Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

A Participatory Urban Acoustics Platform Using Hybrid YAMNet–CNN Classification and Real-Time Geospatial Mapping

1st Reva Mahesh Kudchadkar
Department of Computational Intelligence
School of Computing, College of Engineering & Technology
SRM Institute of Science and Technology
Kattankulathur-603203, India
rk1332@srmist.edu.in

2nd Mehul Siwach
Department of Computational Intelligence
School of Computing, College of Engineering & Technology
SRM Institute of Science and Technology
Kattankulathur-603203, India
ms5254@srmist.edu.in

***Abstract*— Noise that is undesirable is becoming a problem in the contemporary cities. Sleep disorders, heart issues, focus and memory challenges are but some of the health issues that can occur after being exposed to high level of noise over the long run. Most of the current noise monitoring techniques in cities require fixed equipment that is unable to determine the origin of the noise, and is quite expensive to establish around the world. One possible citizen technology is NoiseTube that measures the volume but does not detect the source of the noise. Other techniques like SONYC require advanced equipment to deliver precise results, but cities usually do not have the financial means to acquire the costly equipment. UrbanEcho is one such platform delivered via the web which solves these problems by removing any specialised hardware. Any fellow could get a basic web site and post audio records which include location information. The platform then utilizes a pre-trained algorithm to isolate some specific sound patterns in the recordings. It, then, used a neural network that it built by itself to classify the recordings into 10 categories based on the type of urban noise. The findings online, as well as displayed on an interactive map, are immediately accessible to everyone, including those in the position of authority. The system scored an F1-score of 85.3% and an accuracy of 86.7% when tested using UrbanSound8K. The method was found to work in practical level of real world experiments carried out in four different urban areas, which comprise more than 500 volunteer recordings.**

Keywords — Crowd-sourced acoustic sensing, urban sound classification, deep neural networks, YAMNet embeddings, transfer learning, smart urban infrastructure, spatial noise analytics.

I. INTRODUCTION

City streets are rarely quiet. From the steady rumble of traffic and construction zones to the buzz of factories and the chatter of busy crowds, urban areas are filled with sounds at all hours. Scientists have found that constant exposure to these noises can take a toll on people's well-being—causing everything from hearing damage and heart problems to increased stress and trouble paying attention. Because of these risks, addressing city noise is a major concern for those in charge of public health. The World Health Organization warns that daytime sound levels above 55 decibels can be dangerous for the heart, yet in many rapidly growing cities, people are exposed to even greater noise on a regular basis. The old way of monitoring city noise relied on a network of fixed sensors, but these only report what's happening at a handful of locations and are costly to set up and look afterThe fact is that noises in the city are never constant and therefore it is hard to have a complete picture with the help of stationary sensors. The search of optimal way of tracking auditory perception of urbanites has become possible due to the spread of cellphones and the advances in sound analysis. Every person could record some short videos of the things around him/her using his phone and post them on the site of UrbanEcho. The platform quickly identifies the type of noise collected, and also provides the information immediately with an interactive map. This approach offers a real-time and all-encompassing view of noise issues in a large number of neighbourhoods to municipal officials with minimum investment in permanent equipment.

UrbanEcho is a flexible web-based platform and a large number of independent elements that can be connected to each other through clearly defined web-based interfaces. It then converts the audio into 16 kHz single-channel audio, and sets the volume to compensate differences between different microphones when uploaded by the user via their browser. This is accomplished with the use of the Librosa library to ensure that the audio is in the uniform format. The second step is to feed the processed video to YAMNet, a Google machine learning model that has millions of audio samples to consult. YAMNet translates the recording into numerical representation of the sound characteristics, such as its energy, rhythm and tone. Then the data is analysed by a second specialised neural network and assigned a confidence score, and put in one of 10 categories of noise in the city. All uploads are checked to be valid and processed by the system, with Flask used to handle all the processing, receiving files to performing the analysis, and storing the results. To provide users with access to the noise report on a digital map, the front end is based on Leaflet.js. And it also summarizes data in the form of heat maps, such that anyone can visualize the origin of noise again and again.

