre-defined models AlexNet and ResNet do not provide a stable and accurate classification measure to classify the data in any feature cluster this may be due the fact that they were trained on images to extract features from them and classify them accordingly. The custom-built models provide an accurate and stable measure to classify the audio data accordingly to their classes. The most accurate model in majority of the feature clusters are Custom-CNN and BiGRU-CNN. The Custom-CNN model provided 98.2% of accuracy and BiGRU-CNN provided the 97.9% of accuracy.

Potential Limitations

The confusion matrix provides a clear understanding of the classification model's performance in identifying different sound classes: axe-cutting (Class 0), chainsaw (Class 1), handsaw (Class 2), and negative sounds such as birds, animals, rain, and vehicles (Class 3). A high number of correct classifications, including true positives and true negatives, indicate that the model effectively distinguishes between deforestation-related and irrelevant sounds. Accurate identification of tree-cutting sounds ensures the model's reliability in detecting deforestation activities.

However, Figure 20 shows that the misclassification rates for false positives where non-deforestation sounds are incorrectly classified as tree-cutting sounds and false negatives where tree-cutting sounds are misclassified as non-deforestation sounds are significantly lower than the true positive and true negative values. This analysis suggests that our proposed model performs well in minimizing misclassifications. However, handling misclassification remains a challenge in research.

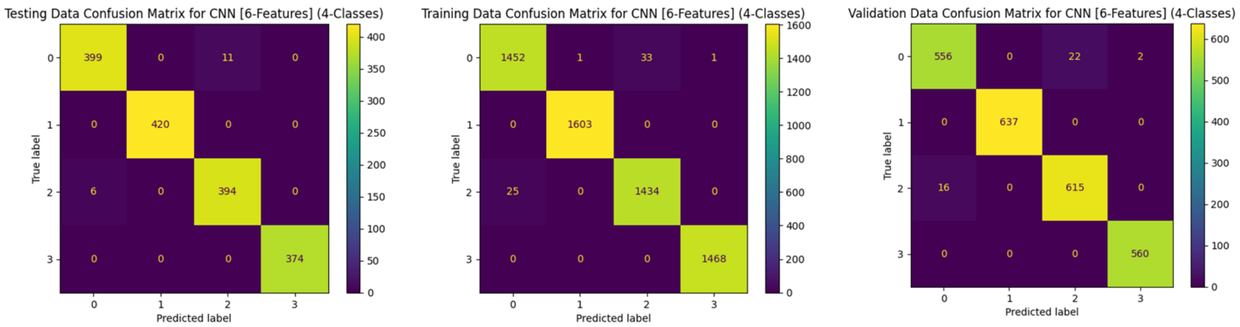


Fig. 20: Confusion matrices for the training, validation, and testing phases of the proposed model

Table 7: Comparison of the Proposed Method with State-of-the-Art Models

Reference	Dataset	Features	Classification models	Result	Environment
Ahmad and Singh, 2022	www.sounddog.com and www.freesound.org.	Spectral Centroid, MFCC	K Means Clustering	92%	Tree cutting sound classification
Mporas *et al.*, 2020	Synthetic data	MFCC	SVM	94.42%	Chainsaw sound classification in the forest
Qurthobi *et al.*, 2025	FSC22	MFCC	Custom CNN	94.4%	Forest sound classification
Andreadis *et al.*, 2021	ESC-50	Mel-Spectrogram, MFCC	Custom-CNN	85%	Tree Cutting
Bandara *et al.*, 2023	FSC22	MFCC, Mel-Spectrogram	Custom-CNN	92.59%	Forest Sound Classification
Paranayapa *et al.*, 2024	FSC22	MFCC, Mel-Spectrogram, Chroma features	MobileNetv3	87.95%	Forest Sound Classification
Proposed	Audioset and Synthetic data	Mel-Spectrogram, MFCC, Chroma, Spectral Contrast, Tonnetz and Spectral Bandwidth	Custom-CNN BiGRU-CNN	98.2% 97.9%	Tree cutting sound classification

Several factors contribute to misclassification, including sound similarity, where environmental noises such as breaking branches, strong winds, or animal calls share frequency patterns with tree-cutting sounds, making differentiation difficult. Feature overlap is another challenge, as extracted sound features like MFCC and Mel Spectrogram may not be entirely distinct across classes. Additionally, dataset imbalance can affect classification accuracy, especially if certain classes have fewer training samples, limiting the model's ability to learn diverse variations. Table 7 presents a comparative analysis of the proposed model against existing research models related to forest environment studies.

