2024 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) | 979-8-3503-9172-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ANTS63515.2024.10898423

Real-Time Car Honk Detection for On-the-Go Application using Lightweight Deep Learning Model

Biswajit Maity*, Akash Chokhani†, Subrata Nandi‡, Sanghita Bhattacharjee§
*Institute of Engineering & Management, India
†‡§National Institute of Technology Durgapur, India,
Email:*biswajit.maity1@gmail.com, †akashchokhani99@gmail.com, ‡subrata.nandi@gmail.com,
§sbhattacharjee.cse@nitdgp.ac.in

***Abstract*—Noise pollution from vehicular honking is a pressing issue in urban environments, necessitating the development of efficient honk detection systems. While deep learning models have shown promise in detecting car honks, their high computational demands limit their deployment on mobile devices. This research introduces a novel approach to car honk detection, introducing a lightweight deep learning models optimized for mobile devices. We developed a mobile application that detects vehicular honks in real-time, offering a practical solution to monitor noise pollution and enhance pedestrian safety. Our proposed model achieved a 97.65% accuracy with low average inference time and latency of 63.8 ms and 288.79 ms, respectively. This application can alert pedestrians wearing earphones or headphones about nearby honks, helping prevent accidents. Our work addresses key challenges in resource optimization, real-time processing, and latency reduction, providing a significant contribution to urban safety and environmental monitoring.**

***Index Terms*—Noise pollution, Vehicular honking, Real-time detection, Lightweight model, Resource optimization**

I. Introduction

The pervasive issue of noise pollution due to vehicular honking [1] in road traffic areas has prompted significant research efforts. Vehicular honks, while a primary source of outdoor noise pollution, play a crucial role in preventing accidents and ensuring pedestrian safety. Consequently, accurately identifying these honks is vital for both noise management and public safety.

Existing research has predominantly focused on processing audio signals [2], [3] or developing deep learning models [4] for honk detection. While these studies have demonstrated the feasibility of honk identification, the models often require high computational resources and sophisticated hardware, making them impractical for deployment on low-end mobile devices or in on-the-move scenarios. This gap highlights a critical challenge: achieving real-time detection of vehicular honks on mobile platforms, which has not yet been adequately addressed.

If we can successfully implement real-time honk detection on mobile devices, it would enable the development of smart traffic management systems, noise maps, and several micro-services. For example, an application could assist hearing-impaired individuals by alerting them to nearby vehicular honks. Additionally, since many people use headphones while moving, an app that generates alerts upon detecting a honk could significantly enhance pedestrian safety. Our research proposes a novel approach using a lightweight deep learning model optimized for mobile devices to detect in real-time, thereby enhancing safety and smart traffic management in urban environments.

Motivation: The motivation for this work arises from the inadequacy of existing literature in addressing the absence of on-move honk detection solutions. While previous studies have delved into honk identification using techniques like FFT [2], MFCC [3], and spectrogram analysis, they predominantly concentrate on stationary settings and lack efficient solutions for honk detection in dynamic environments. Additionally, while some researchers have employed deep learning models [4], [5] for honk identification and system automation, the literature lacks investigations into on-move honk detection using lightweight deep learning models with low latency. This gap underscores the need for our research to develop a mobile application capable of accurately detecting vehicular honks in real-time, even amidst ambient noise, thereby contributing to pedestrian safety and urban environment monitoring.

Issues and Challenges: Developing a car honk detection mobile application presents several challenges that must be addressed to ensure its effectiveness and usability. *Firstly*, optimizing computational resources and achieving real-time processing is crucial, considering the limited computational power and memory available on mobile devices. This requires careful optimization of the model architecture and implementation of efficient algorithms to minimize resource consumption while maintaining real-time performance. *Secondly*, the absence of native audio processing functions in mobile operating systems complicates the implementation of advanced audio processing techniques for honk detection. Developers may need to rely on custom solutions or third-party libraries, introducing additional complexity and potential overhead. *Lastly*, achieving low response latency is essential for providing timely alerts to pedestrians, especially in fast-paced urban environments where split-second decisions can make a difference. By addressing these challenges through efficient algorithm design, optimization, and integration, developers can create a car honk detection application that enhances pedestrian safety and contributes to a safer urban environment.