Here, the structure of the paper is summarised. The next section discusses the historical background of interested studies referencing issues like the classification of urban noises, involvement of people in noise data gathering, and different techniques used in mapping noise levels. Moreover, it clarifies how UrbanEcho is different to those that had already been made. The following section of the paper explains how the UrbanEcho works, starting with the initial step a user takes, continuing with processing of audio files, and concluding with how all the various components of the system interact, including the YAMNet audio analysis engine and the mapping and storage management systems. This is followed by a description of the data that is utilized to train the system, and a visual representation of what each audio clip undergoes in order to become an image on a noise map, starting with the uploading. Then we will discuss the findings of our experimental work, the ability of the tool to differentiate between different types of urban noise, our test work experience using the tool in real neighbourhoods, and its effectiveness as compared to the popular datasets. Next we give the outcomes of a pilot study which was carried out on four cities. We considered the following: the number of people involved, the ability to detect different types of noise by the system, problem spots, and the time of uploads to the recordings. Part two goes into the advantages and disadvantages of utilizing the feedback of the general population, the merits of integrating YAMNet and a CNN, and the role of these by helping challenge planners to keep in touch with city noise. Finally, we make a comparison between the results of UrbanEcho and other platforms, highlight its weaknesses, and offer ideas on how these could be enhanced in the future.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

II. RELATED WORK

Although it affects the daily lives of people, noise in cities barely receives an attention in health policy. Noise, unlike smelly or visible pollutants, is hard to trace down as it varies with changes in factors such as traffic, construction work and social occasions. The majority of cities have few fixed cameras, which are only able to capture the background noises and leave certain areas unguarded. These monitors are very expensive to install and maintain and they are able to measure ambient noise, but not the source of noise. Therefore, be it automobiles, incidences, or equipment, the local authorities do not necessarily know everything to address the underlying cause. To manage urban noise we must have tools with which we can identify the origin of the noise or its magnitude.

Salamon et al. [1] gathered the UrbanSound8K dataset that consists of 8,732 tagged audio files that brought about 10 common urban sounds such as as horns, jackhammers, dogs barking, and street music. The ability to to the dataset smartly divides it into 10 groups allows researchers to train and test their algorithms independently of each other. UrbanSound8K is now a benchmark in research of urban sound detection because it has become a standard to this group. The same types of urban noises can be discussed and compared by researchers all across the world due to the clarity with which it separates them into groups. The data does not reflect the unpredictable, overlapping, and blurred sound you would expect when recording by an ordinary individual in a real-life setting as all the samples were done and labeled in a controlled environment. Although useful, UrbanSound8K [1] fails to reflect the spirit of real time classification of city sounds.

By training convolutional neural networks with AudioSet, a large collection of millions of audio samples categorised into more than 500 sound types, Hershey et al. [2] are able to show how these networks can be fine-tuned to behave well on more domain-specific sound classification tasks. Big, hand-labeled datasets are not as necessary when this strategy is used to achieve strong results on problems such as urban noise classification. Nevertheless, the labels of AudioSet are not always accurate and exhaustive, so there might be problems with the understanding of the model that should be corrected when it is tailored to a new task.

Piczak [3] demonstrated that convolutional neural networks are superior to earlier systems which relied on manually designed Mel-Frequency Cepstral Coefficients in case of inputting log-mel spectrograms in ambient sound recognition. By analyzing the spectrogram as an image, the network can automatically identify important timing and pitch patterns, without the need to select human features. This method performed miracles when attempting to distinguish between noises that otherwise are very similar. Unproblematically, having few data, neural networks might start internalising the training cases and give false results until they are further regularised or augmented with data.

Due to the nature of the co-occurring sounds in urban environments, Mesaros et al. [4] re-focused the interest of the sound event researchers on these environments. Their approaches to comparing and testing systems capable of recognising and classifying overlapping sounds in audio recordings are clear. Their research provided a standard to compare the performance of approaches to this task and demonstrated that single-label picking per recording is not adequate in an urban real-world audio. Regrettably, such systems, like those intended to support such an activity, require a large amount of processing resources, which are less efficient in terms of their ability to run quickly or in real-time.