Performance of Proposed Customized CNN

In this research, the adopted Customized CNN model yielded better results than all the other baseline models that were used in the research. To effectively support the evaluation and subsequent validation of this improved performance, however, a systematic method of data analysis was used. This also involved the use of an ANOVA test to analyze the mean performance of the CNN model to other existing baseline methods and to determine whether there are indeed statistically significant differences in the results. Moreover, a calibration test was carried out to check various aspects of the model to show that the probability estimated by the model was accurate to the output confidence of the system. Lastly, we applied radar chart plotting analysis to represent the performance comparison of multiple features in order to identify the superiority and inferiority of the proposed Customized CNN model. This multiple strategy has made it possible to give strong support to the model's general performance and ability to manage large data.

ANOVA Test

The Analysis of Variance (ANOVA) test helps in assessing whether the observed variances among group means are due to chance or if there are actual differences among them. This test provided the two concluded results that are F-statistics and P-value. In our research the F-value of 3.7 suggests that the variance among the group means significantly larger than the variance within the groups. This indicates that there is a reasonable amount of difference among the groups being compared. The P-value of 0.0204 is less than the conventional significance level of 0.05. This means that the proposed model can reject the null hypothesis, suggesting that there are statistically significant differences among the group means. Figure 21 demonstrates the ANOVA test analysis of proposed model performance.

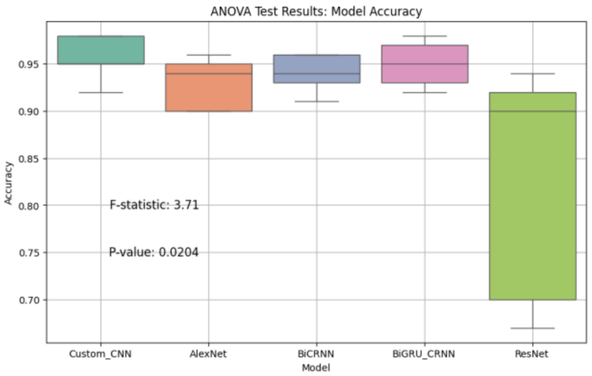


Fig. 21: ANOVA Test result of proposed and pre-trained models

Calibration Curve Analysis

Calibration curve for a multi-class classification model, showing calibration results for each class. Calibration curves are used to measure the accuracy of a model's predicted probabilities by comparing them to the actual outcomes. According to the diagram each class has two lines: One for the CNN model and one for the Baseline models. The classes are labeled as Class 0, Class 1, Class 2, and Class 3. Mean Predicted Probability on X-axis represents the model's predicted probability for each class, averaged over multiple samples. This value ranges from 0 (low confidence) to 1 (high confidence). Fraction of Positives on Y-axis shows the fraction of actual positives for each predicted probability bin. A value of 1 means all samples with a particular predicted probability belong to the positive class, while 0 means none of them do. Perfect Calibration Line (Black Dashed Line) represents perfect calibration. If the model's predictions are perfectly calibrated, the model's predicted probabilities will match the observed fraction of positives, and the lines for each class will align with this dashed line. When a model's line follows the perfect calibration line closely, it indicates that the model is well-calibrated for that class. Deviations from this line suggest that the model is either overconfident (above the line) or under confident (below the line) in its predictions for that class. In our resultant Figure 22 shows that each class has lines for both the CNN and Baseline models, allowing us to compare the calibration of these two models. The proposed customized CNN model's lines for each class are close to the diagonal line, the model is well-calibrated. This means that the model's predicted probabilities match the observed fraction of positives, indicating reliable probability estimates. The CNN model's calibration curve for Class 0 shows some deviation, especially at lower probability levels, but it aligns better with the diagonal line at higher probabilities. The CNN line for Class 1 appears relatively close to the perfect calibration line, suggesting reasonably good calibration for this class. Small deviations are common, but overall alignment is favorable.

Classes 2 and 3 show more fluctuations, indicating potential issues with calibration. These deviations suggest that the CNN model may not be as well-calibrated for these classes.

