DETECTING LOGGING OF FOREST TREES USING DEEP LEARNING MODELS

(Sound Event Detection)

ABSTRACT

Automated detection of certain acoustic signals in environments is becoming an emergent field. One such area explored is the detection of logging of trees in forests. In forests, tree-cutting activities are illegal but due to a shortage of manpower and other resources, governments are not very successful in curbing this menace. One strategy to prevent this is to identify the tree-cutting process as soon as possible so that prompt action may be done to halt it. The simplest method of early detection of tree cutting is to regularly monitor the forest area either manually or using some automatic techniques. As tree cutting generates a lot of noise, it can be detected by regularly monitoring the acoustic signals inside the forest. The tree-cutting sounds are recorded using a microphone, and the sound of various events such as axe knocking, chainsaw sound, and natural sounds such as wind are also recorded in the system and unwanted sounds from this are eliminated using Machine Learning technology. This method is modular when compared to the visual detecting method since it requires a lot of resources and cannot be implemented in a very vast area, it is easy to produce and energy efficient as it relies on audio evidence and uses powerful Machine Learning algorithms. The system can be adapted to different forest features and can be used equally during the day and night. The system is built by collecting audio data of various sounds related to the logging of trees and important features from this audio data are extracted, which is then further used to train the machine learning and deep learning algorithms used to implement this system. This system also holds the capability to extend its use beyond the chosen environment by training the model with suitable audio data for the needed environment.

INTRODUCTION

Forests play a critical part in the preservation of the earth's global biodiversity and ecological equilibrium. In general, forest cover is critical across the world and serves as a measure of the planet's overall health. It has been widely said that forests adequately filter the air, conserve watersheds, reduce erosion, improve water quality, and supply natural resources. Furthermore, forests help to mitigate global warming by absorbing a large amount of carbon dioxide, the principal greenhouse gas, and so help to safeguard the world from climate change. According to numerous estimates, around 1.6 billion people worldwide rely on forest settings for their livelihoods, and roughly 60 million indigenous people rely heavily on forests for their existence and subsistence. In India, forests cover a wide area of land of which the major part remains unexplored due to the landscapes they are present in this becomes an advantage to the people who want to exploit the natural resources present in it. As a result, increasing the efficacy of surveillance for unlawful fires and logging is required. On the other hand, because of a lack of human resources, environmental money, and other resources, onsite monitoring by staff patrols with on-ground control and observation towers is too expensive and time-consuming to offer capillary and widespread monitoring. Therefore, automated detection approaches are required.

Many variables influence the survival and sustainability of forests. Illegal logging is a major problem that can result in uncontrolled and irreversible deforestation. Furthermore, illegal logging occurs. Because forests maintain about 90% of terrestrial biodiversity, they pose the greatest danger to biodiversity. Furthermore, illegal logging endangers the viability of forest ecosystems and can lead to widespread deforestation, which significantly impacts the environment. Flash floods, landslides, drought, climate change, and global warming are the primary consequences of illegal logging. Illegal logging also reduces government income and may contribute to the growth in poverty. Illegal logging operations have an impact on forest-rich countries as well as numerous countries that import and use various wood-based goods from wood-producing countries. Because of the nature of the activity, it is sometimes hard to correctly determine the scope or volume of illegal logging. Illegal forest operations are projected to cost governments throughout the world between USD 10-15 billion in yearly income. Furthermore, it has been said that in the most vulnerable forest zones, more than half of all logging activities are carried out illegally. Curbing this illegal logging of trees is needed to preserve biodiversity and reduce the opposability of a natural disaster.

The automated system which identifies these illegal activities should be built using a large amount of data, since the system should be constructed using machine learning and deep learning algorithms the data that is going to be used for training this algorithms should not be redundant, they should be easily distinguishable such that the features that will be used to train the models are highly accurate in classifying the sounds to a particular class. In our case the data that is going to be used should be in an audio format, the audio files can either be recorded manually or can be collected through an already available audio dataset over the internet. People who have already worked on a similar problem have chosen data that is already available in machine learning repositories like UCI and Kaggle, namely UrbanSound8k which consists of all the sounds that an urban area can have including domestic animal sounds but the data available is not completely related to our problem, it consists of multiple classes which are related to the environment. In our case, we need audio files which determine sounds such as chainsaw, axe knocking etc. The sounds which are related to trees only which will become the positive labels of our system. Based on the sounds an accurately predicting model will be developed using various algorithms. The various classes that are going to be considered for this problem are chainsaw sounds, axe knocking sounds, wind sounds etc.

In the next phase, where the data has been collected and the audios have been identified as distinguishable, the data needs to be pre-processed. The audio files will be pre-processed by using some readily available tools to match the sample rate of all the audio files as one. Next, the files will be spilt into many parts of ten seconds each either manually or by using the audio tools. If the audio files are not spilt into smaller ones the features cannot be extracted because of the large audio file which creates very high-dimensional features over a very long period. After the above pre-processing steps the features are extracted from the audio files by using the librosa library available in python which has many feature extraction techniques such as Mel-frequency Cepstral Coefficients (MFCC's), Spectral Density, Spectral Contrast, Tonnetz etc. Many people have used the above feature extraction techniques which have given good results over the time, they have used it for many problem statements which require these feature extraction techniques for audio files. This phase is the most crucial phase for the development of the system which determines the vectors that will be used to train the models for developing the automated system. These generated features or vectors can either be single-dimensional or multi-dimensional depending on the way or method that has been chosen by the system developers. This feature extraction phase sets the ground for the training of the models that predict the outcome.

After the feature extraction phase the extracted features from the audio files will be used for training the machine learning and deep learning models where the features extracted will be considered as the processed data whereas the audio files which were used to construct these features will be considered as raw data. The processed data either single-dimensional or multi-dimensional will be split into multiple parts which are training, testing and validation sets where each set determines its significance by its name, the training set will be used to train the model, the validation set will be used to validate the training of the models which tests the model while training itself whereas, the testing set will be used to test the model once the training has been completed successfully. Each set of the data is not redundant, no data of processed data will repeat in any set. Many people who have worked in this domain by using audio files to train the models have gone with the above specified approach, it is the most generalized approach as of now. The various models that can be used to train are Logistic regression, Decision trees, Random Forest, Neural Networks etc. For the specified problem the better model would be a customized neural network which will accurately predict the class of the audio being given or recorded.

In this article, we determine our contribution in the field of machine learning using acoustic signals, we introduce an acoustic surveillance-based methodology for detecting logging in a forest. The presented methodology is modular and since it relies on audio evidence, it can be adapted to different forest characteristics and can be operated equally well during day and night. We have used the same flow of the system development cycle mentioned in the above paragraphs, brief research has also been performed on the systems that have been developed in this field by other authors and scientists. The data has been collected by us manually by recording the sounds of the needed classes and by using the AudioSet framework of google which provides the sounds of the needed classes. The popular feature extraction techniques have been used to extract the features from the sounds and custom developed models have been used for training and developing the system. At the end the best accurate model will be developed for the determined problem. By this work we hope to contribute and help develop a good model which will be accurate in identifying sounds and alerting the concerned authorities. The aim of this system will be modular and can be used for implementation in any domain.