Bello et al. [5] created a huge network of sound sensors spread throughout New York City, which was called SONYC. This technology could create detailed noise maps of the city through a combination of human and automated means of source labelling. Their findings showed that in the large regions, it is possible to identify and label city sounds automatically, and it is far more convenient to determine what makes the noise than to measure its level. The purchase, installation and maintenance costs of the equipment to be required by each of the permanent sensors that SONYC would have to install all over the city. Moreover, the system used manual data labellers and this implied that it took some time to update the noise maps of the city therefore not ideal to be used in real-time noise control.

A city-wide capture of accurate sound information without the need to purchase expensive equipment is achievable, as demonstrated by Stevens et al. [6] when people had volunteered to help. The scope of the project was not dictated by money in the city but rather by participation as the more people assisted, the more land was covered. This showed that crowdsourcing data collection can be as good as, or possibly better, than networks of fixed sensors. A significant shortcoming of their device, NoiseTube, was that it was only able to detect the relative volume of the noise and not the noise source itself. This information was therefore not enough to make decisions regarding individual noise sources because it merely indicated the overall noise levels.

Examining the existing research, a split can be seen: highly efficient sound discrimination systems can require highly specialised hardware, which makes them prohibitively expensive to use with a large scale. On the contrary, crowdsourcing systems can cover a vast area, yet they cannot tell the point where the noise originated. A system that will allow automatic detection of noise sources, allow regular people to accumulate data without any extra equipment, and present the results on a real-time map has yet to be developed. UrbanEcho fills this gap. It enables the user to record in their browser and real-time map updating with Firebase, and uses a combination of YAMNet and CNN models to classify sounds. This means that you can obtain source-specific noise data in multiple places at low cost and none of the technological challenges, as anyone can take part, no special application or technical skills are needed.

III. SYSTEM DESIGN AND METHODOLOGY

UrbanEcho is a web-based platform that has various layers for data collecting, signal processing, sound categorisation, information storage and map display. Each piece stands on its own, but also fits well with the others.

A. Dataset Description

The sound classifier was trained and tested using the UrbanSound8K dataset [1]. The dataset consists of 8,732 audio snippets of up to four seconds in length. Each clip is tagged with one of 10 sounds of a city. These include air conditioners, automobile horns, children playing, dogs barking, drilling, engines idling, pistol shots, jackhammers, sirens, and street music. The dataset is split into 10 groups to prevent mixing training and test data. recordings were made in various locations of the city The distance of the microphone from the sound , the background noise , and the gadget utilised for the recording all vary . These distinctions assist the assessment to be more realistic

The approach was evaluated using ten-fold cross validation as stated by Salamon et al. [1]. This allows the findings to be compared with other research. Some noises, like gunshots or sirens, are less frequent in the data. This is why the performance was tested in terms of accuracy, precision, recall and F1-score apart from accuracy. Accuracy alone might be deceiving if the model misses rarer noises.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

B. Platform Architecture and Technology Stack

UrbanEcho combines five strata of technology to construct the entire piece.

User interface is right inside your web browser-there is nothing to install. Anyone is able to record and transmit sounds with any modern browser. When uploading a recording, your browser will add the time and your GPS position to the audio and upload it. The site does not require an account, user name, or personal information, and is easily usable and privately sensitive. It is completely constructed using HTML,CSS and JavaScript and therefore can be easily run on various devices.

Audio Preprocessing The initial stage of handling every sound recording is opening it with the Alban Librosa [2], a Python audio tool. All the files are converted into mono and to 16 kHz, which fits the system. The volume is matched in such a way that it does not have issues with variations between the microphones. In case of long periods of silence, such may be eliminated and the attention paid to the actual sounds. These measures are suitable to ensure that the audio is clean and consistent then it is analyzed.

Feature Extraction YAMNet Once the audio is cleaned up we run it through YAMNet [3]. YAMNet is a tool developed by Google, which was trained on millions of various sound clips on AudioSet, which had hundreds of different types of sounds. It converts any sound into a series of 1,024 digits that adds up to the key characteristics of the recording- such as its tone, rhythm and power- without waiting to be crafted by hand. Since YAMNet was trained with such an enormous assortment of sounds, it will be able to identify even some unusual noises in new data. This is consistent with what Hershey et al. [4] discovered: the larger and more varied datasets trained on, the better able are models to address more specific tasks. YAMNet is deployed straight away as a TensorFlow Hub library, with its settings fixed throughout the remainder of the training, which maintains running efficiency as well.