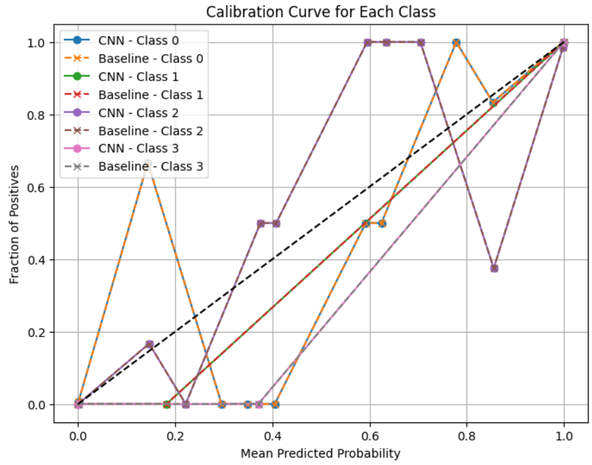
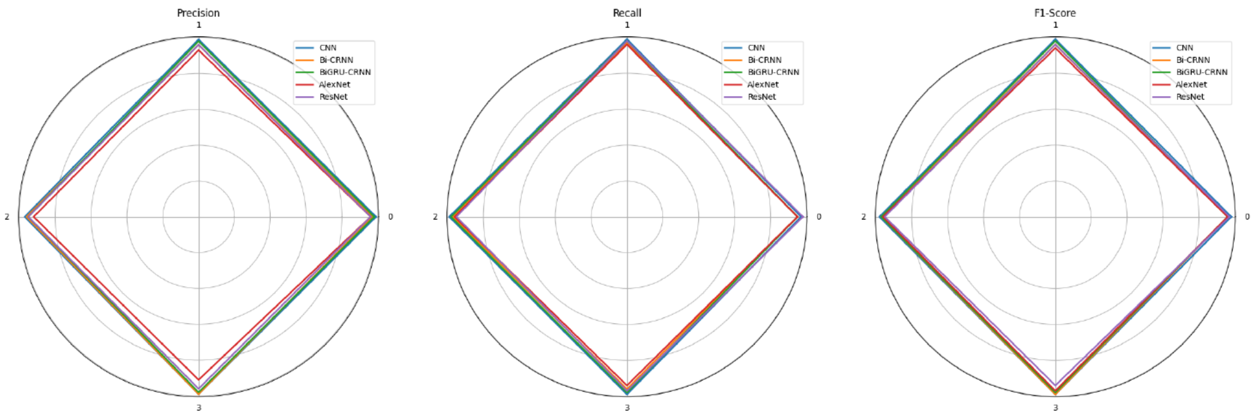


Fig. 22: Calibration Curve of each dataset class for Proposed CNN and Baseline models

Radar Chart Analysis

A radar chart is a graphical method used to display multivariate data in a way that compares different variables across various categories. Here's a breakdown of how to interpret this radar chart: Each axis represents a variable or feature being compared. In this chart, it looks like five different models are being compared: CNN, Bi-CRNN, BiGRU-CRNN, AlexNet, and ResNet. The circular grid represents scales of measurement, allowing a visual assessment of how each model performs relative to the others. Each line plotted on the radar chart represents one model, color-coded for easy identification as shown in the legend. Each axis on the radar chart represents a different performance metric (e.g., precision, recall, accuracy). The customized CNN model and baseline models are plotted on the same chart, with each model forming a polygon. A larger polygon area indicates better performance across those metrics. The closer the plotted line is to the outer edge of the chart, the better the performance of the model in terms of precision. Figure 22 illustrates the performance analysis of Custom-CNN model using Radar Chart.