Natural Resources (Forests) – Impact of Exploiting Forests – Dataset – Feature Extraction – Machine learning algorithms – Our Contributions

LITERATURE SURVEY

Jia-Ching Wang et al. [1] presented effective ambient sound identification system for home automation in this paper. Depending on the sound classes that are discovered, certain home automation services can be activated. To achieve they did it using two main methods: frame-based multiclass support vector machines and independent component analysis Mel-frequency cepstral coefficients for sound detection, respectively, and signal-to-noise ratio-aware subspace-based signal augmentation. They achieved an accuracy of 86.7% with this method.

Siddharth Sigtia et al. [2] Compared different machine learning algorithms as a function of their computing cost in this study. Comparing the rate of performance deterioration that happens when different Automatic Environmental Sound Recognition algorithms are scaled down by a comparable number of computing operations is a corollary to this subject since one class of algorithms may be more resistant to downscaling than another. Finally, they found out that GMM provide a low computational cost.

Shrikanth Narayanan et al. [3] Focused on the identification of ambient noises in this study, with a special emphasis on feature extraction using the matching pursuit (MP) approach. When other audio characteristics, such as MFCCs, fall short in describing sounds, MP offers a solution. When it comes to background noise, they are more durable. The unique use of MP for feature extraction and its usage in unstructured audio processing is the paper's contribution.

Ishitaq Ahmad et al. [4] described a method in this research that uses the hidden Markov model (HMM) approach for classification and the Mel frequency cepstral coefficients (MFCCs) methodology for feature extraction to identify drones from the noises made by their propellers. Two feature vector approaches (one utilising twenty-four MFCCs and the other using the proposed thirty-six MFCCs) are used in the feature extraction step, The HMM-based classifier is then trained using the extracted features giving an accuracy of 100% with drone sounds.

Geard Roma et al. [5] address the issue of feature aggregation for auditory scene recognition in unlabelled audio. They specify a fresh set of descriptors that may be derived from a time series of audio descriptors' similarity matrix using Recurrence Quantification Analysis (RQA). In the framework of the AASP D-CASE [6] competition, they examine their applicability for ambient audio identification when paired with conventional feature statistics.

Agnes Incze et al. [6] presents and evaluates a CNN system for categorising bird sounds using various setups and hyperparameters. A dataset obtained from the Xeno-canto bird song sharing portal, which offers a sizable collection of labelled and categorised recordings, is used to fine-tune the MobileNet pre-trained CNN model. Spectrograms produced from the downloaded data serve as the neural network's input. The accuracy decreases whenever the classes are increased, they have achieved an accuracy of 80% with two classes.

Qiang Yu et al. [7] provide a spike-based paradigm for the ESR problem from a more cerebral standpoint in this paper. Thier framework is a unifying system that consistently combines the three key functional components of fast learning, robust readout, and sparse encoding. Their findings demonstrate the benefits of multispike learning and serve as a selection guide for different spike-based advancements.

Boon-Yaik Ooi et al. [8] did this research to assess how practical it is to listen for machine operations. It is done using sound recognition for operation status tracking. They specifically assess the efficiency of Mel Frequency Cepstral Coefficient (MFCC) to identify actual machine sound and infer the machine's operational condition. The trial findings, which indicate an accuracy of 95.4% and loss of 0.04% percent, are encouraging.

O.K. Toffa et al. [9] introduce a novel method for categorising ambient noises that combines audio data with a textural feature called a local binary pattern (LBP). ESC-10 and ESC-50 datasets were used to assess their system using traditional machine learning techniques such support vector machines (SVM), random forests, and k-nearest neighbour (kNN). The outcomes demonstrated that the LBP features performed better than the traditional audio characteristics. Their best mixed model, which combines LBP features and audio descriptors, produces results that are at the cutting edge of ambient sound categorization, scoring 88.5% on ESC-10 and 64.6% on ESC50.

Luka Vujosevic et al. [10] use image classification to address the sound classification issue, they associate sound files with their corresponding picture representations, namely the mel spectrogram, tonal centroid, spectral contrast, and chromagram, and train a CNN deep learning network using these image representations. Using 10-fold cross validation, the suggested approach obtains a mean accuracy of 73%. Given the nature of the dataset and the fact that ambient noises are far more difficult to categorise than music and speech, this result is quite satisfying. The experimental findings also demonstrate a significant accuracy advantage of the deep learning technique over fully connected NN 59% accuracy.

Paper No.	Paper Name	Type of Data set	Feature Extraction	Models Used	Results (Accuracy)
[1]	Robust Environmental Sound Recognition for Home Automation	Home Sounds Events	ICA-transformed MFCC's, Perceptual Features	SVM	File: 83.5% Frame: 86.7%
[2]	Automatic Environmental Sound Recognition: Performance Versus Computational Cost	Audio data environmental sounds	MFCC's, Spectral Centroid, Spectral Flatness, Spectral rolloff and zero crossing rate	GMM, SVM	GMM EER: 14.0% SVM EER: 13.5%
[3]	Environmental Sound Recognition Using MP-Based Features	Environment Sound Events from audio data	Matching Pursuit (MP)	KNN, GMM	MP: 72.5% MFCC: 70.9% MP+MFCC: 90%
[4]	Hidden Markov Model based Drone Sound Recognition using MFCC Technique in Practical Noisy Environments	Drone Sounds, Environment Sounds	MFCC	GMM	Accuracy: 80%
[5]	Recurrence Qualification Analysis Features for Environmental Sound Recognition	Auditory events, Environmental Sound	RQA, MIR, AR	HMM	Frame-based: 1.46 Event-based: 3.33 Class - based: 3.41
[6]	Bird Sound Recognition Using a Convolutional Neural Network	Bird CLEF	STFT, Colormap, Normalization	CNN	Accuracy: 74%
[7]	Robust Environmental Sound Recognition with Sparse Key-Point Encoding and Efficient Multi spike Learning	Speech Babble	KP Encoding Frontend, STFT	CNN, DNN	Accuracy: 90%
[8]	Non-Intrusive Operation Status Tracking for Legacy Machines via Sound Recognition	Machine Sound Events	FFT, MFCC	Euclidean distance function, DTW	Accuracy: 95.4%, Error rate: 0.04%
[9]	Environmental Sound Classification Using LBP and Audio Features Collaboration	ESC-50	MFCC, GFCC, CQT	CNN, SVM	Accuracy: 88.5%
[10]	Deep learning-based classification of environmental sounds	UrbanSound8k	Mel-Spec, Tonnetz, Spectral Features	CNN	Accuracy: 73%

IMPLEMENTATION

Methodology:

One of the most exciting and developing fields in ML research is surveillance. There are many situations across the world when it is important to identify forest anomalies quickly and accurately. Recent developments in machine learning have produced a promising performance in several of its application areas. The goal of this project is to employ machine learning to detect logging of trees in the forest so that authorities are notified on time, hence eliminating the need for labour of surveillance. There are several numbers of sounds produced in a forest, which is a vast ecosystem with many types of plant species from a sapling to a largest tree. The system being developed will record the sounds that are occurring in the forest and pre-process it by analysing the created metadata by the system for the recorded data from the microphone, while keeping in mind the audio sampling metrics along with converting the data into time series data, feature extraction is done and the trained sound model is used to classify the sounds detected after the feature extraction. In this classification if the model finds any sounds that are related to the logging sounds, then the system designed will send a message to the relevant authorities. This process may occur when logging of a tree(s) is detected. The process flow of the logging detection system will be recording, pre-processing, storing, extracting features, classifying, detection and output which is alerting the authorities in the given case, and fluidity of the processes lies on the number of sounds being recorded and stored for each event the fluidity can increase exponentially.

