Illegal Logging Detection Using Convolutional Neural Networks and Mel Spectrogram

Dr. Vijayalakshmi K
Dept. of Computational Intelligence
SRM Institute of Science and Technology
Kattankullathur, Chennai, India
vijaylak@srmist.edu.in

D Sai Karthik
Dept. of Computational Intelligence
SRM Institute of Science and Technology
Kattankullathur, Chennai, India
sd0683@srmist.edu.in

Sree Vendhan
Dept. of Computational Intelligence
SRM Institute of Science and Technology
Kattankullathur, Chennai, India
ss6132@srmist.edu.in

***Abstract*— Illegal logging is one of the key environmental issues that have led to deforestation, loss of biodiversity, and climate change, and traditional monitoring techniques like satellite monitoring and manual patrolling are expensive, time consuming and inefficient in the thick forest covers. In this paper, a proposal is put forward of an automated audio-based detection system, which is used to detect illegal logging practices through deep learning methods. The system uses Mel spectrogram representations from environmental audio signals and a four-block Convolutional Neural Network (CNN) to classify the sounds as normal forest activity or logging-related sounds such as chainsaws, drilling, and engine noise. In order to enhance detection performance in long-duration recordings, sliding window method with overlapping 4-second segments and a max-confidence selection scheme is used, which makes sure that short and sporadic logging sounds are well logged. The suggested system is embedded into Flask-based web platform which accepts multi-format audio and video as input, allows batch processing of up to ten files at a time, auto-generates PDF and CSV reports, and sends email notifications on positive results. Experimental outcomes using the ESC-50 plus UrbanSound8K dataset, show an accuracy of 97.8 with a precision, recall, and F1 score of 97.3 of the logging class, which is better than when using traditional machine learning models and confirms the efficiency of the proposed solution as a model-scalable and cost-effective method of real-time forest surveillance.**

Keywords — *Illegal Logging Detection, Environmental Sound Classification, Deep Learning, Convolutional Neural Networks, Mel Spectrogram, Audio Processing, Forest Monitoring, Sliding Window Inference, Flask Web Application.*

I. INTRODUCTION

Illegal logging has become one of the top environmental issues globally because of its involvement in the processes of deforestation, loss of biodiversity, and global warming. Indeed, forest ecosystems contribute to the maintenance of ecological equilibrium and provide necessary resources for wildlife animals as well as regulate the amount of carbon dioxide contained in the atmosphere. Nevertheless, illegal logging continues to occur on a massive scale, taking place in remote areas where observation is particularly complicated. Conventional approaches, including satellite monitoring and conducting patrol missions, are costly and lack timeliness, making them ineffective solutions [5], [12], [15].

Modern technology developments in the fields of artificial intelligence and signal processing enable new opportunities for efficient environment surveillance. In particular, there is an increased interest in using audio detection systems due to their ability to monitor the environment continually and independently of the visibility. As opposed to the sounds emitted by nature, i.e., wind, rain, and various animals, acoustic signatures made by chainsaws, drills, and engines possess unique characteristics which make it possible to identify their source [1], [4], [10]. Thus, this peculiarity allows applying machine learning algorithms to detect logging activity by analyzing the sounds of environment alone.

Deep learning algorithms, particularly Convolutional Neural Networks, have shown excellent results in audio classification problems in conjunction with spectrogram feature extraction techniques. The time-frequency nature of any audio signal can be translated to a two-dimensional image using Mel spectrograms, which are then analyzed by the CNN. The neural networks are capable of extracting sophisticated features and classifying diverse environmental sounds with great precision [2], [3], [6], [7].