DOI: 10.3844/jcssp.2025.2772.2801



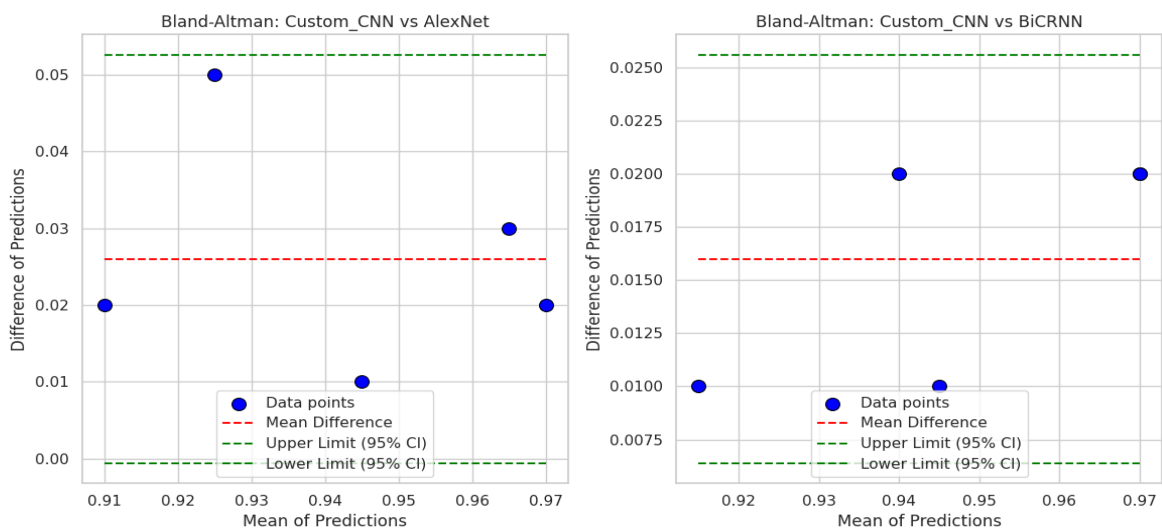
(a) Precision (b) Recall (c) F1-Score

Fig. 22: Performance Analysis of Custom-CNN model using Radar Chart

Bland-Altman Test

The Bland-Altman test also known as the Bland-Altman plot is useful in detecting any systematic bias (consistent differences) and in determining the range within which most of the differences between the models fall. The resultant plotted figure of Bland-Altman test has x-axis that represents the mean of the two predispositions. The y-axis represents the difference between the predictions of Custom CNN and the comparison model. Mean Difference (Red Line) represents the difference between Custom CNN and the comparison model. If the mean difference is close to zero, it indicates that both models are predicting similarly. Upper and Lower Limits (Green Lines) lines represent the 95% confidence interval (±1.96 standard deviations from the mean). They indicate the range within which 95% of the differences between the models are expected to lie. If the mean difference line (red) is significantly above or below zero, it indicates a consistent bias, where one model overestimates or underestimates relative to the other. According to the results the mean difference between Custom CNN and AlexNet is close to zero or slightly positive, it means that Custom CNN is performing slightly better or similarly to AlexNet.

Custom CNN has a slightly positive mean difference compared to BiCRNN, this suggests that Custom CNN is performing slightly better on average. The mean difference between BiGRU CRNN and Custom CNN is close to zero and slightly positive. In this case the Custom CNN performed comparably better than BiGRU CRNN. ResNet shows more significant differences, especially when compared with the Custom CNN model. The mean difference higher and this conclude that Custom CNN consistently performed better than ResNet. Thus, based on the Bland-Altman test, Custom CNN better than ResNet, and performs similarly or slightly better than the other models AlexNet, BiCRNN, BiGRU CRNN. Figure 23 presents the Bland-Altman test based performance analysis of the proposed CNN model with implemented baseline models.



DOI: 10.3844/jcssp.2025.2772.2801

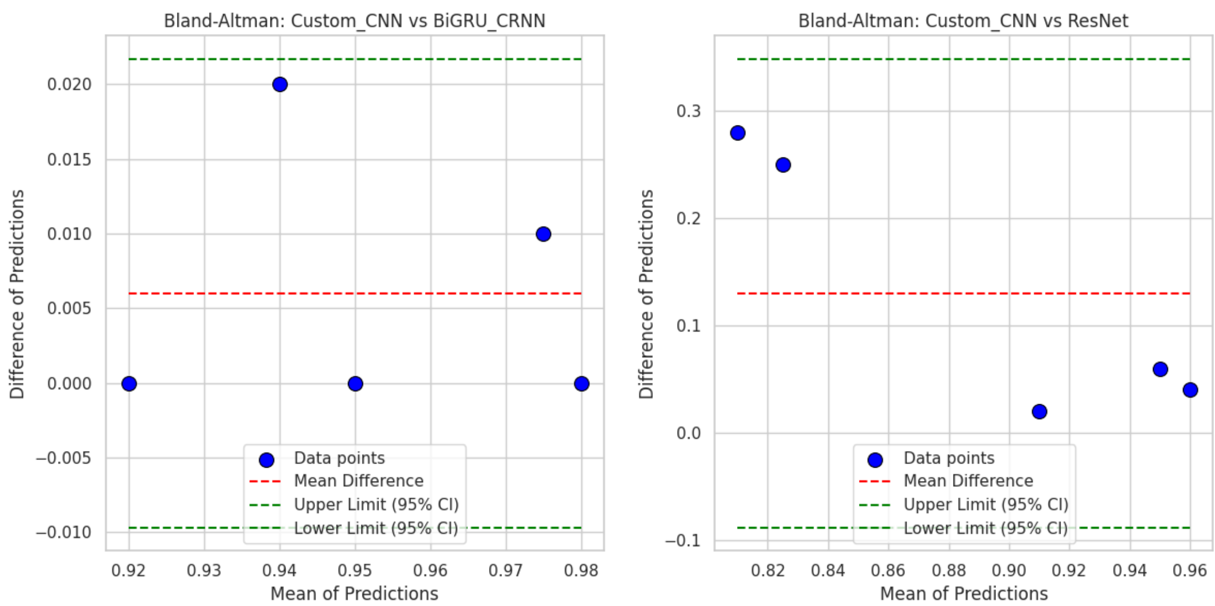


Fig. 23: Bland-Altman Test result of Custom-CNN model with baseline models

Challenges

Sound classification is a solution for Illegal logging in forests, but faces challenges such as hardware limitations, environmental noise, and misclassification risks. This section discussed the key challenges and mitigation strategies, especially for urban-adjacent forests where false positives are a significant consideration.