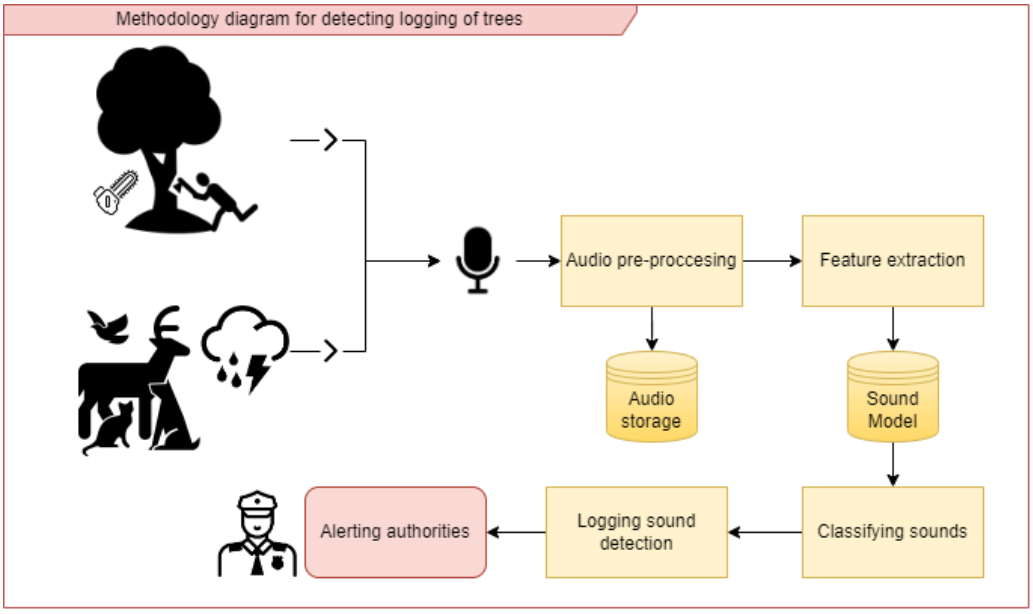

Fig. 1 Methodology of the System.

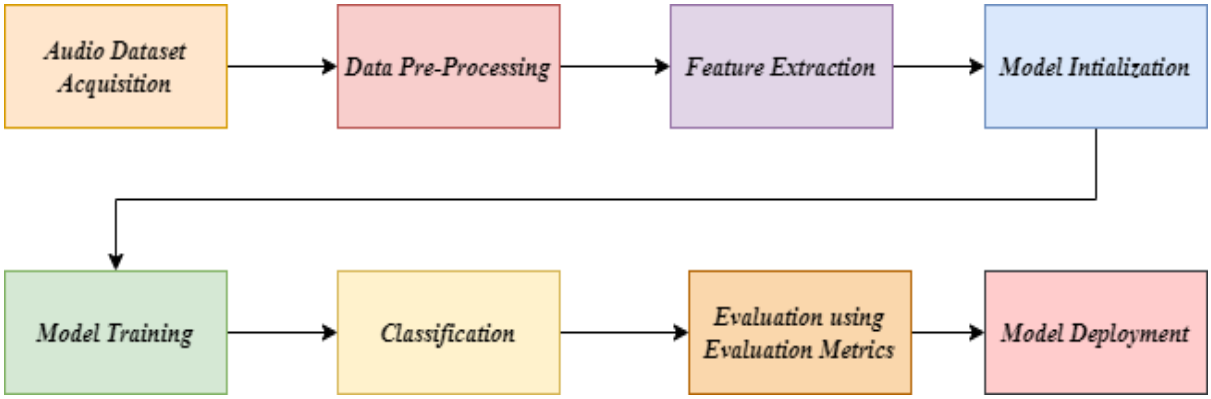

Fig. 2 Workflow of the Audio classification System**.**

Audio Dataset:

A dataset is defined as a collection of similar records. It is generally stored in tabular formats in which a column describes each parameter, and each row represents a separate record. These datasets can be used in artificial intelligence and data science to generate models for certain predictions. It can also be used for visualization and analysis of data for various purposes. But here in our application a dataset in a tabular format cannot be applied since the data is in waveform (i.e., Audio Signals) so, the data that we have is audio recordings of the sounds which can be classified as illegal logging of trees like cutting sounds, sawing sounds, falling sounds, and any data relating to the wood logging activity. As per our concern we have selected six classes for now to which are:

- Axe cutting Sounds.
- Chainsaw Sounds.
- Hand Saw Sounds.
- Rain & Thunder Sounds.
- Wind Sounds.
- Forest Sounds.

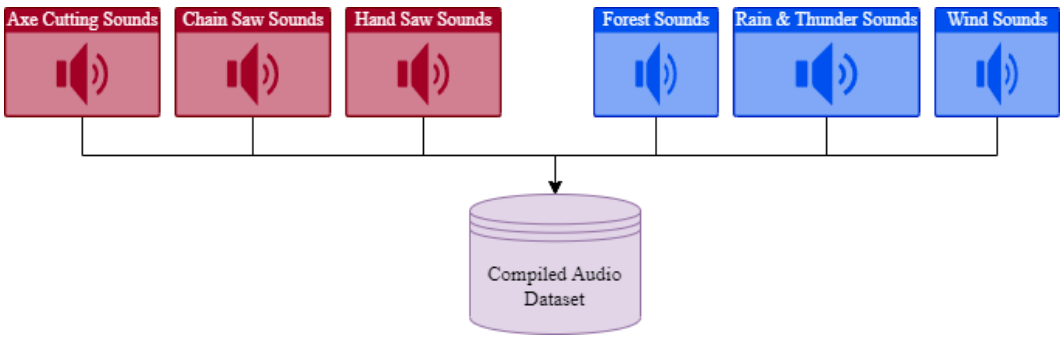

Fig. 3 Compiled Dataset.

Each Class of Sound taken here is from different resources where majority of the audio samples were collected from video streaming service, and audio-set framework of google which provided the sounds for specific classes directly. Some of the sounds were collected manually by recording the needed sound in a particular environment. The audio files collected were around three hours long for each class which were split into multiple samples based on the sampling rate and class. Recorded sounds are then chunked down to ten second samples and are down sampled to a fixed sample rate. The samples in each class taken are 1,200 each from the split samples. The below figures represent the samples before and after consideration.

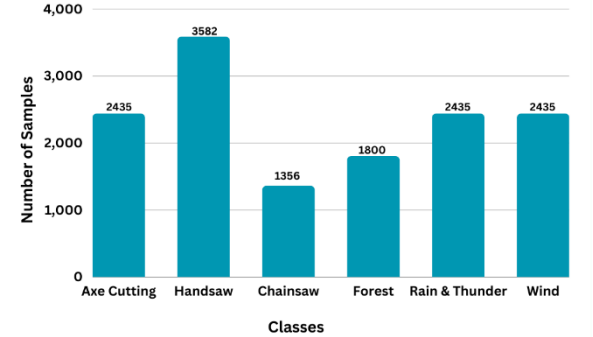

Fig. 4 Imbalanced Dataset.