Noise Classification Custom CNN A have a lightweight convolutional neural network with only ten UrbanSound8K classes trained on the embedding of the YAMNet will take the YAMNet embedding as input and output a predicted class label and a floating-point estimated confidence. This is accomplished via a number of dense layers with ReLU functions, and decreasing out littering layers to enhance trustworthiness, particularly when the audio records are of lesser quality, which is frequently an unavoidable element when participants chronicle in uncontrolled conditions. Learning happens with the Adam optimizer [5] and categorical loss based on cross-entropy. Displaying the confidence score on the interface provides the user with real time feedback regarding the confidence level of certain predictions which is not common with such tools. Earlier studies indicated by Piczak [6] that convolutional networks based on learned spectral features perform better than those based on manually constructed MFCCs. In this case, the design expands upon that work with the transfer-learned embeddings of YAMNet in place of log-mel features, which offer finer details of sound without requiring additional labelling.

Data Storage — Firebase Realtime Database Events of all kinds are stored in Firebase Realtime Database. This service was selected based on three primary reasons that can not be addressed by traditional relational databases. First, it uses WebSocket technology thereby ensuring that any new information is immediately depicted on all browsers connected to it, so the Leaflet map is real-time as and whenever a result is available. Second, due to its document-based format it can be easily stored with metadata such as labels, confidence, timestamps and coordinates, and these attributes can be added, or additional features added, later on without any complex migrations. Third, scaling is automatically managed by Firebase, which means that a database administrator does not have to work on managing frequent spikes in activity.

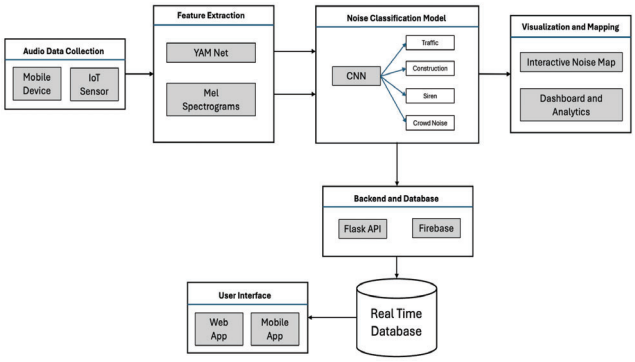

Fig.1. Workflow Diagram

The workflow diagram illustrates how the user's recording is processed in the UrbanEcho system, from the recording in the web browser to the rendering as a labelled marker on the live noise map.

YAMNet takes the audio input and maps it to a set of 1,024 integers that characterise the sound qualities. The CNN classifier then utilises this data to select the noise category and the confidence level of this decision. Firebase stores these data, together with the GPS position and the time the sound was captured and they display right away on the Leaflet map for everyone to view. The graphic also shows that the API backend, processing engine and user interface are distinct so that each portion may be scaled up independently if lots of people submit recordings at the same time

.

C. Real-Time Processing and Geospatial Insights

Once somebody sets up a clip, it will just take some few seconds before it comes up as a pin on the map. Such instant update is quite unlike the normal noises complaints that take out batched processing. UrbanEcho can display real-time noise events to city workers, like an unexpected group, a bang of equipment or a group of emergency-response vehicles, instead of rebuilding it based on historic information.

On receiving noise complaints, the resulting information is pooled into a map that is almost like the layout of the city. Where there are major roads there will be traffic noise and where there is building work there will be construction noise. Parks and pedestrian areas will also make noises to you. To include a time aspect to your noise map can show that you are experiencing a peak of noises during the rush hour or evening concerts that would otherwise not have been apparent on a bare noise map. And being aware what sort of sound it is. When a place is repeatedly mentioned in construction noise, the city may vary the work schedules or constructing sound barriers. However, in the case of live music, perhaps it is better to permit places where they can perform than to close them.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