Environmental Noise and Misclassification Risks

Natural sound overlap challenges distinguishing tree-cutting sounds from other environmental noises, such as wind, rain, and animal calls, which may share similar frequency patterns. Human activity interference further challenge for detection, as sounds from construction, farming, and road maintenance near forests can be miss classified as illegal logging. Echo and reverberation in dense forests can effect on sound waves, that leads to reduce the accuracy of classification models and increasing the risk of misidentification.

Data and Model Challenges

Dataset imbalance can result in biased predictions, as insufficient samples for certain sound classes prevent the model from learning adequate variations. Feature extractions impact classification performance. The methods such as MFCC and Mel Spectrogram may not capture differences between similar sounds. The performance issues arise when models trained in one forest environment fail to perform well in others due to variations in acoustic conditions.

False Positives in Urban-Adjacent Forests

Urban-adjacent forest environments are become challenge due to human-created sounds that lead to false positives in illegal logging. Construction activities, agricultural machinery, road maintenance, and industrial operations may produce sounds similar to tree-cutting, causing misclassification. These false alerts reduce system reliability and divert attention from illegal logging.

Hardware Challenges

Microphones and computing units used in forest monitoring systems maintain durability to harsh weather conditions such as humidity, heavy rainfall, dust, and temperature fluctuations. Ensuring a continuous power supply is another challenge, as remote forest areas often lack access to electricity. Solar-powered systems need long-lasting battery capacity. Microphone sensitivity plays an important role in capturing tree-cutting sounds. Proper placement and the use of directional microphones can minimize background noise and improve detection efficiency.

Conclusion

In this research, we proposed a robust sound event detection model for forest monitoring by implementing deep learning techniques to classify and analyze forest-related sounds. By constructing a comprehensive dataset with various sound classes and applying advanced feature extraction methods, we effectively captured key audio features crucial for distinguishing between different environmental sounds. The evaluation was trained with multiple deep learning models, including a Customized 1D Convolutional Neural Network, Bi-directional Convolutional Recurrent Neural Networks, and pre-trained models like AlexNet and ResNet. Apart from that the Customized CNN achieved 98% accuracy. The results concluded that sound-based monitoring systems are a scalable and efficient solution for detecting deforestation, which can significantly contribute to forest conservation systems.

Future work should focus on enhancing hardware durability with weather-resistant materials and optimizing solar-powered solutions for sustained operation. Advanced noise filtering techniques, such as deep learning-based denoising and adaptive filtering, can improve classification accuracy. Expanding datasets with real-world recordings and utilizing transfer learning will enhance model generalization across different environments. To reduce false positives in urban-adjacent forests, integrating geospatial data, motion detection, and confidence-based alerts can improve detection precision. Additionally, incorporating human-in-the-loop verification will refine system reliability. These improvements will strengthen the effectiveness of sound-based illegal logging detection systems, ensuring more accurate and scalable forest monitoring solutions.

Acknowledgment

I would like to express my sincere gratitude to Prof. Adil Anwar for his invaluable guidance, support, and encouragement throughout the preparation of this article.

Funding Information

The authors have no support or funding to report.

Author’s Contributions

Sallauddin Mohmmad: Conceptualization, methodology, formal analysis, writing original draft, data curation, software, validation, visualization.

Suresh Kumar Sanampudi: Supervision, project administration, writing review and edited.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

Competing Interests

The authors declare that they have no competing interests.