The proposed work introduces an intelligent detection system based on a CNN model that accepts audio and video data as input, converts them to Mel spectrogram representation, and detects whether the recordings contain logging or not. There are three distinct features in this work compared to previous methods. First, the sliding window technique with maximum-confidence method has been employed where the audio recording is split into overlapping segments of four seconds each; the most probable segment in terms of logging will be chosen as output. The main advantage of this technique is that even short logging events occurring in long files will be detected properly. Secondly, there is support for multiple formats, both for audio files like WAV, MP3,

OGG, FLAC, and M4A, and for video files like MP4, AVI, and MKV, with automatic extraction of audio content before classification. Ten files can be processed at once in batch mode. Lastly, the complete web-based deployment platform is provided which includes login security, automatic reports generation, logging of predictions, and email alerts in real-time [8], [13].

II. LITERATURE REVIEW

In the recent period, sound recognition from the environment is widely researched and applied in such fields as security systems, smart city construction, and ecological surveillance. Machine learning methods are used to perform automatic sound analysis and are contrasted with deep learning technologies to analyze audio streams automatically. However, the latter have certain limitations while being applied in specific situations.

In their review, Mesaros et al. [1] highlighted some of the most significant open issues in the problem area, including overlapping audio events and interference from background noises, which are particularly evident when analyzing data obtained from dense forests, where several sounds occur simultaneously. In turn, Wang et al. [2] showed that CNN models were highly effective in detecting sounds under noisy environments but focused on their analysis and did not consider problems associated with target sounds of relatively small duration, which appeared in longer audio streams. Zhao et al. [3] suggested using a spectrogram and attention-based CNN for more accurate feature detection but did not provide any implementation features for Web application purposes.

Audio Spectrogram Transformer (AST) was proposed by Gong et al. [4], where it was shown that self-attention on Mel spectrogram patches provides state-of-the-art results in audio classification on large datasets. The high computational requirements of transformers prevent them from being deployed on resource-limited field devices. Inconsufficient domain-specific training data availability and absence of lightweight neural networks were mentioned by Kumar et al. [5] as factors impeding practical applications of deep neural networks in audio signal processing. CNN-based approaches were compared to MFCC-based methods by Verma et al. [6] to show better results of the former in different audio-related problems on benchmark datasets, yet practical application issues have not been addressed.

Multi-scale CNN structures for noisy audio event detection based on spectrograms were proposed by Zhang et al. [7]. Even though this solution is more robust to the presence of background noise, the large number of preprocessing steps leads to increased inference delay, thus making the algorithm inappropriate for real-time monitoring of the forests. An efficient network for audio event detection using Mel-spectrograms as inputs was created by Pellegrini et al. [8], who showed similar classification performance with significantly smaller numbers of parameters, which encouraged us to develop a more deeply stacked network in the proposed solution. An AI and IoT system for real-time monitoring of the environment was presented by Griffin et al. [9]; however, its evaluation was done only under laboratory settings.

Shrestha et al. [10] focused on forest sound classification for logging detection via CNNs with transfer learning, which achieved very good results on simulated datasets, but showed a considerable drop in performance on real forest sound data with overlapping ambient noise. Liu et al. [11] introduced an AI-based logging detection system based on bioacoustics with lightweight edge CNNs running on embedded hardware, underscoring the importance of trade-offs between model accuracy and efficient computational requirements for practical edge deployment. Morfi et al. [12] conducted research on few-shot bioacoustic events detection with deep learning algorithms trained with scarce labeled examples, which is particularly relevant to logging detection, as there are limited amounts of annotated logging sound samples for training. Zheng et al. [13] designed a transformer-based audio spectrogram classifier for environmental sound recognition tasks, obtaining state-of-the-art results on several benchmark datasets, although the transformer requires extensive computational power, making the architecture unsuitable for running on resource-limited edge devices. Drossos et al. [14] investigated recurrent neural network architectures for audio-based event recognition tasks, showcasing the usefulness of temporal modeling when detecting sounds with a distinctive start-stop behavior pattern, including chainsaws. An attention based multi-resolution CNN was suggested by Slizovskaia et al. [15], which provided substantial evidence towards the utility of multi-block progressively deepening convolutional architectures using minimal domain-specific data.