979-8-3503-9172-5/24/$31.00 ©2024 IEEE

Contribution: The contribution of this work encompasses several innovative aspects aimed at enhancing the efficiency and functionality of the car honk detection application:

- Initially, our dataset consisted of a different types of honk samples collected from road traffic scenarios. Furthermore, to enhance the volume and diversity of the dataset, we implemented a data augmentation technique that involved incorporating various ambient noises into the honk samples. This augmentation process involved merging 16,000 honk samples with nine different ambient noises, resulting in the generation of 30,720 augmented samples. By integrating diverse urban noise conditions into the dataset, our objective is to train the model to recognize and differentiate honks amidst a variety of real-world environmental sounds.
- We developed a lightweight deep learning model designed to run on resource-constrained devices, which is the primary motivation for this research. Our proposed model achieved 97.65% accuracy with low inference time and latency, specifically 63.8 ms and 288.79 ms, respectively.
- We developed an Android application to enhance pedestrian safety by detecting car honks in real-time for users wearing earphones or headphones. The app loads the model upon startup, runs in the background, and uses the device's microphone to monitor for honks. When a honk is detected, it alerts the user with a message in their headphones, saying "Alert", helping to prevent accidents.

Rest of the paper is organized as follows: Related works are discussed in Section II. In Section III, the framework of our proposed word is discussed. The details methodology is illustrated in Section IV. Result and its analysis along with micro-service is depicted in Section V. Finally, a conclusion is provided in Section VI.

II. Related Work

Noise pollution, characterized by excessive noise disturbances, significantly impacts our ecosystem [6]. In recent decades, researchers have meticulously explored various techniques to characterize and quantify noise pollution levels in road traffic environments. Vehicular honking, a major contributor to outdoor noise pollution, predominantly occurs in traffic-heavy areas. Addressing this issue by identifying and classifying different types of vehicular honks is crucial for comprehensively assessing noise pollution levels and developing a range of services. Researchers have approached this challenge through two primary methods:

A. Identification of Vehicular Honks from Raw Audio Samples

Advanced signal processing techniques such as Fast Fourier Transform (FFT), Mel-frequency cepstral coefficients (MFCC), and spectrogram analysis are employed to identify vehicular honks directly from audio samples. In [2], FFT is used for honk identification, and band-pass filtering is implemented to eliminate various noises from the original audio data. However, this approach had limitations as the selected threshold values for the band-pass filters did not consistently ensure accurate honk or noise detection. Additionally, there is no automated system devised for honk identification directly from raw audio files. In [7], an FFT-based method is used to detect honks, specifically to assist hearing-impaired individuals while driving. Author also developed a smartphone-based system for honk detection and alerting the hearing-impaired [8], while [9] created an embedded system to identify emergency honks. Additionally, [3] explored MFCC-based techniques for honk detection.

B. Honk Identification through Environmental Sound Classification

Many studies have focused on identifying vehicular honking by categorizing environmental sounds, with car honking as one of the classes [5], [10]. These studies primarily relied on datasets such as ESC-10, ESC-50, or UrbanSound8K, collected in controlled environments without ambient noise. Various CNN-based models, including SBCNN [11], Dilated CNN [12], CNN 2020 [13], TFCNN [14], and others, are developed to improve accuracy. The highest accuracy reported is 96.28%, as seen in [15] for the ESC-10 dataset. It is important to note that the ESC-10 dataset does not include the honk class. Researchers have also used the DCASE-2017 ASC dataset to identify the honk class [5].