References

Ahmad, S. F., & Singh, D. K. (2022). Automatic detection of tree cutting in forests using acoustic properties. *Journal of King Saud University - Computer and Information Sciences*, *34*(3), 757–763. https://doi.org/10.1016/j.jksuci.2019.01.016

Akbal, E. (2020). An automated environmental sound classification methods based on statistical and textural feature. *Applied Acoustics*, *167*, 107413. https://doi.org/10.1016/j.apacoust.2020.107413

Andreadis, A., Giambene, G., & Zambon, R. (2021). Monitoring Illegal Tree Cutting through Ultra-Low-Power Smart IoT Devices. *Sensors*, *21*(22), 7593. https://doi.org/10.3390/s21227593

Anees, S. A., Mehmood, K., Khan, W. R., Sajjad, M., Alahmadi, T. A., Alharbi, S. A., & Luo, M. (2024). Integration of machine learning and remote sensing for above ground biomass estimation through Landsat-9 and field data in temperate forests of the Himalayan region. *Ecological Informatics*, *82*, 102732. https://doi.org/10.1016/j.ecoinf.2024.102732

Ayoub Shaikh, T., Rasool, T., & Rasheed Lone, F. (2022). Towards leveraging the role of machine learning and artificial intelligence in precision agriculture and smart farming. *Computers and Electronics in Agriculture*, *198*, 107119. https://doi.org/10.1016/j.compag.2022.107119

Bandara, M., Jayasundara, R., Ariyarathne, I., Meedeniya, D., & Perera, C. (2023). Forest Sound Classification Dataset: FSC22. *Sensors*, *23*(4), 2032. https://doi.org/10.3390/s23042032

Behera, T. K., Bakshi, S., & Sa, P. K. (2023). A Lightweight Deep Learning Architecture for Vegetation Segmentation using UAV-captured Aerial Images. *Sustainable Computing: Informatics and Systems*, *37*, 100841. https://doi.org/10.1016/j.suscom.2022.100841

Buchelt, A., Adrowitzer, A., Kieseberg, P., Gollob, C., Nothdurft, A., Eresheim, S., Tschiatschek, S., Stampfer, K., & Holzinger, A. (2024). Exploring artificial intelligence for applications of drones in forest ecology and management. *Forest Ecology and Management*, *551*, 121530. https://doi.org/10.1016/j.foreco.2023.121530

Chavhan, S., Srinitha, B., Kathat, S., Dutta, A. K., & Rodrigues, J. J. P. C. (2024). Design and Development of Drone Seed Dispersal Mechanism using Novel Narcondam Hornbill Algorithm in Barren Lands. *2024 9th International Conference on Smart and Sustainable Technologies (SpliTech).* https://doi.org/10.23919/splitech61897.2024.10612540

da Silva, S. D. P., Eugenio, F. C., Fantinel, R. A., Amaral, L. de P., dos Santos, A. R., Mallmann, C. L., dos Santos, F. D., Pereira, R. S., & Ruoso, R. (2023). Modeling and detection of invasive trees using UAV image and machine learning in a subtropical forest in Brazil. *Ecological Informatics*, *74*, 101989. https://doi.org/10.1016/j.ecoinf.2023.101989

Das, N., Padhy, N., Dey, N., Mukherjee, A., & Maiti, A. (2022). Building of an edge enabled drone network ecosystem for bird species identification. *Ecological Informatics*, *68*, 101540. https://doi.org/10.1016/j.ecoinf.2021.101540

Folliot, A., Haupert, S., Ducrettet, M., Sèbe, F., & Sueur, J. (2022). Using acoustics and artificial intelligence to monitor pollination by insects and tree use by woodpeckers. *Science of The Total Environment*, *838*, 155883. https://doi.org/10.1016/j.scitotenv.2022.155883

Forest Survey of India. (2023). *Inventory reports*. https://fsi.nic.in/inventory-reports

Haq, S. M., Calixto, E. S., Rashid, I., Srivastava, G., & Khuroo, A. A. (2022). Tree diversity, distribution and regeneration in major forest types along an extensive elevational gradient in Indian Himalaya: Implications for sustainable forest management. *Forest Ecology and Management*, *506*, 119968. https://doi.org/10.1016/j.foreco.2021.119968

Jiang, M., Kong, J., Zhang, Z., Hu, J., Qin, Y., Shang, K., Zhao, M., & Zhang, J. (2023). Seeing Trees from Drones: The Role of Leaf Phenology Transition in Mapping Species Distribution in Species-Rich Montane Forests. *Forests*, *14*(5), 908. https://doi.org/10.3390/f14050908

Kasyap, V. L., Sumathi, D. J., Alluri, K., Reddy CH, P., Thilakarathne, N., & Shafi, R. M. (2022). Early Detection of Forest Fire Using Mixed Learning Techniques and UAV. *Computational Intelligence and Neuroscience*, *2022*, 1–12. https://doi.org/10.1155/2022/3170244

Kentsch, S., Lopez Caceres, M. L., Serrano, D., Roure, F., & Diez, Y. (2020). Computer Vision and Deep Learning Techniques for the Analysis of Drone-Acquired Forest Images, a Transfer Learning Study. *Remote Sensing*, *12*(8), 1287. https://doi.org/10.3390/rs12081287