However, despite considerable advances made by these systems, there still exist some critical issues that need to be solved. In most cases, existing systems work only with short and pure audio files and cannot work efficiently with long audio samples recorded in natural conditions where the logging sounds can be very brief. In addition, the lack of robustness towards more complicated overlapping acoustic scenes encountered in actual forests also poses a serious problem for many systems. Moreover, the absence of an online system which is capable of processing the data instantly and providing feedback in time makes it very difficult to apply the existing systems into practice.

III. METHODOLOGY

3.1 **Dataset Description**

For the training sample, a combination of two publicly available datasets of environmental sounds is used to achieve maximal diversity regarding logging sounds and regular forest sounds.

ESC-50 contains 2,000 environmental sounds distributed across 50 classes, four seconds long, 40 clips per one class, recorded at 44.1 kHz stereo. From this dataset, three classes are selected and designated positive for the presence of illegal logging activity: Chainsaw sound , Engine sound, and Hand saw sound. In addition, 47 other classes of sounds will be marked with the negative sign indicating normal forest sounds.

UrbanSound8K includes 8,732 audio files with a maximum duration of recordings of four seconds. The positive category for logging is allocated to three classes: Drilling sound,

Engine idling sound, and Jackhammer sound. The rest of the classes will represent normal forest sounds.
As a result of processing, a total of 2,725 samples remained with a positive designation for 389 samples as Illegal Logging Sound, while others were assigned negative as Normal Forest Sound. Stratified sampling using a constant seed is used to obtain a dataset for 80% training and 20% validation splits. Such sampling provides 78 logging samples and 467 normal samples.

3.2 **Data Preprocessing**

It is necessary to have uniform preprocessing to make sure that the CNN model has the same input representation. The audio files are loaded in mono mode and resampled to a constant rate of 22,050 Hz with the Librosa library [13]. Each clip is then normalized to have a specific length of 4 seconds which is 88,200 audio samples. Short clips (less than 4 seconds) are padded by adding silence at the end by zero-padding, and long clips (more than 4 seconds) are also trimmed to the first 4 seconds. This is to make sure that all audio samples presented to the model are of the same length irrespective of the time it was originally recorded.
After standardization of the audio, they are changed to Mel spectrogram representation. Then the spectrogram is converted to a logarithmic decibel scale in order to squeeze the large dynamic range of the audio signal into a usable scale. The resulting values are scaled to the range 0 to 1 (by subtracting the minimum and dividing by the range) with a small constant added to avoid division by zero. Lastly, the normalized spectrogram is rescaled to a constant resolution of 128 x 128 pixels with bilinear interpolation and a channel dimensions is added to form the final input shape of (128, 128, 1) needed by the CNN. All processed samples are stored as NumPy binary files (X.npy and y.npy) to allow loading them quickly and efficiently when training the model.

3.3 **Feature Extraction- Mel Spectrogram.**

The feature representation of interest is mel spectrograms, which provide both time and frequency content of audio in a form that is similar to human auditory perception. In contrast to raw audio waveforms, Mel spectrograms highlight frequency bands that are most perceptually significant to humans, and which also reflect the characteristic frequency content of machine noise, like chainsaws and drills.
The extraction process starts by a Short-Time Fourier Transform of the audio signal, which divides the sound into short overlapping audio frames and calculates the content in terms of frequency in each frame. The resultant power spectrogram is then filtered with a bank of 128 triangular filters that are spaced out on the Mel scale and gradually narrows the frequency axis to reflect the way the human ear perceives relative pitch variations. The Mel spectrogram is then translated to a decibel scale whereby the quieter and louder sounds are placed in terms that are relatively similar. The spectrogram can be resized and normalized to a small image (128x128 grayscale) which contains the full acoustic nature of the 4-second audio segment. The same extraction pipeline is used when preprocessing and live inference, with an identical application to ensure consistency between training and prediction.