The aforementioned studies demonstrate ongoing efforts in honk identification and environmental sound classification. However, fine-grained vehicular honk identification in real-time on low-end systems remains an unexplored area. Therefore, in this research, we propose a novel lightweight deep learning model that achieves high accuracy, low latency, and rapid inference rates, making it suitable for resource-constrained systems and real-time honk detection. The insights gained from these honk sources inform not only traffic dynamics and road conditions but also potential pollution levels. These insights are essential for developing micro-services that enhance urban living, such as an app designed to safeguard pedestrians who may be using headphones while crossing roads and unable to detect vehicular honks.

III. Framework

The framework consists of three main components. First, in the Dataset Preparation phase, audio data is collected from various sources and subsequently processed and augmented for training. In the Model Training and Testing phase, the augmented audio data is transformed into spectrogram images, which are utilized to train the model. Post-training, the model undergoes rigorous testing and analysis to evaluate its performance. Finally, in the App Development and Integration phase, the trained model is incorporated into a mobile application. The app is then thoroughly tested to ensure functionality, and once it passes all evaluations, it is finalized and prepared for user deployment. The details structure of the framework is depicted in Fig. 1.

IV. Methodology

A. Dataset Description

The development of a robust car honk detection system hinges on the availability of a diverse and comprehensive

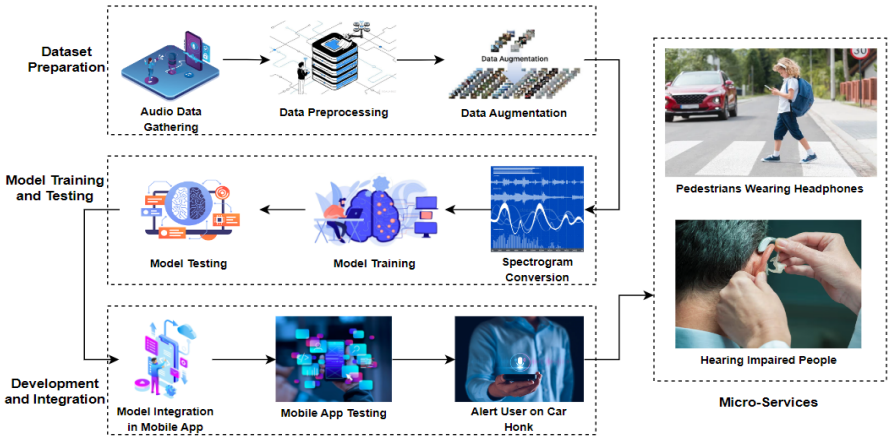


Figure 1: Workflow of our proposed system

dataset that can adequately represent the complex acoustic environments encountered in urban settings. However, a notable challenge we encountered in this endeavor is the scarcity of suitable data readily available for training a dedicated honk detection model. Existing datasets like YouTube-8M and AudioSet do not meet these criteria. To overcome this obstacle, we adopted a specialized approach to data collection, prioritizing the acquisition of a wide variety of honk sounds over the sheer volume or duration of these sounds. By focusing on diversity rather than quantity, we aimed to ensure that our dataset encompassed the full spectrum of honk variations encountered in real-world scenarios, thereby enhancing the effectiveness and generalizability of our honk detection system.

The initial phase of our research involved collecting a diverse set of raw audio samples comprising both honk and non-honk instances from various environmental settings. The non-honk samples encompassed a wide array of urban sounds, including conversations, loudspeaker announcements, and engine noises from different types of vehicles. Concurrently, honk recordings are obtained from multiple vehicle types and settings to ensure variability in the acoustic characteristics of the honks.