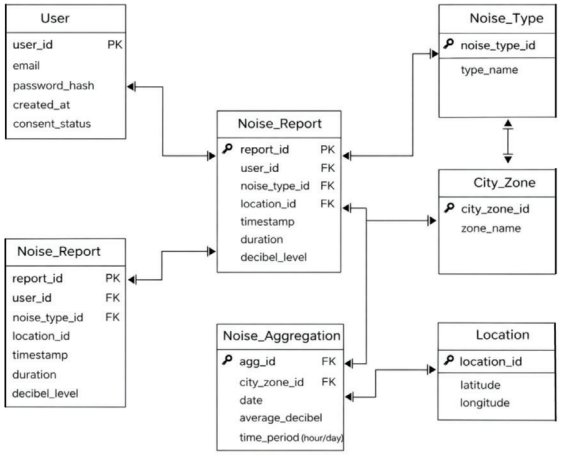

Fig. 2. Schema Diagram

A Noise_Report records the person doing the reporting, the type of noises that were detected, the location where it was observed (including coordinates), the time it occurred and an approximate volume. Noise webs The noises are created between urban areas and locations, storing averages ahead of time in an optimized, pre-computed form, to make the heatmap load quickly and prevent it from needing to search all locations each time. This format enables it easy to search the particularities of a single event by clicking on a marker or to obtain general noise patterns of larger regions - handy when city employees need to check noise by hour or day.

D. Deployment

The full platform is running within Docker containers [7]. This means for developers, everything functions the same way as it does on the cloud. The three major pieces - API backend (Flask), inference engine (TensorFlow) and frontend server (JavaScript) are all running in separate containers. Each has its own constraints on resources and regulations for starting back up if there are problems.

IV. RESULT

In UrbanEcho construction, the team trained its classifier with UrbanSound8K [1] dataset which composed more than 8,700 labeled audio clips of a variety of ten noise categories in a city. This data was made sure to be split in such a way that the sounds that would be used in training it were segregated and the tests on the system were those that were not used to train the system. With a recording, it is converted to standard format (mono, 16 kHz) first using Librosa [2]. This is followed by the analysis of the audio by YAMNet [3] to form a detailed profile generated due to the patterns it has learned on millions of other sound samples. The thing is that a custom neural network determines what type of noise was captured and its confidence with that judgment. This data, unlike the how and when of the capturing of the sound is stored in Firebase and instantly displayed as a point on the map. Backed by Flask, the backend receives and processes recordings, verifying any errors to proceed. To train the classifier on top of YAMNet, a general strategy, known as categorical cross-entropy and Adam optimizer, is used [5]. It is designed in HTML, CSS, and JavaScript, and based on Leaflet.js to display clusters and fluctuations in noise across the city. All of this works in Docker [7] containers, by keeping things operating identically in both testing and live, and allowing each component of the platform to deal with additional users when required.

A. Model Description

UrbanEcho sorts sounds with the help of two interconnected systems. The former is a Google model, YAMNet [3], which is already trained on millions of audio files. YAMNet hearkens to a new recording and converts it into a summary- a sequence of numbers which catalog the significant characteristics of the sound. The second, and specially designed neural network uses those figures and determines what it is that type of noise. The YAMNet itself is not modified, such that it retains all the general knowledge that it is initially trained with, with the only difference that the second network is also only trained to work in the specific job of UrbanEcho. This is effective and suitable to such targeted work as the study by Hershey et al. demonstrates [4].

The custom neural network of UrbanEcho considers the features of YAMNet and classifies each sound into one of the ten UrbanSound8K categories. It is a constructed network with a few dense layers activated by ReLU, alongside dropout layers to maintain consistency in results, although recording quality is not necessarily high. It is trained with Adam optimizer [5] and the loss function is categorical cross-entropy. Previous studies by Piczak [6] demonstrated that application of features learned by neural networks performs better as compared to older, hand made features (MFCCs). UrbanEcho extends this concept by basing on the functionality of YAMNet rather than log-mel ones, which is more detailed without additional labelling effort. Every prediction is also accompanied by a confidence score to allow users to know how confident a system is about its decision which many other systems do not provide.

B. Evaluation Metrics

The model findings were tested using four distinct metrics, outlined below.

Accuracy is the fraction of test instances that the model categorised correctly as the proper noise type out of all 10 categories

.

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

Precision checks of each category how many instances that the model predicted to be in a group actually were in a group. It aids in demonstrating an instance of model errors through guessing false.