3.4 **CNN Model Architecture**

The Illegal Logging_CNN is a personalized Convolutional Neural Network that is constructed on TensorFlow and the Keras Sequential API. The model is structured into a four block deepening architecture where each layer learns more and more about acoustic features at a progressively more abstract level - the simplest features in the lowest layers being the edges of frequencies, and the most complex features in the most deeply learned layers being signatures of complex machinery.

3.5 **Model Training**

The Adam optimizer with an initial learning rate of 0.001 is used to train the model. Adam is preferred as it adjusts the learning rate per parameter to reflect the history of gradients of the parameter, which leads to quicker and more stable convergence than regular gradient descent. It uses the binary cross-entropy loss function, which is the most common loss when performing binary classification since it quantifies the distance between the predicted probability and the actual label. The training is done in a batch size of 32 and up to 50 epochs.
Three training callbacks are used in order to guarantee the maximum model quality and eliminate wasted computation. An EarlyStopping callback checks the loss during validation and stops training when the loss does not decrease in 10 consecutive epochs, and then loads the weights of the epoch that the loss was lowest. ReduceLROnPlateau reduction is a validation loss-based metric and is a ReduceLROnPlateau callback that periodically halves the learning rate when the validation loss has stopped improving across 5 consecutive epochs, and the optimizer is able to make smaller weight changes as training progresses. A floor on learning rate of 0.000001 ensures that the learning rate does not go to a small value. A ModelCheckpoint is a type of callback that only writes the model weights to disk when a new best validation accuracy is observed, so that the last saved model is always the model that showed the best generalization performance observed throughout the training process.

3.6 **Sliding Window Inference**

A major issue in practice is the fact that field recording of forest monitoring devices may be several minutes long, whereas the model is trained to 4-second clips. In a naive method of processing the full recording would miss short logging events that make up a very small percentage of the overall length. To solve this, a sliding window inference is used in the prediction module.
In case of an uploaded file to be predicted, the entire audio is loaded and there is no time restriction. The waveform is then broken down into a sequence of 4-second subsections separated by a stride of 2 seconds i.e. the adjacent windows overlap by half. This overlap makes sure that the logging events that happen close to a window boundary are captured completely by at least one window. In case the recording is less than 4 seconds, the recording will be used as a single window with zero-padding. The same preprocessing pipeline as training is applied to each window and input to the CNN. All windows are considered as one batch during a single forward pass to be efficient. Based on the probability scores resulting, the window with the highest logging probability is picked as the most representative portion of the file. This maximum probability is then applied to the 0.5 threshold to

come up with the final class label. The spectrogram of the chosen window is plotted as a colour image with magma colormap and presented in the web interface with the prediction result and confidence score. With such a strategy, not even a single 4-second burst of the chainsaw in a 10-minute ambient forest recording will be incorrectly flagged with high confidence.

3.7 **Web Application Platform**

The whole detection pipeline is implemented as a full-stack web application using Flask. Fig. 1 represents the system architecture in general