Data Preparation & Augmentation: We follow different steps to prepare our dataset: *Step1:* we randomly select raw audio files for honks and background noises from their respective datasets. This randomness helps expose the model to a wide range of audio features. *Step2:* we extract a clip from the selected honk audio file, with the duration randomly set between 0.25 to 1 second. This variability in duration mimics real-world scenarios where honks can vary in length. *Step3:* We also randomly choose the position within the audio file from which to extract the honk clip. This step is important for capturing different parts of a honk, such as the onset, peak, or tail, providing diverse examples for the model to learn from. *Step4:* At the same time, we extract a 1-second clip of background noise from a randomly selected position within its audio file. This background noise serves as the context in which the honk must be detected, replicating the noisy environments where the application will be used. *Step5:* We adjust the loudness of the honk clip by multiplying its amplitude by a random factor ranging from 0.75 to 1. This step simulates different distances and orientations from which a honk might be heard. *Step6:* Finally, we superimpose the prepared honk clip onto the 1-second background noise clip at a random position. This combination creates a realistic scenario where the honk is embedded within ambient sounds, challenging the model to identify the honk amidst competing noises. The entire process is depicted in Fig. 2.

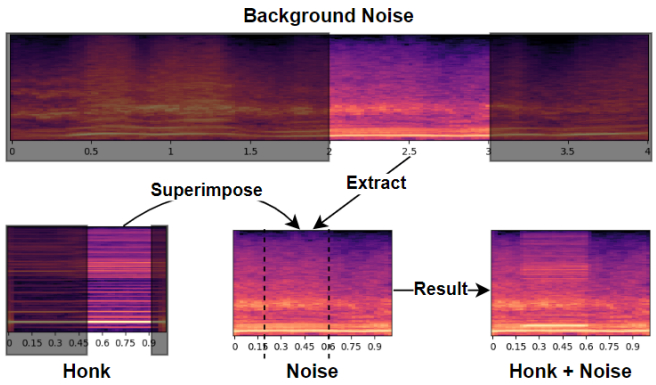


Figure 2: Illustration of Dataset preparation and augmentation technique

B. Model Design and Development

We aim to develop a lightweight and efficient model for mobile applications, prioritizing minimal resource consumption. The model architecture begins with the input of a 1-second audio clip sampled at 22,050 Hz, which is augmented to reflect real-world conditions. The initial processing stage involves a custom Mel Spectrogram Image (MSI) layer that converts the raw audio waveform into a log-mel spectrogram image using TensorFlow functions, resulting in a 128x128x1 image that effectively captures the frequency content of the sound over time. Due to the unavailability of any popular library for spectrogram conversion in Flutter, we introduced a custom layer called the MSI layer in our model that converts raw audio into a log-mel-spectrogram image. This layer is

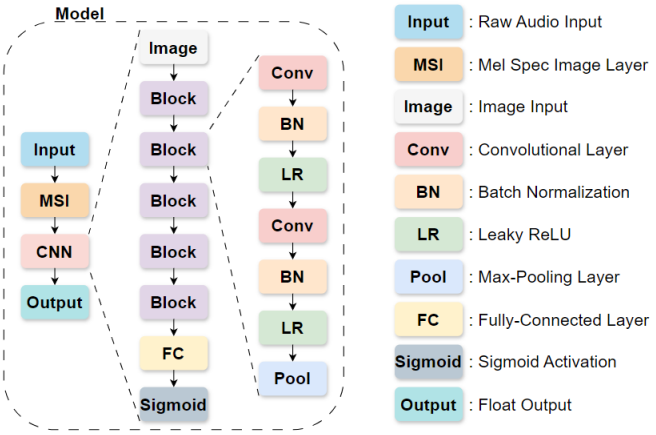


Figure 3: Illustration of model architecture

fully compatible with TensorFlow Lite models, as it relies solely on TensorFlow functions. The introduction of this MSI layer removes the need for a separate spectrogram conversion library in mobile app development frameworks. Additionally, this layer's versatility allows it to be paired with any other pre-trained models for audio detection or classification that take images as input.