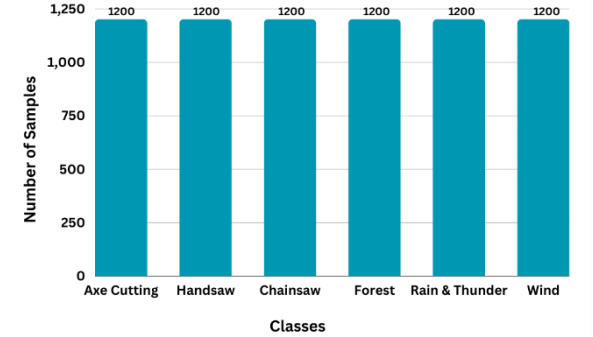

Fig. 5 Imbalanced Dataset.

Pre-processing and Visualization:

Here, we pre-process and analyse the audio data into visual forms by using the IPython.display.Audio module present in python programming language, below is the representation of audio classes that are taken into consideration in wave form. The pre-processing steps involve sample rate conversion, bit-depth manipulation etc.

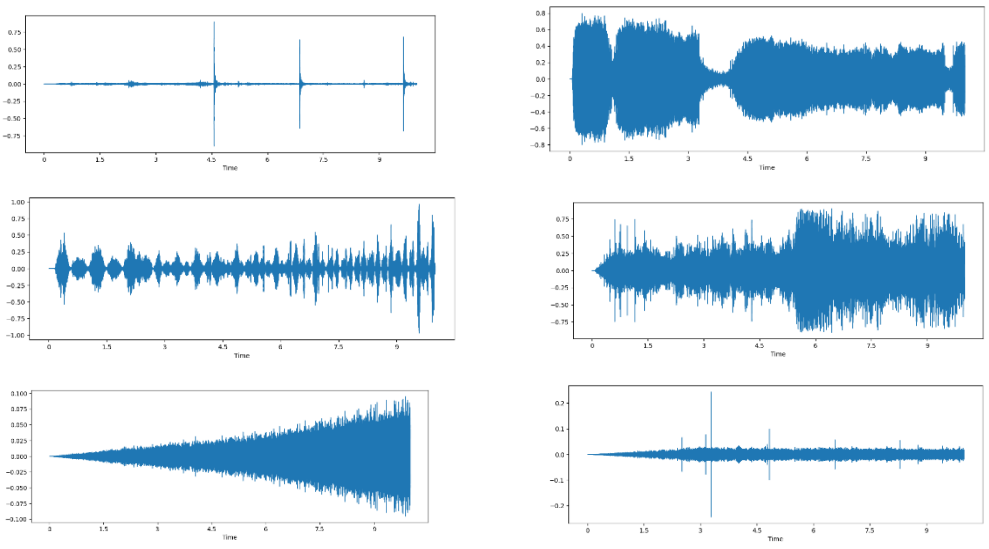

Fig. 6 Audio files Visualization (Six classes).

Feature Extraction:

The process of reducing a large initial set of data into a manageable size for machine learning is known as feature extraction. A starting set of unprocessed raw data is divided into smaller, more manageable groups using the dimensionality reduction method. These enormous data sets have a lot of variables, which means processing them takes a lot of computing power. The term "feature extraction" refers to techniques for choosing and/or combining variables into features, which significantly reduces the amount of data that needs to be processed while properly and fully characterising the initial data set. The feature extraction approaches for audio data work in tandem with the audio pre-processing techniques used on the audio data. The most popular feature extraction techniques are Mel Frequency Cepstral Coefficients (MFCC), Linear Prediction Coefficients (LPC), Linear Prediction Cepstral Coefficients (LPCC), Line Spectral Frequencies (LSF), Discrete Wave Transform (DWT). These methods are helpful in extracting the crucial features form the audio signals that would help us detect the logging of trees. These methods take the processed audio data as input and give vectors as the output which will be used for training and testing purposes.

The Feature extraction techniques used for our purpose are: MFCC, Spectral Contrast, Mel-Spectrogram, Chroma, and Tonnetz. We get a 2-dimension array of the specified features; we calculate the mean of each row for the above features and combine all the mean of all the features in a single one dimension array which will further be saved into a excel or comma separated values file which will be further used for the training of the model. The MFCC extracted consists of 40 Columns in the dataset, whereas Mel-Spectrogram consists of 127 Columns, Chroma consists of 12 columns, Spectral Contrast consists of 7 Columns and Tonnetz consists of 6 Columns and finally a class label which specify which class do they belong to, this will be the final dataset that will be used to train the model accordingly.

Model Training Features	Columns
MFCC	40
Chromagram	12
Mel Spectrogram	128
Spectral Contrast	7
Tonal Contrast	6

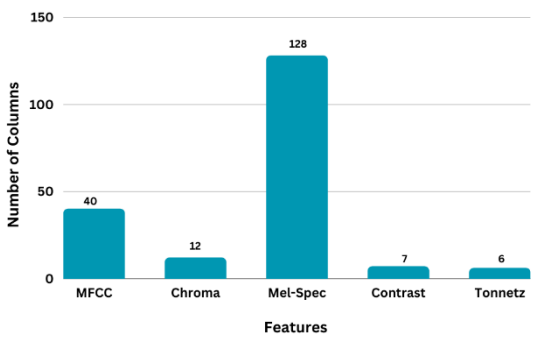

Table. 1 No. of Columns in the Dataset.

Model Initialization & Training:

Giving the training data as input to the model for it to alter the parameters and make predictions is called model training. It is then evaluated on the testing data using accuracy score, confusion matrix, specificity etc. In our case the model will be trained using the vectors that are generated by the feature extraction techniques, these vectors will be passed to a Convolution Neural Network (CNN) and Bi-Directional Convolutional Recurrent Neural Network (Bi-CRNN) to train it for the prediction and classification of the logging sounds. The best model will be selected based on the accuracy. The model will be trained by splitting the feature extracted data which will be split into training and testing where the training data size will be around 80% and testing data size will be around 20%. The training data will be used to train the model whereas the testing data will be used to test the model. The description of the models is given below:

1. Convolutional Neural Network (CNN):

This model is used widely to classify the images usually but, in our consideration, we are going to classify audio using this model. Since the data we are using is linear, so we used a one-dimensional application of the convolutional layer known as (Conv1d) which is implemented using the keras library. The architecture of the model is depicted below:

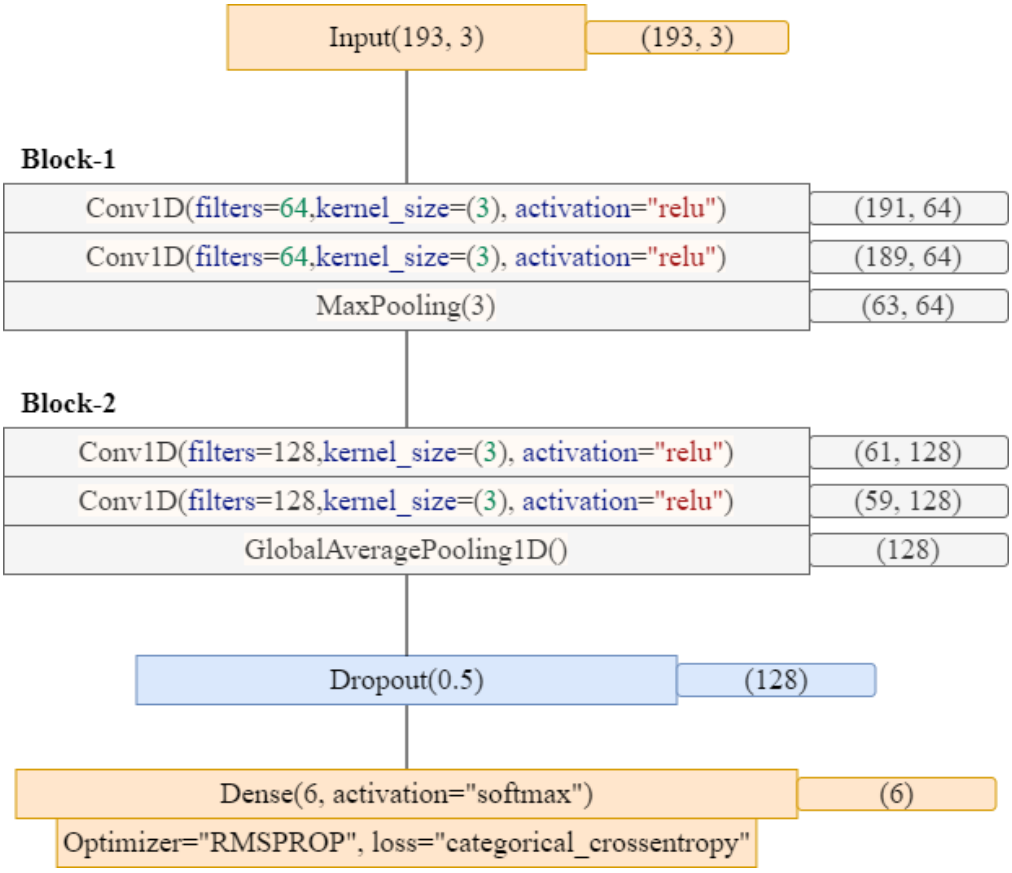

Fig. 7 CNN Architecture.

Figure 7 is the architecture of a Convolutional neural network with Sequential model. The model consists of eight layers in which Conv1D layer with 64 filters and a kernel size of 3. It has 256 parameters, a Conv1D layer with 64 filters and a kernel size of 3. It has 12352 parameters, MaxPooling1D layer, which performs max pooling on the output of the previous layer, Conv1D layer with 128 filters and a kernel size of 3. It has 24704 parameters, Conv1D layer with 128 filters and a kernel size of 3. It has 49280 parameters, GlobalAveragePooling1D layer, which performs average pooling on the output of the previous layer. Dropout layer, which applies dropout regularization to the output of the previous layer. Dense layer with 6 units and activation function, this is the output layer of the model and has 774 parameters. The output shape of each layer is shown, along with the number of parameters for each layer. The total number of parameters for the model is 87,366, and all of them are trainable. Therefore, there are no non-trainable parameters.

2. Bi-Directional Convolutional Recurrent Neural Network (Bi-CRNN):

This model is widely used as a substitute to CNN which usually gives better accuracy than other models, the layers used in this model are also one-dimensional application of the actual layers present in the library. This model is implemented using multiple layers such as Long-Short term memory (LSTM) and Convolutional Layers. The architecture of the model is specified below:

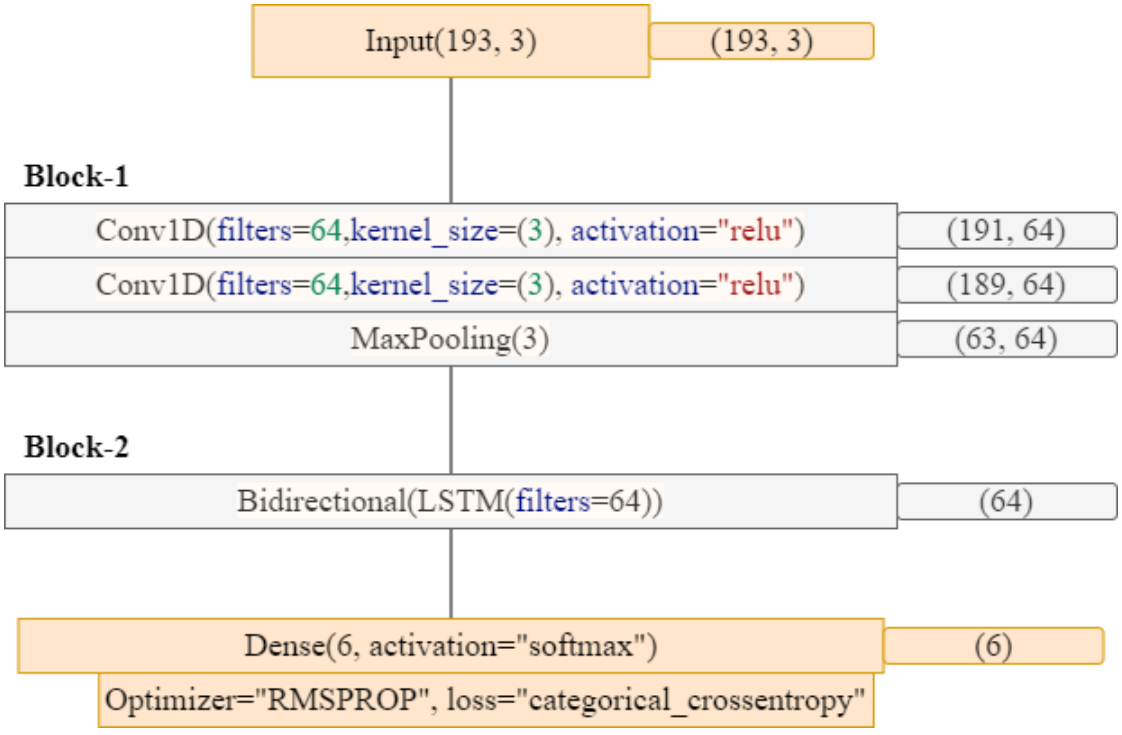

Fig. 8 Bi-CRNN Architecture.

Figure 8 is the architecture of a Bi-Directional Convolutional Recurrent Neural Network with Sequential model. The model consists of five layers in which Conv1D layer with 64 filters and a kernel size of 3. It has 256 parameters, a Conv1D layer with 64 filters and a kernel size of 3. It has 12352 parameters. MaxPooling1D layer, which performs max pooling on the output of the previous layer. Bidirectional layer with a GRU layer, which implements a bidirectional version of the GRU layer. It has 66048 parameters. Dense layer with 6 units and an activation function, this is the output layer of the model and has 774 parameters. The output shape of each layer is shown, along with the number of parameters for each layer. The total number of parameters for the model is 79,430, and all of them are trainable. Consequently, there are no non-trainable parameters.

Model Testing:

In machine learning, model testing is referred to as the process where the performance of a fully trained model is evaluated on a testing set. The testing set consisting of a set of testing samples should be separated from both training and validation sets, but it should follow the same probability distribution as the training set. Here, the testing dataset is used to test the performance of the fully trained model on the audio dataset and the performance analysis is done on it. If the model is giving Feasible results, then it is deployed in real time and the logging of trees can be detected effectively.