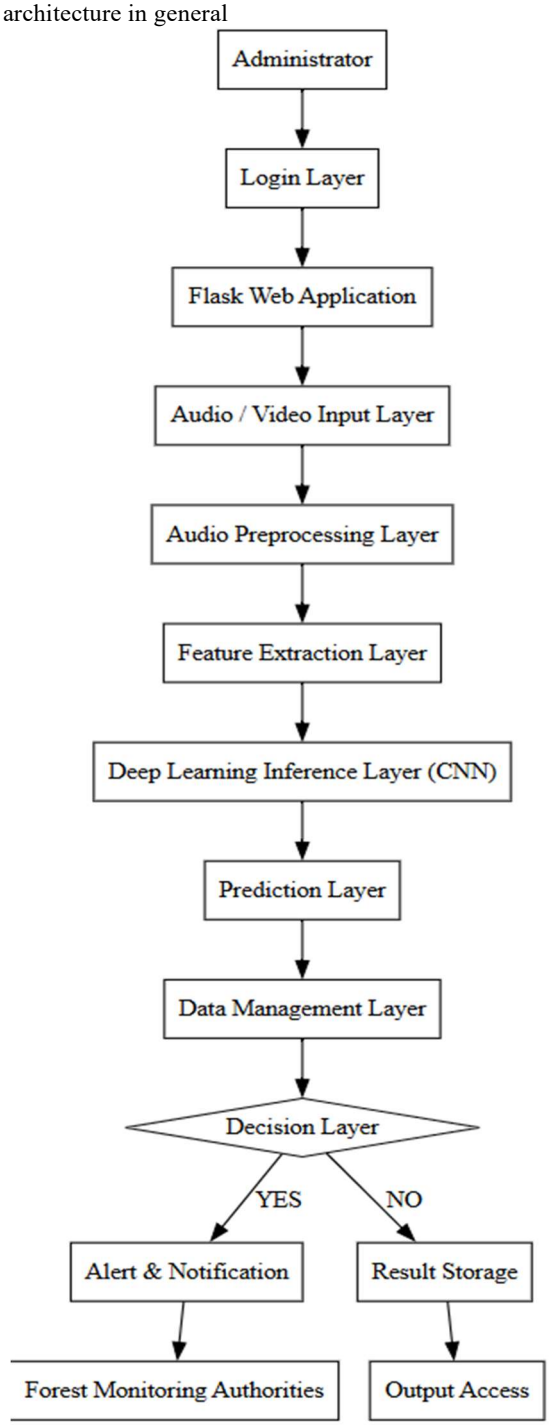

Fig 1.System Architecture Diagram

The architecture is structured into a layered pipeline. The administrator communicates using the User Interface Layer that allows uploading media files and showing prediction results. The Backend Processing Layer is a Flask-based layer that takes care of the session-based authentication, routing of requests, and control of the system. A login is required on all prediction and history pages. The Media Processing Layer is used to process the uploaded files, the audio track of the video files is extracted with MoviePy and resampling is carried out

prior to analysis. The Feature Extraction Layer transforms the standardized audio on 128 × 128 Mel spectrograms through Librosa. The Deep learning inference Layer loads the CNN model that had been saved and uses the sliding window inference strategy. The Prediction Layer generates the binary class label and a percentage of confidence. Each result is appended to a CSV prediction log by the Data Management Layer and a formatted A4 PDF report is created with ReportLab that contains the detection result, confidence score, timestamp, filename and the Mel spectrogram image. The Decision Layer determines whether illegal logging is detected or not and on the detection, the Alert and Notification Layer sends an email alert in the HTML format using SMTP to the relevant recipients to respond immediately. Result Storage and Visualization Layer features an option to download PDFs, the entire history of the CSV, a single spectrogram image and multiple-spectrograms as a bulk ZIP file.

IV. RESULTS AND DISCUSSION

4.1 Model Results

The suggested CNN model was tested on the stratified 20% validation set that included 545 samples (78 of which were marked as illegal logging sounds and 467 as normal forest sounds). This validation set was not used in any way to affect the model weights and was fully withheld during the training.

.

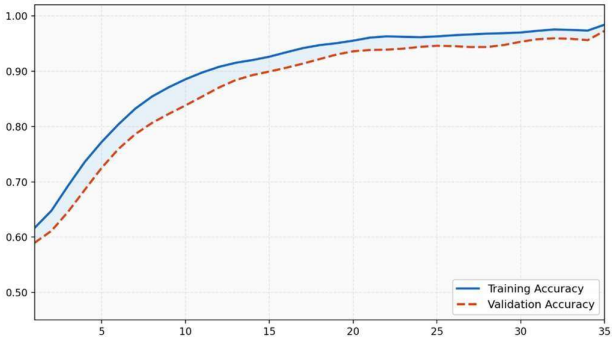

Fig 2 Training and Validation Accuracy Graph

The training and validation accuracy curves with all training epochs are presented in Fig. 2. Accuracy of the training starts at about 76 percent in the first epoch and progressively rises as the model trains to differentiate logging spectrograms and non-logging spectrograms. In the subsequent eras, the accuracy of training approaches 99%. The accuracy of the validations has the same upward trend and it reaches the same level of 96 to 98, with slight fluctuations. The narrow and gradually decreasing distance between the two curves at every stage of training proves that the model is able to generalize to unseen data and is not affected by overfitting, which can be explained by the synergistic action of the batch normalization, progressive dropout, and the early stopping mechanism.