Following the MSI layer, the spectrogram image is processed by a lightweight Convolutional Neural Network (CNN) designed to detect and interpret various features within the log-mel spectrogram. Despite its compact size of only 1,597 parameters, the CNN performs this task efficiently without overburdening the mobile device's computational resources. This ensures the model's suitability for real-time processing on mobile platforms. The output from the CNN is a float ranging from 0 to 1, representing the probability or confidence level of a honk being present in the audio. A value closer to 0 indicates a low likelihood of a honk, while a value closer to 1 suggests a high probability. This probabilistic output allows for flexible decision-making processes, where thresholds can be adjusted based on specific application needs or environmental conditions.

The architecture is designed with flexibility in mind, allowing the replacement of the CNN with any other model built using Keras, ensuring that the application can be updated or refined with more advanced models in future development cycles. The model architecture is visually represented in Fig. 3, providing a clear diagrammatic flow from the MSI layer through the CNN to the output. To integrate the model into the Flutter-based mobile application, the TensorFlow model is converted into TensorFlow Lite format. This conversion optimizes the model for mobile environments, significantly reducing its size and enhancing execution speed without compromising accuracy. The TensorFlow Lite model is fully compatible with Flutter, ensuring smooth integration and operation within the mobile application. This detailed architecture highlights the innovative and efficient design of our honk detection system, tailored for real-time, on-device processing in mobile environments.

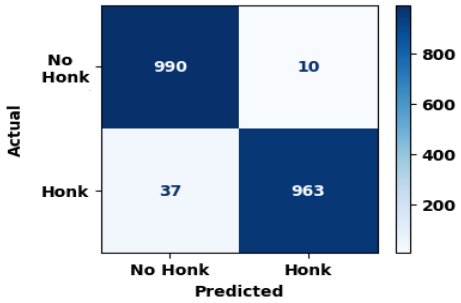


Figure 4: Confusion matrix of our proposed model

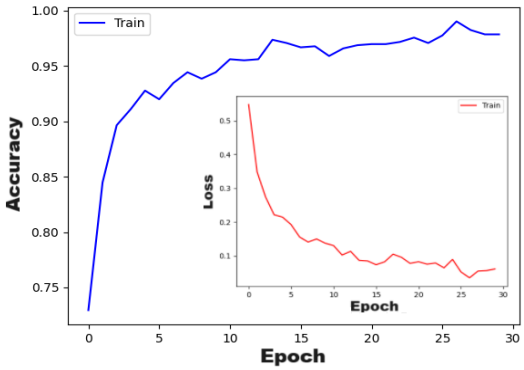


Figure 5: Epoch vs accuracy plot at the time of the training (inset plot represents the Loss vs Epoch plot at the time of training)

V. Results and Analysis

A. Performance analysis of our proposed model

The performance analysis of our proposed model demonstrates its efficacy in detecting honks within diverse audio environments, which is crucial for its integration into a mobile application. For the dataset, we utilized a training set comprising 16,000 honk samples. This is augmented to create 30,720 samples for training, equally divided between honk and non-honk samples, and 2,000 samples for testing. Our CNN model, designed to be lightweight and efficient, employs a binary cross-entropy loss function, the Adam optimizer with a learning rate of 0.001, and a batch size of 8. The image input shape is (1, 128, 128). The model achieved an impressive accuracy of 97.65%, precision of 0.9897, recall of 0.963, kappa score of 0.953, ROC AUC of 0.9982, and MCC of 0.9533. In Fig. 4, the confusion matrix illustrates the performance of our classification model on test data with known true values. The matrix shows a True Positive (TP) rate of 96% and a True Negative (TN) rate of 99%. These results demonstrate that our proposed model performs exceptionally well in accurately identifying honks and non-honks, highlighting its reliability and suitability for real-world applications. Moreover, in Fig. 5, we provided Epoch vs accuracy plot at the time of the training (inset plot represents the Loss vs Epoch plot at the time of training) to demonstrate how well our model is working.