Classification Metrics:

Classification metrics are tools used to evaluate the performance of a classification model by measuring how well it can correctly classify examples into their respective classes. There are various metrics that are commonly used to evaluate a model's performance, such as accuracy, precision, recall, F1-score, confusion matrix, and ROC curve. Accuracy measures the percentage of correctly classified examples out of all examples in the dataset, while precision measures the proportion of true positives out of all examples classified as positive. Recall measures the proportion of true positives out of all actual positive examples. The F1-score provides a balance between precision and recall by summarizing the overall performance of the model. A confusion matrix is a useful tool for visualizing the performance of the model and identifying areas where it may be making errors. Finally, an ROC curve is used to evaluate the performance of a binary classifier at different thresholds. The choice of classification metric depends on the specific application and desired trade-offs between different types of errors.

RESULTS

Below are the results of the executed models from the previous section. These results help us understand the prediction accuracy of the models and how well a model can classify the sound to their respective class.

1. Convolutional Neural Network (CNN):

	Precision	Recall	F1-Score	Support
0	0.98	0.94	0.96	176
1	0.98	0.95	0.97	200
2	0.99	1.00	1.00	173
3	0.98	1.00	0.99	192
4	0.94	0.86	0.90	167
5	0.83	0.95	0.89	172
Accuracy			**0.95**	1080
Macro avg	0.95	0.95	0.95	1080
Weighted avg	0.95	0.95	0.95	1080

Table. 2 CNN Classification report.

Table 2 is an output of a model evaluation for CNN. It shows the performance of the model on a classification task. For each class, the performance of the model is reported for the above-mentioned metrics. In the output, the first line shows the class label, and the subsequent lines show the corresponding precision, recall, f1-score and support. The last three lines provide macro-averaged, weighted average, and accuracy scores, respectively. The weighted-average score is the average of the individual class scores weighted by their support. The accuracy score is the proportion of correct predictions among all predictions. In this output, the model has an accuracy score of **0.95** and the performance of the model is generally good, as evidenced by the high precision, recall, and f1-score scores.

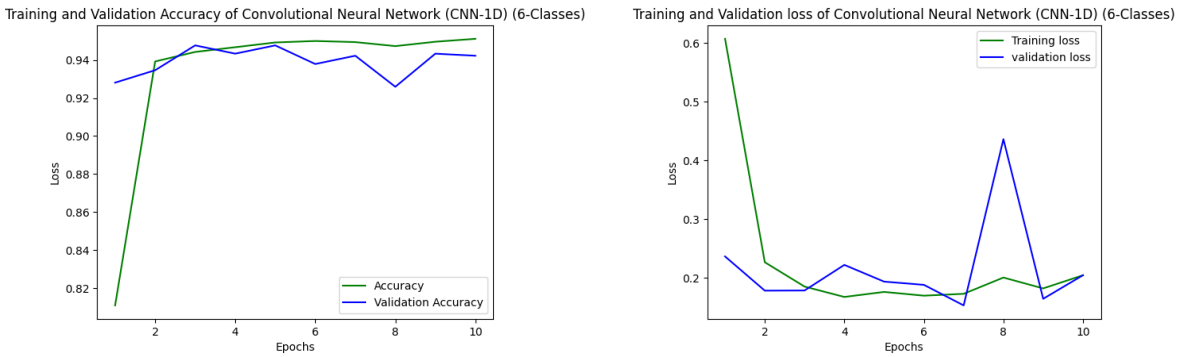

Fig. 9 Training & Validation Accuracy, Loss of CNN.

The training and validation accuracy, loss of the CNN model is shown in the above graphs, it is clearly depicted where the loss and accuracy are increasing and decreasing over the period of execution of **10 epochs** for training the model.

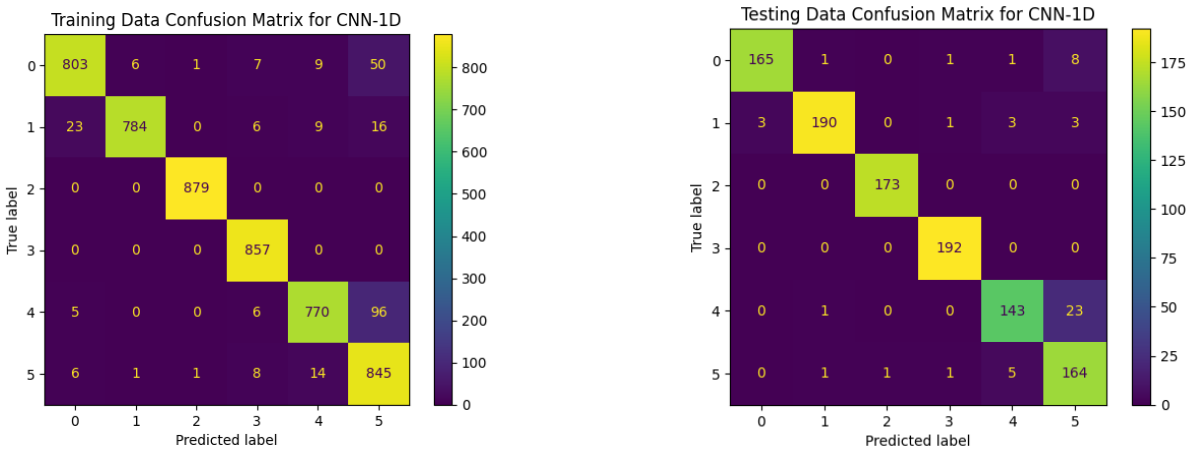

Fig. 9 CNN Training and Testing Data Confusion Matrix.

Figure 9 are the confusion matrices of a model's predictions on a multi-class classification task. These are used to evaluate the accuracy of a model's predictions by comparing the true labels of the data to the predicted labels in CNN-1D. The columns of the matrices represent the predicted labels, while the rows represent the true labels. The elements of the matrix represent the number of samples that have a true label of the rows and a predicted label of the columns. In these matrices, the rows are labeled 0 to 5 and the columns are also labeled 0 to 5, indicating that there are 6 classes in the data. The diagonal elements of the matrices are the number of samples that have been correctly classified. For example, the first row and first column show that 803 and 165 samples were correctly classified as class 0 for training data and testing data respectively. The off-diagonal elements represent the misclassified samples. For example, the first row and second column show that 6 samples were misclassified as class 1 instead of class 0. Overall, the confusion matrices provide a summary of how well the model is performing on the classification task and can be used to identify areas for improvement in the model.

Type	**CNN**	
	Accuracy	**Loss**
Training	0.951	0.204
Testing	0.950	1.680
Validation	0.942	0.204

Table. 3 CNN Accuracy and Loss.

2. Bi-Directional Convolutional Recurrent Neural Network (Bi-CRNN):

	Precision	Recall	F1-Score	Support
0	0.94	0.95	0.95	176
1	0.96	0.97	0.96	200
2	1.00	1.00	1.00	173
3	0.94	1.00	0.97	192
4	0.93	0.91	0.92	167
5	0.94	0.87	0.91	172
Accuracy			**0.95**	1080
Macro avg	0.95	0.95	0.95	1080
Weighted avg	0.95	0.95	0.95	1080

Table. 4 BI-CRNN Classification report.

Table 4 is the result of a Bi-CRNN model evaluation. It displays how well the model performed in a multi-class classification task. The model's performance is presented for each class for the metrics. The class label is displayed on the first line of the output, followed by the matching precision, recall, f1-score, and support. The scores in the final three lines are macro-averaged, weighted-average, and accuracy-specific. The average of each class's scores, each of which is weighed according to its support, is the weighted-average score. The average of the individual class scores, weighted or not, constitutes the macro-average score. The percentage of accurate forecasts among all predictions is the accuracy score. Overall, the model gave an accuracy of **0.95** where the classification of all the classes is >0.91.