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

The recall examines each category and sees what part of the genuine examples model could locate. It measures how frequently the model misses something it should have picked up.

$$Recall = \frac{TP}{TP + FN}$$

The F1-Score brings precision and recall together into one number, making sure both are taken into account. This is useful, especially when some types of city sounds are much more common than others.

$$F1 - Score = \frac{2\ (Precision)\ (Recall)}{(Precision) + (Recall)}$$

Sharing all four metrics is key, because certain city noises such as gunshots and sirens occur far less often than say traffic or construction. If you simply look at accuracy, the results may appear better than they actually are, even if unusual sounds are missed.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

C. Classification Performance and Model Comparison

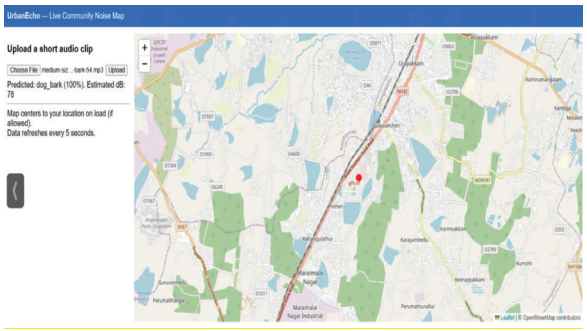

Fig. 3. Noise Input

F figure 3. The UrbanEcho website after a user has uploaded a recorded audio. Once a file is selected within the browser, the website rapidly reveals the estimated noise kind and the system's confidence on its decision. And at the same time, the map jumps to the user's GPS position, so when the system is done, it may drop a pin for the new recording.

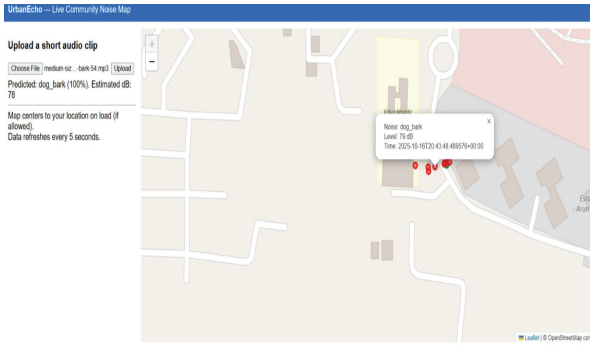

Fig. 4. Real-Time Predictions

The live map after community members upload recordings is shown in Figure 4. Each event has its own symbol on the map and you can mouse over it to discover what sort of event it is. The most recent results are shown in the page's sidebar, with the system's confidence level and the time of each result.

TABLE I. COMPARISON OF CLASSIFICATION MODELS ON URBANSOUND8K

S.No.	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
1	Random Forest + MFCC	72.3	71.1	70.4	70.7
2	MLP + MFCC	75.8	74.2	73.9	74.0
3	CNN (Baseline)	78.4	76.9	75.8	76.3
4	CNN + Log-Mel Spectrogram	81.2	80.1	79.5	79.8
5	VGGish + Linear Classifier	83.1	82.4	81.9	82.1
6	YAMNet + CNN (Proposed)	86.7	85.9	84.8	85.3

The UrbanSound8K data was used to test six different models. Below are examples of some using simple, hand-crafted features and others more sophisticated deep learning architectures. Each model was tested by the ten-fold cross-validation technique. The results presented in Table I show that the YAMNet-CN model was the most accurate with a total of 86.7% and a F1-score of 85.3%. This design achieved us 3.6 percentage points high compared to VGGish-linear. This shows that using a convolutional network to extend the features already learned can be used to more accurately differentiate between different sounds. It also scored lower than MFCC-CNN by 5.5 points, which can be characterized by the advantageous effect of learning on a large sample of sounds. In comparison, easier models, such as Random Forest and MLP with MFCCs did not perform as high and it fits the results of other studies conducted by Boddapati et al. [8]- deep learning is more successful than previous methods in analysing ambient noises. Overall, the leap between basic and deep models presents a lesson that the file of representations of sound data is more crucial than simply changing the classifier.