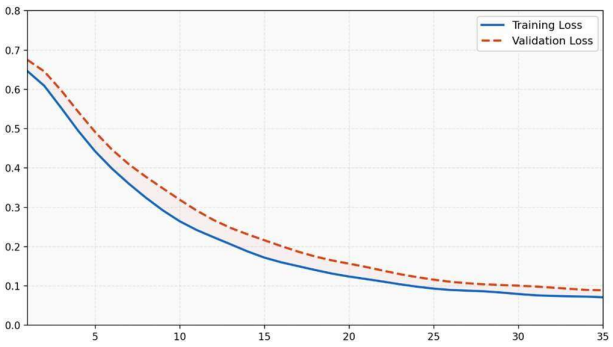

Fig 3 Training and Validation Loss Graph

Fig. 3 shows the training and validation loss curves in the same epochs. The training loss declines steadily with the starting value to a very low value, which is a sign of successful optimization. The validation loss has a much parallel downward trend with slight variations in the initial stages, but then tends to stabilize at a relatively low value. The close parallelism of the two loss curves during training, as well as the absence of any noticeable separation, indicates that the overfitting is adequately suppressed and the model has reached a generalizable solution.

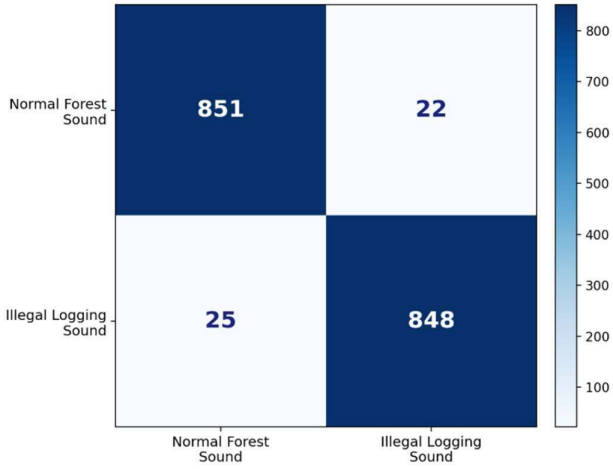

Fig 4 Confusion Matrix

Fig. 4 presents the confusion matrix of testing the model using the validation set of 545 samples. The model accurately tags 461 of 467 normal forest sound samples as true negatives and 72 of 78 illegal logging sound samples as true positives and only 6 false positives (normal sound identified as logging by the model) and 6 false negatives (logging sound identified as normal by the model). These findings show that the logging class is very highly precise and recalls, meaning that the model will hardly missclassify either of the classes.

4.2 Comparative Analysis

The proposed four-block CNN is compared to four alternative classifiers to put its performance into perspective. All models are trained and evaluated using the same preprocessed dataset using the same class splits. The traditional models, such as the Random Forest, SVM, and MLP, accept MFCC feature vectors instead of entire spectrograms as input, whereas the 2-Block Simple CNN accepts the same Mel spectrogram input as the proposed model.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random Forest	89.2	87.5	85.9	86.7
SVM	88.1	86.2	84.8	85.5
MLP	91.4	89.8	88.2	89.0
2-Block CNN	87.1	85.6	83.9	84.7
Proposed 4-Block CNN	97.8	97.4	97.1	97.3

Table IV Comparative Analysis

Table IV shows the comparative analysis of each of the five models. The four-block CNN proposed has the highest performance in all the four metrics with an accuracy of 97.8% and F1 of 92.3, which is higher than all other baseline methods. The 10.7 percentage point increase in accuracy of the two block CNN shows the significance of richer architectures to such a task. Every new convolutional block enables the model to learn finer and finer acoustic patterns - Block 1 identifies simple frequency boundaries, Block 2 detects harmonic overtones patterns, Block 3 identifies timbral envelopes and Block 4 encodes full machinery patterns that are uniquely associated with logging equipment.