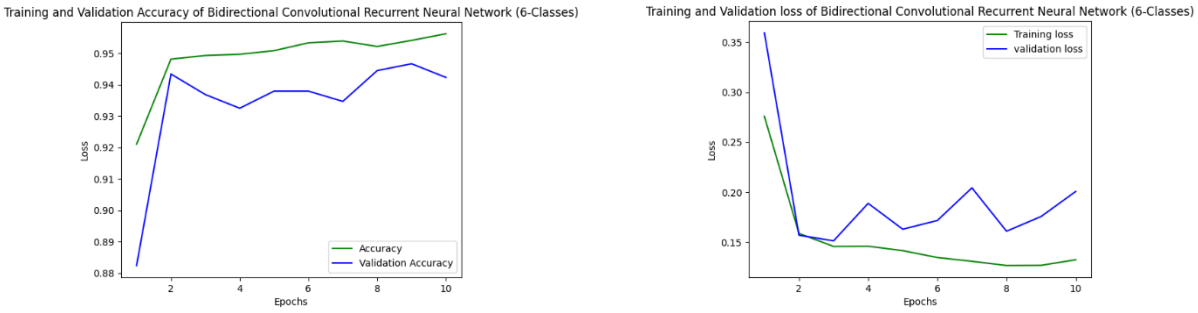

Fig. 10 Training & Validation Accuracy, Loss of BI-CRNN.

The training and validation accuracy, loss of the BI-CRNN model is shown in the above graphs, it is clearly depicted where the loss and accuracy are increasing and decreasing over the period of execution of **10 epochs** for training the model. This model remains stable when compared to the training of CNN model whose graph shows unstable peaks at various epochs, this may cause the model to overfit or underfit.

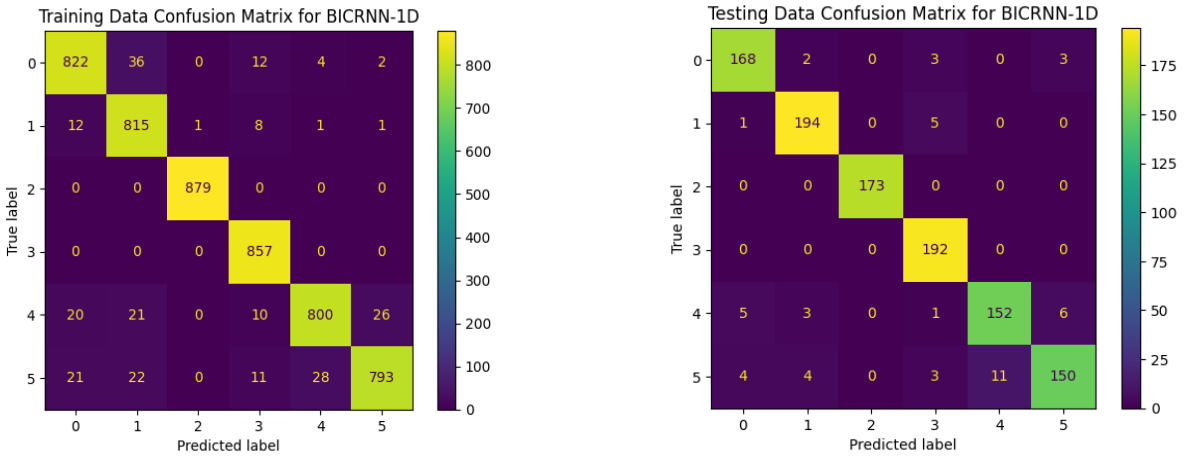

Fig. 11 Bi-CRNN Training and Testing data Confusion Matrix.

Figure 11 is confusion matrix for the BICRNN-1D model shows the performance of the model in classifying the different classes in the training data. The rows represent the actual class labels, and the columns represent the predicted class labels. For class 0, the model has correctly predicted 822 instances out of a total of 857 instances in training whereas 168 instances out of total 192 instances in testing, while it has misclassified 36 instances and 2 instances as belonging to some other classes respectively. Likewise, class 1 model has correctly predicted 815 instances out of a total of 879 instances in training whereas 173 instances out of a total of 194 instances in testing of class 2 , while it has misclassified 12 instances and 3 instances as belonging to some other classes. Similarly, the performance of the model can be seen for the other classes as well. Based on the confusion matrices, we can conclude that the model is performing well in classifying the instances in the training data and testing data. The above confusion matrix is like that of CNN's confusion matrix but has some minor improvement in classification of audio files to their classes which is an advantage when compared to the previous model.

Type	**BI-CRNN**	
	Accuracy	**Loss**
Training	0.956	0.132
Testing	0.952	0.159
Validation	0.942	0.200

Table. 5 BI-CRNN Accuracy and Loss.

3. Model Comparison:

Convolutional Neural Network (CNN-1D) trained on 6 classes data (Axe cutting, Chainsaw, Forest, Handsaw, Rain & Thunder, wind). The model has been evaluated on 216 instances and has achieved an accuracy of 95.09% and a loss of 0.1680. The accuracy metric indicates the proportion of correct predictions made by the model out of the total number of predictions. In this case, the model made correct predictions on 205 out of the 216 instances, giving an accuracy of 95.09%. The loss metric, in this case, the binary cross-entropy loss, is a measure of how well the model is able to predict the target variable. A lower loss value indicates that the model is making better predictions. In this case, the loss value of 0.1680 is relatively low, which suggests that the model is making good predictions.

Bi-Directional Convolutional Recurrent Neural Network (BI-CRNN-1D) trained on 6 classes, and it contains evaluation metrics (Axe cutting, Chainsaw, Forest, Handsaw, Rain & Thunder, Wind). The model's accuracy and loss were both 95.28% and 0.1599 respectively after being tested on 216 instances. The accuracy measure displays the percentage of accurate predictions made by the model out of all forecasts. In this situation, the model accurately predicted 205 out of 216 events, or 95.28% of the time. The binary cross-entropy loss used in this instance serves as the loss metric, which gauges how effectively the model can forecast the target variable. The model is producing better predictions when the loss value is lower. The loss value in this instance is 0.1599, which is pretty low and shows that the model is producing accurate predictions. The BI-CRNN 1-D model has a marginally greater accuracy and a marginally lower loss compared to the CNN 1-D model, which suggests that the BI-CRNN model might perform better on this dataset. It's crucial to remember that high accuracy and low loss numbers do not always imply good performance on unknown data, and that these metrics should be understood in the context of the problem and dataset the model was trained on. To guarantee the model's vigorous and hypothesis, more analysis and testing are required.

Type	**CNN**		**BI-CRNN**	
	Accuracy	**Loss**	**Accuracy**	**Loss**
Training	0.951	0.204	**0.956**	**0.132**
Testing	0.950	1.680	**0.952**	**0.159**
Validation	0.942	0.204	**0.942**	**0.200**

Table. 6 Accuracy and Loss Comparison Table.

