IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 April 2023, accepted 7 June 2023, date of publication 12 June 2023, date of current version 19 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3285429

RESEARCH ARTICLE

Deep Convolutional Neural Networks for the Classification and Detection of Human Vocal Exclamations of Panic in Subway Systems

YO-PING HUANG[1,2,3,4], (Fellow, IEEE), AND RICHARD MUSHI[1]

[1]Department of Electrical Engineering, National Taipei University of Technology, Taipei 10608, Taiwan
[2]Department of Electrical Engineering, National Penghu University of Science and Technology, Penghu 88046, Taiwan
[3]Department of Computer Science and Information Engineering, National Taipei University, New Taipei City 23741, Taiwan
[4]Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan

Corresponding author: Yo-Ping Huang (yphuang@gms.npu.edu.tw)

This work was supported in part by the National Science and Technology Council, Taiwan, under Grant MOST108-2221-E-346-006-MY3 and Grant MOST111-2221-E-346-002-MY3; and in part by the Acer Group Research Project 210D001-1.

ABSTRACT The automated classification and detection of vocal exclamations of panic made by human beings in subway systems can enable more effective emergency response. Thus, in this study, we designed four multiscale deep convolutional neural networks (models 1-4) with one- and two-dimensional layers for detecting and classifying vocal exclamations of panic. First, we applied a decision-making framing-padding algorithm formulated to preprocess vocal exclamations of panic. Vocal sounds were then mixed with noise signals. Mel spectrogram, log-Mel spectrogram, and signal waveform data were used as learning data. The implementation of an ensemble technique in model 1 improved classification performance by 0.25% and 0.75% in terms of the F1 score at signal-to-noise ratios (SNRs) of 15 and −15, respectively. Models 4 and 2 exhibited the best classification performance and achieved F1 scores of 99.74% (under SNR = 15) and 80.56% (under SNR = -15), respectively. Model 2 performed the best in detecting screaming, quarrelling, and loud talking when SNR = 15 (F1 scores of 94.59%, 49.06%, and 64.94%, respectively). Model 2 also performed the best in distinguishing screaming and non-screaming. Our models outperformed their state-of-the-art counterparts in detection and classification at SNRs of 15 and 10.

INDEX TERMS Automatic classification, automatic detection, convolutional neural network (CNN), panic sounds, signal preprocessing.

I. INTRODUCTION

Automated systems can be used to detect exclamations of panic in an emergency and distinguish them from sounds from people quarreling or talking loudly; these systems are useful in subway systems because they can help station staff detect and react to emergencies quickly [1]. Thus, in the present study, we developed a convolutional neural network (CNN) that classifies and detects panicked vocalizations on the mass rapid transit (MRT) system in Taipei. We found that Taipei MRT cars operate quite smoothly. It is unusual to see passengers making much noise. Thus, loud exclamations are highly likely to stem from an emergency. However, at present, station staff are made aware of these sounds only if they are physically near the sound source or if a passenger makes a noise complaint.

Due to a lack of datasets for human vocal exclamation of panic and noise sounds, we developed our own two datasets: one dataset with vocal signals collected from MRT cars and the other dataset with noise signals. The vocal signals were subject to deep preprocessing because (1) they differed in sample size and (2) they differed in how the vocal volume evolved over time. For example, sounds of people talking loudly tend to fluctuate in volume because they are punctuated by brief moments of silence, whereas sounds of screaming are consistently loud. Thus, to ensure that these various types of signals can be processed by our system, we developed an algorithm that we call the decision-making

The associate editor coordinating the review of this manuscript and approving it for publication was Manuel Rosa-Zurera.

framing–padding algorithm. We blended sounds of human voices that were screaming, talking loudly, or quarreling with ambient noise, such as the sounds of moving or braking trains or sounds of alerts for closing doors. We did so to produce sounds that mimic those in a real-world train.

Feature extraction plays a key role in sound recognition [2]. Various two-dimensional (2D) and one-dimensional (1D) features have been proposed in the literature [3]. In the present study, we used spectrogram (2D) features and signal waveform (1D) features because they have been used in many promising developments in the field [4], [5], [6], especially those involving deep learning. The spectrogram is based on the transformation of the signal waveform into a time-frequency representation. In this manner, the amplitude of the human panic sound is varied over time at different frequency scales. On the other hand, signal waveform characterizes variations in the amplitude of human panic sound over time. Because each feature is differently described, we doubt that the performance of each feature will be different. Therefore, using two features concurrently can boost our model performance.

Our convolutional layers are based on a multiscale system, of which two types have been described in the literature [7], [8], [9]. Our multiscale system has different kernel sizes. The proposed system differs from its state-of-the-art counterparts in that 1D and 2D layers were consolidated and executed simultaneously on the same model that facilitates the presented work to overcome the drawbacks of each layer, resulting in more accurate and robust.