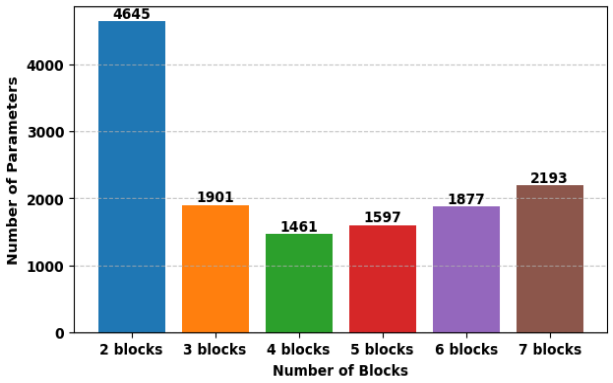


Figure 6: Required number of parameters for varying block size

B. Significance of block in the proposed model

In Fig. 3, we have used five blocks. Now, the question may arise: why five? The architecture of our model is chosen through a meticulous process to ensure the best performance while maintaining efficiency for mobile applications. To select the optimal model, we used a systematic approach by designing a block that includes a combination of convolutional layers, max pooling layers, and batch normalization layers. We then built multiple models by stacking different numbers of these blocks, training each configuration, and evaluating their performance. This method allowed us to find the best balance between accuracy and memory usage, which is crucial for deployment on resource-constrained mobile devices.

For our analysis, we constructed and evaluated six models, each with a varying number of blocks ranging from 2 to 7. We limited the maximum number of blocks to 7 to avoid redundancy and unnecessary complexity. The total number of parameters for each model configuration is depicted in the Fig. 6.

It is observed that all models achieved over 95% accuracy during training, however, their testing accuracy varied significantly. Notably, the model with 5 blocks demonstrated superior performance, achieving a testing accuracy of over 97.65%, outperforming models with both fewer and more blocks. We have observed that accuracy for 2-block, 3-block, 4-block, 6-blocks, and 7-blocks are 91.9%, 88.45%, 92.7%, 91.75, and 87% respectively. This indicates that simply increasing the depth of the model or the number of parameters does not necessarily improve performance. Instead, an optimal balance, as found in the 5-block model, is key to achieving high accuracy while maintaining efficiency.

C. Performance Analysis with Baseline Models

In this section, we compare the performance of our proposed model with baseline models like YAMNet [16], SBCNN [17], and TFCNN [18]. Table I illustrates that "Our Model" demonstrates a notable improvement in accuracy and other performance metrics compared to baseline models. Specifically, our proposed model achieves the highest accuracy of 97.65%, which translates to a 21.19% increase over SBCNN, the lowest-performing baseline with 76.46% accuracy, and a 15.4% increase over YAMNet, which has the highest baseline accuracy at 82.25%.

Table I: Comparing performance metrics of our proposed model with baseline models

Models	Accuracy	F1-Score	Precision	Recall
YAMNet [16]	82.25%	0.83	0.82	0.81
SBCNN [17]	76.46%	0.77	0.78	0.78
TFCNN [18]	79.85%	0.79	0.79	0.78
Our Model	97.65%	0.97	0.98	0.96

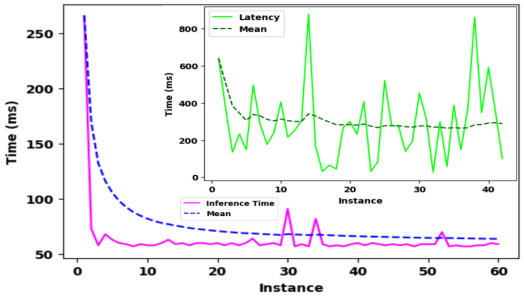


Figure 7: Depicted the inference time in (ms) and the inset plot represents the latency of proposed model