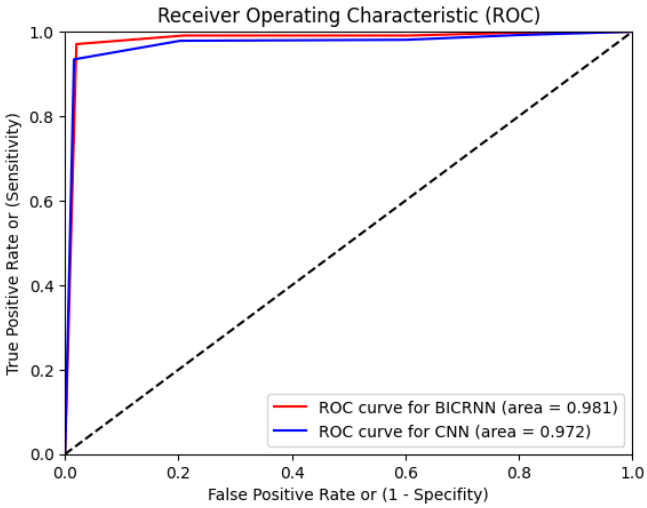

Fig. 12 ROC Curve.

The above figure represents the ROC curve, the ROC (Receiver Operating Characteristic) curve is a graphical representation of the performance of a binary classification model. It is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

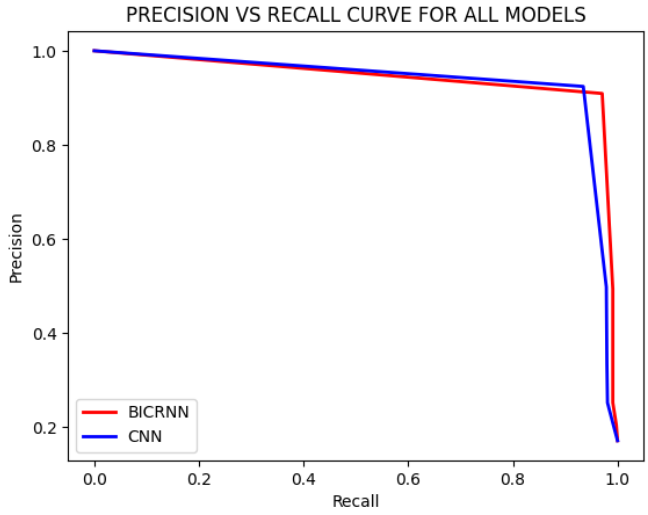

Fig. 13 Precision vs Recall Curve.

The above figure represents the Precision and Recall curve for all the models, they are two important metrics for evaluating the performance of a classification model, particularly in imbalanced datasets. The precision-recall curve is a graphical representation of the trade-off between precision and recall for different classification thresholds. Precision is the ratio of true positive (TP) cases to the total number of positive (TP+FP) cases predicted by the model, while recall is the ratio of true positive (TP) cases to the total number of positive (TP+FN) cases in the dataset. All these comparisons between the models with different metrics help us determine the best model out of all the models trained.

CONCLUSION

Here, a Concept for automatic detection of logging activity in forests using audio recordings was presented. This Concept needs monitoring stations installed in the forest for audio recordings using microphones, and then records audio samples which are then processed and automatically classified into logging or not logging sounds. Two Classification classification algorithms were tested, using well known and widely used audio descriptors during the feature extraction step, with the evaluation focusing on the cutting sound identification whether axe, chainsaw, and Handsaw sound identification during logging in the forests.

We deem that the presented concept greatly contributes as an affordable solution in the development of systems for monitoring forests and for preserving the sustainability of the environment, to reduce illegal deforestation and protect biodiversity. In the presented approach, CNN based models were proposed to recognize and classify six categories of audios. A Res-Net model was also trained to accomplish the classification task. Furthermore, the proposed architecture based on CNN is economically resilient at categorizing the audios. We acquired accuracy rates of 95.2 in the BI-CRNN model. The proposed work performs well for Detection logging of forest trees using sound event detection.

LIMITATIONS AND FUTURE SCOPE

To begin, the model was only investigated using six classes, hence our findings are limited to six audio classes. There are various other classes that also play a vital role in detecting the logging of trees effectively and classifying them, including the sounds which do not occur in the forest. In the future, the proposed approach can be applied to various other sound event detection domains from which the data obtained would be more accurate and less noisy.

Additionally, the method can effectively classify the six different kinds of audios. The proposed work performs well for detecting logging of trees. However, a little performance degradation was observed for audios with excess noise. In the future, we plan to cover this limitation. Moreover, the models developed can also be improved by applying techniques of localization where the precise area from where the sound is being generated is also found with accurate margin.

REFERENCES

[1] Wang, Jia-Ching & Lee, Hsiao-Ping & Wang, Jhing-Fa & Lin, Cai-Bei. (2008). Robust Environmental Sound Recognition for Home Automation. Automation Science and Engineering, IEEE Transactions on. 5. 25 - 31. 10.1109/TASE.2007.911680.

[2] Siddharth Sigtia, Adam M. Stark, Sacha Krstulovic, Mark D. Plumbley, Siddharth Sigtia, Adam M. Stark, Sacha Krstulovic, and Mark D. Plumbley. 2016. Automatic Environmental Sound Recognition: Performance Versus Computational Cost. IEEE/ACM Trans. Audio, Speech, and Lang. Proc. 24, 11 (November 2016), 2096–2107. https://doi.org/10.1109/TASLP.2016.2592698

[3] S. Chu, S. Narayanan and C. -C. J. Kuo, "Environmental sound recognition using MP-based features," 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, USA, 2008, pp. 1-4, doi: 10.1109/ICASSP.2008.4517531.

[4] L. Shi, I. Ahmad, Y. He and K. Chang, "Hidden Markov model-based drone sound recognition using MFCC technique in practical noisy environments," in Journal of Communications and Networks, vol. 20, no. 5, pp. 509-518, Oct. 2018, doi: 10.1109/JCN.2018.000075

[5] G. Roma, W. Nogueira and P. Herrera, "Recurrence quantification analysis features for environmental sound recognition," 2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY, USA, 2013, pp. 1-4, doi: 10.1109/WASPAA.2013.6701890.

[6] Á. Incze, H. -B. Jancsó, Z. Szilágyi, A. Farkas and C. Sulyok, "Bird Sound Recognition Using a Convolutional Neural Network," 2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 2018, pp. 000295-000300, doi: 10.1109/SISY.2018.8524677.

[7] Q. Yu, Y. Yao, L. Wang, H. Tang, J. Dang and K. C. Tan, "Robust Environmental Sound Recognition with Sparse Key-Point Encoding and Efficient Multispike Learning," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 2, pp. 625-638, Feb. 2021, doi: 10.1109/TNNLS.2020.2978764.

[8] B. -Y. Ooi, J. J. -W. Lim, W. -K. Lee and S. Shirmohammadi, "Non-Intrusive Operation Status Tracking for Legacy Machines via Sound Recognition," 2020 IEEE International

Instrumentation and Measurement Technology Conference (I2MTC), Dubrovnik, Croatia, 2020, pp. 1-6, doi: 10.1109/I2MTC43012.2020.9129526.

[9] O. K. Toffa and M. Mignotte, "Environmental Sound Classification Using Local Binary Pattern and Audio Features Collaboration," in IEEE Transactions on Multimedia, vol. 23, pp. 3978-3985, 2021, doi: 10.1109/TMM.2020.3035275.

[10] L. Vujošević and S. Đukanović, "Deep learning-based classification of environmental sounds," 2021 25th International Conference on Information Technology (IT), Zabljak, Montenegro, 2021, pp. 1-4, doi: 10.1109/IT51528.2021.9390124.