Khan, S., & Khan, A. (2022). FFireNet: Deep Learning Based Forest Fire Classification and Detection in Smart Cities. *Symmetry*, *14*(10), 2155. https://doi.org/10.3390/sym14102155

Khare, S. K., Bajaj, V., & Sinha, G. R. (2020). Hybrid Computerized Method for Environmental Sound Classification. *IEEE Access*, *8*, 124055–124065. https://doi.org/10.1109/access.2020.3006082

Latha, R. S., Sreekanth, G. R., & Suvalakshmi, K. (2023). *Detection of Bird and Frog Species from Audio Dataset Using Deep Learning*. 336–350. https://doi.org/10.1007/978-3-031-28183-9_24

Lohit, G. V. S. (2021). Reforestation Using Drones and Deep Learning Techniques. *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 1320–1325. https://doi.org/10.1109/icaccs51430.2021.9442053

Luo, L., Zhang, Q., Mao, Y., Peng, Y., Wang, T., & Xu, J. (2023). A Study on the Soundscape Preferences of the Elderly in the Urban Forest Parks of Underdeveloped Cities in China. *Forests*, *14*(6), 1266. https://doi.org/10.3390/f14061266

Mondal, P., & Southworth, J. (2010). Protection vs. commercial management: Spatial and temporal analysis of land cover changes in the tropical forests of Central India. *Forest Ecology and Management*, *259*(5), 1009–1017. https://doi.org/10.1016/j.foreco.2009.12.007

Mporas, I., Perikos, I., Kelefouras, V., & Paraskevas, M. (2020). Illegal Logging Detection Based on Acoustic Surveillance of Forest. *Applied Sciences*, *10*(20), 7379. https://doi.org/10.3390/app10207379

Panwar, E., Kukunuri, A. N. J., Singh, D., Sharma, A. K., & Kumar, H. (2022). An Efficient Machine Learning Enabled Non-Destructive Technique for Remote Monitoring of Sugarcane Crop Health. *IEEE Access*, *10*, 75956–75970. https://doi.org/10.1109/access.2022.3190716

Paranayapa, T., Ranasinghe, P., Ranmal, D., Meedeniya, D., & Perera, C. (2024). A Comparative Study of Preprocessing and Model Compression Techniques in Deep Learning for Forest Sound Classification. *Sensors*, *24*(4), 1149. https://doi.org/10.3390/s24041149

Purwins, H., Li, B., Virtanen, T., Schluter, J., Chang, S.-Y., & Sainath, T. (2019). Deep Learning for Audio Signal Processing. *IEEE Journal of Selected Topics in Signal Processing*, *13*(2), 206–219. https://doi.org/10.1109/jstsp.2019.2908700

Permana, S. D. H., Saputra, G., Arifitama, B., Yaddarabullah, Caesarendra, W., & Rahim, R. (2022). Classification of bird sounds as an early warning method of forest fires using Convolutional Neural Network (CNN) algorithm. *Journal of King Saud University - Computer and Information Sciences*, *34*(7), 4345–4357. https://doi.org/10.1016/j.jksuci.2021.04.013

Qadeer, A., Shakir, M., Wang, L., & Talha, S. M. (2024). Evaluating machine learning approaches for aboveground biomass prediction in fragmented high-elevated forests using multi-sensor satellite data. *Remote Sensing Applications: Society and Environment*, *36*, 101291. https://doi.org/10.1016/j.rsase.2024.101291

Quinn, C. A., Burns, P., Gill, G., Baligar, S., Snyder, R. L., Salas, L., Goetz, S. J., & Clark, M. L. (2022). Soundscape classification with convolutional neural networks reveals temporal and geographic patterns in ecoacoustic data. *Ecological Indicators*, *138*, 108831. https://doi.org/10.1016/j.ecolind.2022.108831

Qurthobi, A., Damaševičius, R., Barzdaitis, V., & Maskeliūnas, R. (2025). Robust Forest Sound Classification Using Pareto-Mordukhovich Optimized MFCC in Environmental Monitoring. *IEEE Access*, *13*, 20923–20944. https://doi.org/10.1109/access.2025.3535796

Rathod, T., Patil, V., Harikrishnan, R., & Shahane, P. (2023). Multipurpose deep learning-powered UAV for forest fire prevention and emergency response. *HardwareX*, *16*, e00479. https://doi.org/10.1016/j.ohx.2023.e00479