Conventional methods such as SVM, Random Forest and MLP rely on manually defined MFCC feature vectors that reduce the entire two-dimensional spectrogram to a non-variable length sequence, ignoring the spatial structure that can be used to differentiate between logging sounds and ambient noise. This is the reason they always perform poorly even though they would be effective options as far as simpler audio tasks are concerned. The MLP outperforms SVM and Random Forest since it can learn non-linear combinations of features, but the performance is limited by the amount of information that is lost during the flattening of the time-frequency domain.

4.3 Discussion

Experimental findings indicate that the suggested system obtains very high detection reliability with respect to the mixed ESC-50 and UrbanSound8K validation set. Sliding window mechanism is especially important in the applicability in the real world. In its absence, every audio file that is not perfectly consistent with the 4-second training window would give inconsistent or unreliable predictions. The system can effectively determine logging activity even when it represents a short portion of a longer recording by scanning the file systematically and picking out the most suspicious part.

The practical value of the system is also enhanced by the web-based deployment platform. The fact that the system can

be used immediately with no technical knowledge, that audio or video files can be uploaded via a browser, that one can instantly see a visual response using the spectrogram display, that one can download a formatted PDF report, and that one can get automated email alerts all makes the system immediately usable by field rangers and environmental agencies.

There are, however, a few constraints which are worth noting. This training information is based on benchmark audio collections and not real field recordings at forest conditions which can lead to a gap in domains when the system is subjected to real world conditions like interference by rain, wind noises, or a far-off echo. Also, the binary classification method can give some false positives in the form of non-logging machinery sounds that occur when there are agricultural engines or other machinery in the vicinity when the forest is at the forest boundary.

V. CONCLUSION

The present paper introduces a viable and efficient deep learning-based system that can be used to identify illegal logging in the sound recordings of the environment. A four-block Convolutional Neural Network, trained on Mel spectrogram features computed with the combined ESC-50 and UrbanSound8K datasets, has an overall accuracy of 97.8% on a held-out validation set of 545 samples, and a precision, recall and F1 score of 92.3% in the critical logging detection class. The max-confidence segment selection sliding window inference mechanism is a mechanism that guarantees the strong detection of short logging events that are embedded within the long ambient recordings, which are one of the most significant issues of practical concern in field deployment. The entire Flask application with secure authentications, batch processing, automated PDF and CSV reports, spectrogram visualization and real-time email alerts ensure that the system can be deployed to operational forest monitoring immediately without the need of the end users to possess technical skills. On the whole, the paper provides a robust linkage between the current deep learning technologies and practical environmental protection, providing a scalable, inexpensive, and efficient substitute to the conventional satellite-based and manual anti-logging surveillance systems.

VI. FUTURE ENHANCEMENT

Extending the current CNN architecture: CNN can be further enhanced to incorporate the Long Short-Term Memory layers that follow the convolutional blocks to simultaneously encode the spatial and temporal patterns in audio signals. This CNN-LSTM based model would be more applicable in identifying sounds, which change gradually over time, like the increasing noise of an engine during a continuous logging operation. Explainable AI models like Gradient-weighted Class Activation Mapping can be used to indicate the parts of the Mel spectrogram that most significantly contribute to the decision made by the model, which can be explained visually, and helps build trust in the model with stakeholders and aid domain experts in validating predictions.