TABLE II. PER-CLASS PERFORMANCE OF YAMNET+CNN (URBANSOUND8K)

Noise Class	Precision (%)	Recall (%)	F1-Score (%)
Air Conditioner	76.8	79.4	78.1
Car Horn	92.0	90.5	91.2
Children Playing	81.5	83.2	82.3
Dog Barking	87.3	86.1	86.7
Drilling	88.9	87.4	88.1
Engine Idling	78.4	80.9	79.6
Gunshot	91.7	90.2	90.9
Jackhammer	89.5	91.9	90.7
Siren	94.1	92.8	93.4
Street Music	82.6	85.1	83.8

Table II displays the accuracy, recall and F1-score for each of the 10 UrbanSound8K sound categories when evaluated with the YAMNet–CNN model. The best recognition accuracy is obtained for sounds like sirens (F1: 93.4 %), automobile horns (F1: 91.2 %) and gunshots (F1: 90.9 %). These kinds of sounds are easy to identify from the others since the patterns of sound are significantly distinct from the other groups.

D. Pilot Deployment and Real-World Evaluation

The tests were conducted in four places: a neighborhood, a shopping and dining street, a transport center, and a college campus. These positions were selected to represent mixes of the various sounds you would have in a rollout into reality. The volunteers were given their own devices to record more than 500 clips during the study process and the time and location were automatically added into the browser with the submission. There was no instruction on the type of sounds to be captured, thus, making the findings more true-to-life.The recordings that people mailed in (approximately 78 percent of them) were mostly of traffic, construction, street music, running engines, and sirens, just as has been previously observed by earlier city sound surveys. The system was 81.3% sure their predictions were correct in accepting clips on average, demonstrating that it performed well even with audio which was not taken under the ideal conditions of the laboratory. Recording below 60% was scored as of low confidence and not included in the public map, typically due to noises in the background or lower-quality microphones and not necessarily due to a system malfunction. When a person uploaded a clip, it only required less than two seconds to be submitted and be displayed on the map, even when many people were using it at the same time. This compares to older systems where one required a person to go through results before them being posted online.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

E. Comparison with Related Systems

TABLE III. COMPARISON OF URBANEECHO WITH CLOSELY RELATED SYSTEMS

Attribute	**UrbanEcho (Proposed)**	**SONYC (Bello et al.)**	**NoiseTube (Stevens et al.)**
Data Collection	Citizen smartphone browser upload	Fixed hardware sensor network	Citizen mobile app (GPS + dB)
Classification Method	YAMNet + CNN (86.7% acc.)	DNN + human annotation	None (dB level only)
Noise Source Identification	Yes (10 urban categories)	Yes (multi-label tagging)	No
Real-Time Map Update	Yes (Firebase WebSocket)	Partial (delayed by annotation)	No (batch updates)
Hardware Required	None (browser-based)	Yes (acoustic sensor nodes)	Smartphone (app install)
Geospatial Visualization	Interactive Leaflet.js public map	Internal analyst dashboard	Public collective noise map
Scalability	High (Docker + cloud)	Limited (hardware bottleneck)	Medium (app adoption barrier)
Deployment Cost	Low (software-only)	High (hardware + maintenance)	Low (app distribution)

Table III considers UrbanEcho alongside SONYC [10] and NoiseTube [11] in eight key aspects. SONYC was effective at the New York City classifying sounds with the network of sensors, but necessitated a level of specialized equipment at every location, and involved a human to verify activities which in turn were costly and sluggish, particularly in cities with limited resources. NoiseTube also allowed individuals to gather noise data with their gadgets, but it was only used to record the level of loudness. That is the information only indicated the overall noise levels and not the cause of noise levels. UrbanEcho is as accurate as SONYC when it comes to determining what sound was recorded, though it allows anyone to add it with no special equipment- such as NoiseTube. But more to that, UrbanEcho is the only one that auto-updates its map in the real time, with Firebase. None of the existing systems have combined automatic sound recognition, no hardware required, automatic updates to the map, and the public site to be used by everyone in a single package. It is that mix that makes UrbanEcho unique.

F. Limitations

UrbanSound8K only has one label for each audio clip, yet genuine city sounds are usually a mixture, with many noises occurring at the same time. This reduces the ability of the classifier to deal with overlapping sound scenarios. There's also a data coverage problem, where locations where fewer people are participating or areas where not too many people have cellphones receive less data on the map. It's not been documented and the lack of this data might lead to the perception that certain locations are relatively calm. To combat this, communities should consider installing permanent sensors or putting obvious markings on regions with little data.