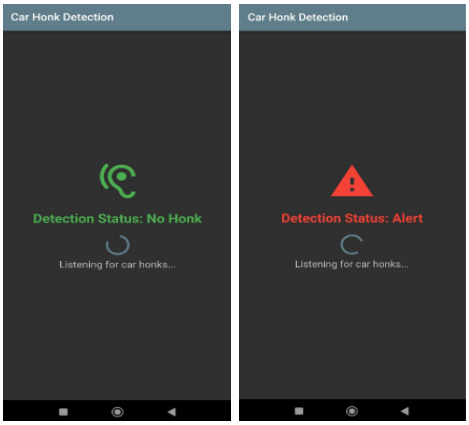
D. Estimation of Inference Time and Latency of our propsed model

In addition to evaluating classification accuracy, we analyzed the latency and inference time of our model to ensure real-time performance. The audio is recorded continuously in a stream. When the stream exceeds 1 second, the first 1-second audio clip is sent to the model for inference on a separate thread, and the initial 0.5 seconds are clipped. This process allows the model to detect honks every 0.5 seconds. Consequently, in n seconds, the model performs 2n inferences. The average inference time is 63.8 ms, as shown in Fig. 7.

We also measured the latency to assess the effectiveness of our proposed model. The model is designed to recognize a honk if there is a continuous honking sound for at least 0.25 seconds within a 1-second audio clip. Every 0.5 seconds, the model checks if the audio clip contains a honk of at least 0.25 seconds. Therefore, the best-case latency is 0.25 seconds if the honking starts at the beginning of the second half of the 0.5-second interval. The worst-case latency is 0.75 seconds if the honking starts after the beginning of the second half of the 0.5-second interval. The average latency is 288.79 ms, as depicted in the inset plot of Fig. 7. To accurately measure inference time and latency, we simulated a real-time scenario on an Android device. A pre-recorded, time-stamped audio sample is fed to the application, bypassing the microphone input. The application's alert timestamps are then recorded and compared to the original timestamps of the audio sample. The resulting time differences are visualized in a graph to analyze performance metrics.

E. Micro-Service

One use case of this application is for pedestrians walking and crossing streets while using earphones or headphones.



(a) (b)

Figure 8: Screenshot of our developed app, where (a) represents no honk is detected, and (b) denotes an alert message due to a honk

Upon startup, the application loads the model into memory and begins detecting car honks immediately using the device's microphone, and it can also run in the background, allowing the user to continue using their mobile device. Audio is continuously recorded at a sample rate of 22,050 Hz, and once the length of the recorded audio stream exceeds one second, the app extracts the first 1-second segment and sends it asynchronously to the TensorFlow Lite model for honk detection. This asynchronous processing ensures the application remains responsive and continues recording audio while processing previous data. After sending the 1-second audio clip to the model, the app removes the first 0.5 seconds of audio from the stream, freeing up resources and setting up the cycle to detect a honk every 0.5 seconds, maintaining continuous recording without overlap or redundancy. The TensorFlow Lite model processes the 1-second audio clip and returns a confidence score between 0 and 1. If the score exceeds 0.7, the application classifies the sound as a honk and alerts the user by playing a sound in their earphones and triggering a 0.5-second vibration, ensuring a high confidence level in honk detection and reducing false negatives. The snap of our developed app running in Android is depicted in Fig. 8, where Fig. 8(a) illustrates a scenario where a honk is not detected, while Fig. 8(b) shows real-time detection of a honk.

VI. Conclusion

This research introduces a novel approach to car honk detection aimed at enhancing pedestrian safety, particularly for individuals using earphones or headphones in urban settings. We have developed a lightweight deep learning model optimized for resource-constrained mobile devices, achieving high accuracy (97.65%) and low latency (average 49.89 ms) in real-time honk detection. The integration of this model into an Android application demonstrates its practicality, providing timely auditory alerts to users upon detecting honks, thereby mitigating potential accidents. Our findings underscore the feasibility of deploying efficient honk detection systems using modern smartphone capabilities, contributing not only to pedestrian safety but also to noise pollution monitoring in urban areas. In the future, we can add Cross-platform compatibility, real-time data analytics, and user feedback mechanisms for optimizing system performance and user experience. Moreover, we can use other audio processing techniques, such as multi-modal sensor fusion, could improve detection accuracy and reliability.