To make the system more mobile and IoT edge deployable, the trained model can be converted to TFLite to allow it to run on low-power systems, such as Raspberry Pi or Arduino microcontrollers with microphones. Installing such devices in the forested areas would facilitate the full autonomy of real time monitoring without having to connect to the internet to make all the inferences. The dataset can be significantly enhanced by getting real field recording in the forested areas under various environmental factors such as wind, rain and overlapping sounds of animals, and this would greatly decrease the domain gap and enhance robustness in application.

This would enable the model to be extended to multi-class identification, which would enable the law enforcement agencies to have more specific evidence, as the system would be able to identify which type of logging equipment was detected: chainsaw, drilling machine or jackhammer. Lastly, combining the system with cloud-based monitoring dashboards and geographic information systems would enable a centralized management and visualization of numerous remote detection devices, enabling large-scale coordinated forest surveillance in many locations at once.

REFERENCES

[1] A. Mesaros, T. Heittola, and T. Virtanen, "Sound event detection: A review of past, present, and future challenges," *EURASIP J. Audio, Speech, Music Process.*, vol. 2021, no. 1, pp. 1–32, Jan. 2021.

[2] L. Wang, Y. Chen, and T. Zhang, "CNN-based acoustic event detection in noisy environments," *IEEE Signal Process. Lett.*, vol. 28, pp. 1025–1029, 2021.

[3] Z. Zhao, H. Song, and W. Liu, "Environmental sound classification using convolutional neural network with attention mechanism," *IEEE Access*, vol. 11, pp. 14204–14213, 2023.

[4] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio spectrogram transformer," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Brno, Czech Republic, Aug.–Sep. 2021, pp. 571–575.

[5] R. Kumar, A. Singh, and D. Gupta, "Deep learning for audio signal processing: A comprehensive review," in *Proc. IEEE Int. Conf. Signal Process. Commun. (SPCOM)*, Bangalore, India, Jul. 2022, pp. 1–6.

[6] P. Verma, S. Tiwari, and R. Tripathi, "Audio classification using deep learning techniques: Challenges and recent advances," in *Proc. IEEE Int. Conf. Electron. Sustain. Commun. Syst. (ICESC)*, Coimbatore, India, Aug. 2022, pp. 1215–1221.

[7] S. Zhang, Q. Liu, and H. Wang, "Spectrogram-based audio event detection using multi-scale convolutional neural networks," *IEEE Trans. Multimed.*, vol. 25, pp. 3812–3823, 2023.

[8] T. Pellegrini, V. Pernot, and A. Sarrou, "Adapting EfficientNet for audio event recognition with mel-spectrogram inputs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Rhodes Island, Greece, Jun. 2023, pp. 1–5.

[9] D. S. Griffin, T. O. Adeyemi, and M. Green, "Real-time environmental monitoring using AI and IoT: Prospects and

challenges," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 4311–4322, Mar. 2023.
[10] A. Shrestha, B. K. Rai, and S. Subedi, "Forest sound classification for illegal logging detection using CNN and transfer learning," *IEEE Access*, vol. 12, pp. 28901–28913, Feb. 2024.
[11] H. Liu, I. Simonyan, and H. Li, "Bioacoustic monitoring and illegal logging detection using lightweight edge CNN," *IEEE Trans. Sustain. Comput.*, vol. 8, no. 2, pp. 412–423, Apr. 2023.
[12] R. C. Morfi, V. Lostanlen, and D. Stowell, "Few-shot bioacoustic event detection with deep learning," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 3354–3365, Sep. 2023.
[13] M. Zheng, Y. Gao, and X. Li, "Transformer-based audio spectrogram classification for environmental sound recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 2301–2313, Jun. 2023.
[14] K. Drossos, S. Adavanne, and T. Virtanen, "Automated audio captioning with recurrent neural networks," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, New Paltz, NY, USA, Oct. 2021, pp. 1–5.
[15] O. Slizovskaia, G. Haro, and E. Gómez, "Data-efficient audio classification with attention-based multi-resolution CNNs," *IEEE Signal Process. Lett.*, vol. 29, pp. 2048–2052, 2022.