G. Future Prospects

UrbanEcho's future stages are focused on three major advancements. First, they're going to introduce live streaming, so users won't have to submit separate recordings. This will assist detect short-lived sounds that could occur in between regular uploads. Secondly they would want to include a tool that predicts future noise levels based on prior patterns in the data. Thirdly, they want to create a mobile application, to make it simpler for users to send in reports, particularly if they have problems using the internet. Over time as new data comes in from the community the model would be retrained periodically, to keep up with changes in city noise patterns. At last, the initiative may become a permanent tool for city planners, not merely a study project, by linking UrbanEcho directly to municipal environmental dashboards.

V. CONCLUSION

UrbanEcho shows you don't need fancy equipment to decode the city's sounds. Unlike sensors and hardware it provides a way for users to capture sounds using their phones or PCs. Almost 87 percent of the time the system was able to identify what kind of noise it was hearing, whether traffic, construction or music, better than anything else they matched it to. This helps municipal personnel understand how to react to diverse sounds, not simply how loud a location is." UrbanEcho runs in any web browser and doesn't require any specific equipment, so as more people hear about it, more people may take part.

Some things to work on. Now, the algorithm can only identify one primary sound per recording. But city life is usually loud, and frequently there are numerous things occurring at the same time. Also, certain neighbourhoods don't have as much data since fewer people there have cellphones, so such places may not show up on the map. The team intends to make it more simpler to use with live listening, tools to forecast future noise and a phone app. UrbanEcho will learn and develop as more people upload recordings. It might be linked directly to municipal systems in the future, and become a regular tool for city noise control.

REFERENCE

[1] J. Salamon, C. Jacoby, and J. P. Bello, "A Dataset and Taxonomy for Urban Sound Research," *Proc. ACM Int. Conf. Multimedia*, pp. 1041–1044, 2014.

[2] B. McFee et al., "Librosa: Audio and Music Signal Analysis in Python," *Proc. 14th Python in Science Conf. (SciPy)*, pp. 18–25, 2015.

[3] Google, "YAMNet: Pre-trained Audio Event Classifier," TensorFlow Hub, 2020. Available: https://tfhub.dev/google/yamnet/1

[4] S. Hershey et al., "CNN Architectures for Large-Scale Audio Classification," *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 131–135, 2017.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

[5] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *Int. Conf. Learning Representations (ICLR)*, 2015.

[6] K. J. Piczak, "Environmental Sound Classification with Convolutional Neural Networks," *IEEE 25th Int. Workshop Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2015.

[7] D. Merkel, "Docker: Lightweight Linux Containers for Consistent Development and Deployment," *Linux Journal*, vol. 2014, no. 239, 2014.

[8] V. Boddapati, A. Petef, J. Rasmusson, and L. Lundberg, "Classifying Environmental Sounds Using Image Recognition Networks," *Procedia Computer Science*, vol. 112, pp. 2048–2056, 2017.

[9] J. Kang, *Urban Sound Environment*, Taylor & Francis, London, 2006.

[10] J. P. Bello et al., "SONYC: A System for the Monitoring, Analysis and Mitigation of Urban Noise Pollution," *Communications of the ACM*, vol. 62, no. 2, pp. 68–77, 2019.

[11] N. Stevens, J. Steels, and L. Steels, "NoiseTube: Measuring and Mapping Noise Pollution with Mobile Phones," *Proc. Int. Symp. Information Technologies in Environmental Engineering (ITEE)*, pp. 215–228, 2009.

[12] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic Scene Classification: An Overview," *IEEE Signal Processing Magazine*, vol. 35, no. 3, pp. 81–90, 2018.

[13] Y. Tokozume and T. Harada, "Learning Environmental Sounds with End-to-End Convolutional Neural Network," *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2721–2725, 2017.

[14] S. Santini, A. Matic, and A. Osmani, "Wireless Sensor Networks in Smart Environments: Experiences and Challenges," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 614–628, 2009.

[15] L. M. Aiello, R. Schifanella, D. Quercia, and F. Aletta, "Chatty Maps: Constructing Sound Maps of Urban Areas from Social Media Data," *Royal Society Open Science*, vol. 3, no. 3, p. 150690, 2016.