References

[1] R. Vijay, A. Sharma, T. Chakrabarti, and R. Gupta, "Assessment of honking impact on traffic noise in urban traffic environment of nagpur, india," *Journal of environmental health science and engineering*, vol. 13, pp. 1–10, 2015.

[2] R. Sen, B. Raman, and P. Sharma, "Horn-ok-please," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pp. 137–150, 2010.

[3] R. Banerjee and A. Sinha, "Two stage feature extraction using modified mfcc for honk detection," in *2012 International Conference on Communications, Devices and Intelligent Systems (CODIS)*, pp. 97–100, IEEE, 2012.

[4] B. Maity, A. Alim, S. Bhattacharjee, and S. Nandi, "Dehonk: A deep learning based system to characterize vehicular honks in presence of ambient noise," *Pervasive and Mobile Computing*, vol. 88, p. 101727, 2022.

[5] F. Demir, D. A. Abdullah, and A. Sengur, "A new deep cnn model for environmental sound classification," *IEEE Access*, vol. 8, pp. 66529–66537, 2020.

[6] A.-M. O. Mohamed, E. K. Paleologos, and F. M. Howari, "Noise pollution and its impact on human health and the environment," in *Pollution assessment for sustainable practices in applied sciences and engineering*, pp. 975–1026, Elsevier, 2021.

[7] C. A. Dim, R. M. Feitosa, M. P. Mota, and J. M. de Morais, "A smartphone application for car horn detection to assist hearing-impaired people in driving," in *International Conference on Computational Science and Its Applications*, pp. 104–116, Springer, 2020.

[8] K. Takeuchi, T. Matsumoto, Y. Takeuchi, H. Kudo, and N. Ohnishi, "A smart-phone based system to detect warning sound for hearing impaired people," in *Computers Helping People with Special Needs: 14th International Conference, ICCHP 2014, Paris, France, July 9-11, 2014, Proceedings, Part II 14*, pp. 506–511, Springer, 2014.

[9] J. Palecek and M. Cerny, "Emergency horn detection using embedded systems," in *2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pp. 257–261, IEEE, 2016.

[10] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Esresnet: Environmental sound classification based on visual domain models," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4933–4940, IEEE, 2021.

[11] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal processing letters*, vol. 24, no. 3, pp. 279–283, 2017.

[12] C. Wang, D. Chen, L. Hao, X. Liu, Y. Zeng, J. Chen, and G. Zhang, "Pulmonary image classification based on inception-v3 transfer learning model," *IEEE Access*, vol. 7, pp. 146533–146541, 2019.

[13] F. Demir, D. A. Abdullah, and A. Sengur, "A new deep cnn model for environmental sound classification," *IEEE Access*, vol. 8, pp. 66529–66537, 2020.

[14] W. Mu, B. Yin, X. Huang, J. Xu, and Z. Du, "Environmental sound classification using temporal-frequency attention based convolutional neural network," *Scientific Reports*, vol. 11, no. 1, p. 21552, 2021.

[15] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Esresnet: Environmental sound classification based on visual domain models," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4933–4940, IEEE, 2021.

[16] N. H. Valliappan, S. D. Pande, and S. R. Vinta, "Enhancing gun detection with transfer learning and yamnet audio classification," *IEEE Access*, 2024.

[17] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal processing letters*, vol. 24, no. 3, pp. 279–283, 2017.

[18] W. Mu, B. Yin, X. Huang, J. Xu, and Z. Du, "Environmental sound classification using temporal-frequency attention based convolutional neural network," *Scientific Reports*, vol. 11, no. 1, p. 21552, 2021.