Ritts, M., Simlai, T., & Gabrys, J. (2024). The environmentality of digital acoustic monitoring: Emerging formations of spatial power in forests. *Political Geography*, *110*, 103074. https://doi.org/10.1016/j.polgeo.2024.103074

Roe, P., Eichinski, P., Fuller, R. A., McDonald, P. G., Schwarzkopf, L., Towsey, M., Truskinger, A., Tucker, D., & Watson, D. M. (2021). The Australian Acoustic Observatory. *Methods in Ecology and Evolution*, *12*(10), 1802–1808. https://doi.org/10.1111/2041-210x.13660

Serizel, R., Turpault, N., Shah, A., & Salamon, J. (2020). Sound Event Detection in Synthetic Domestic Environments. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://doi.org/10.1109/icassp40776.2020.9054478

Serrurier, A., Zdroik, P., Isler, R., Kornienko, T., Peris-Morente, E., Sattler, T., & Pradervand, J. (2024). Mountain is calling – decrypting the vocal phenology of an alpine bird species using passive acoustic monitoring. *Ibis*, *166*(4), 1338–1353. https://doi.org/10.1111/ibi.13314

Sethuraman, S. C., Tadkapally, G. R., Mohanty, S. P., & Subramanian, A. (2022). iDrone: IoT-Enabled Unmanned Aerial Vehicles for Detecting Wildfires Using Convolutional Neural Networks. *SN Computer Science*, *3*(3). https://doi.org/10.1007/s42979-022-01160-7

Singh, P., Srivastava, P. K., Verrelst, J., Mall, R. K., Rivera, J. P., Dugesar, V., & Prasad, R. (2023). High resolution retrieval of leaf chlorophyll content over Himalayan pine forest using Visible/IR sensors mounted on UAV and radiative transfer model. *Ecological Informatics*, *75*, 102099. https://doi.org/10.1016/j.ecoinf.2023.102099

Singha, C., Swain, K. C., Moghimi, A., Foroughnia, F., & Swain, S. K. (2024). Integrating geospatial, remote sensing, and machine learning for climate-induced forest fire susceptibility mapping in Similipal Tiger Reserve, India. *Forest Ecology and Management*, *555*, 121729. https://doi.org/10.1016/j.foreco.2024.121729

Supriya, Y., & Gadekallu, T. R. (2023). Particle Swarm-Based Federated Learning Approach for Early Detection of Forest Fires. *Sustainability*, *15*(2), 964. https://doi.org/10.3390/su15020964

Tewari, V. P., Sukumar, R., Kumar, R., & Gadow, K. v. (2014). Forest observational studies in India: Past developments and considerations for the future. *Forest Ecology and Management*, *316*, 32–46. https://doi.org/10.1016/j.foreco.2013.06.050

Thepade, S. D., & Chaudhari, P. R. (2021). Land Usage Identification with Fusion of Thepade SBTC and Sauvola Thresholding Features of Aerial Images Using Ensemble of Machine Learning Algorithms. *Applied Artificial Intelligence*, *35*(2), 154–170. https://doi.org/10.1080/08839514.2020.1842627

Vinod, P., Trivedi, S., Hebbar, Raghavendra, & Jha, C. S. (2023). Assessment of Trees Outside Forest (TOF) in Urban Landscape Using High-Resolution Satellite Images and Deep Learning Techniques. *Journal of the Indian Society of Remote Sensing*, *51*(3), 549–564. https://doi.org/10.1007/s12524-022-01646-0

Wägele, J. W., Bodesheim, P., Bourlat, S. J., Denzler, J., Diepenbroek, M., Fonseca, V., Frommolt, K.-H., Geiger, M. F., Gemeinholzer, B., Glöckner, F. O., Haucke, T., Kirse, A., Kölpin, A., Kostadinov, I., Kühl, H. S., Kurth, F., Lasseck, M., Liedke, S., Losch, F., … Wildermann, S. (2022). Towards a multisensor station for automated biodiversity monitoring. *Basic and Applied Ecology*, *59*, 105–138. https://doi.org/10.1016/j.baae.2022.01.003

Wang, M., Chen, M., Wang, Z., Guo, Y., Wu, Y., Zhao, W., & Liu, X. (2024). Estimating rainfall intensity based on surveillance audio and deep-learning. *Environmental Science and Ecotechnology*, *22*, 100450. https://doi.org/10.1016/j.ese.2024.100450

Zhao, Q., Yu, S., Zhao, F., Tian, L., & Zhao, Z. (2019). Comparison of machine learning algorithms for forest parameter estimations and application for forest quality assessments. *Forest Ecology and Management*, *434*, 224–234. https://doi.org/10.1016/j.foreco.2018.12.019