Additionally, four deep convolutional learning models were proposed. The models differed in structure which resulted in different performance benefits. The convolutions were 1D and 2D layers. The first model contained two models of 1D and 2D separately. These two models were concurrently trained, and their predicted probabilities were fused to each other. The remaining models were developed by consolidating the structure of the first model. The classification and detection performance of each model were compared with each other. Furthermore, for the first time, we compared the classification and detection performance of the proposed models with the state-of-the-art counterparts.

The remaining parts of the article are structured as follows. Section II provides a literature review. Section III details the classification phase. Section IV discusses feature extraction and deep learning models. Section V describes the detection phase. Section VI presents the results of our evaluation experiments and Section VII gives the conclusions of this study.

II. OVERVIEW OF THE RELATED RESEARCH

In this section, we overview the research related to signal preprocessing, noise and sounds in the subway systems, and deep learning models.

A. SIGNAL PREPROCESSING

Many signal preprocessing techniques have been proposed in the literature [10], [11], [12]. For example, in [13], a random-padding algorithm was proposed to eliminate temporal differences between environmental sound signals in preprocessing. In [14], a zero-padding system was created for preprocessing. In this system, signals of a shorter-than-normal duration are padded with 0s. In [15], a preprocessing system was developed for decomposing signals into small sizes.

B. NOISE AND SOUNDS IN SUBWAY SYSTEMS

In [16], sounds from bells ringing, trains moving, and trains braking, which are commonly found in subway systems, were analyzed. We used similar types of noise in this study; however, our implementation differed from that in [16], as described in Section III of this paper.

C. DEEP LEARNING MODELS

Various multiscale systems have been proposed in the literature. Gong et al. [17] proposed three types of multiscale CNNs with parallel convolutions to classify 1D, 2D, and three-dimensional hyperspectral images. Thuwajit et al. [18] proposed a multiscale CNN for detecting electroencephalogram seizures, and this model performed well on three datasets. Liu et al. [19] used a multiscale 1D CNN to diagnose motor faults and determined the optimal kernel size. Jiang et al. [9], formulated a multiscale coarse-grained system with 1D parallel convolutions to diagnose faults in wind turbine gearboxes.

The major objective and contributions of this paper are as follows:

1) Our system classifies and detects various vocal exclamations of panic. The classification was performed under different noise conditions whereas the detection was achieved by collecting more vocal sounds and sample evaluation to determine the conditions of classifying noise.
2) We compiled a dataset by recording sounds of vocal exclamations (which were then mixed with noise) or using similar sounds from existing databases.
3) We designed a decision-making framing-padding algorithm for preprocessing sound data. This algorithm differs from the framing [15] and padding [13] algorithms in six respects. First, our algorithm proceeds in two stages, and in each stage, the algorithm accepts a sound signal only if it is louder than a certain threshold (the level of human audibility in this study). Second, signals with a smaller-than-desired sample size are fed into the algorithm again until the desired sample size is obtained. Third, no overlapping samples are present between frames. Fourth, random padding is not used. Fifth, zero padding is not used. Sixth, each generated signal is labeled.
4) We developed two algorithms that are implemented during classification and detection. One algorithm fuses predicted probabilities from two models in the classification phase, and another algorithm fuses the

TABLE 1. Data collection for classification data.

Class	Primary data			Secondary data		Total Dur[b]. (s)	Total QTY
	QTY	Seg[a]. QTY	Dur[b]. (s)	QTY	Dur[b]. (s)		
Scream	46	98	269.9	186	343.5	613.4	**284**
L-talk	16	-	2830.3	8	1320.6	4150.8	**24**
Quarrel	5	-	721.2	18	940.2	1661.4	**23**
Total	**67**	**98**	**3821.4**	**212**	**2604.3**	**6425.6**	**331**

[a]Seg = segmentation, [b]Dur = duration, QTY = quantity, Scream = screaming, L-talk = loud talking, and Quarrel = quarrelling.

loud-talking and quarrelling categories into a non-screaming category. These algorithms aided the evaluation of how well our system distinguished screaming from non-screaming.

III. CLASSIFICATION DATA COLLECTION AND DATA PREPROCESSING ALGORITHM

In this section, we describe our approach to data collection, our decision-making framing–padding algorithm for data preprocessing, and our method for mixing vocal and noise signals.

A. CLASSIFICATION DATA COLLECTION

We collected primary and secondary data. To collect primary data, we recruited 17 volunteers (aged 21–26 years) who made sounds of screaming, talking loudly, and quarreling that were recorded on a Vivo mobile phone (Table 1). The distance between the sound source and the mobile phone was 50–300 cm.

Sounds of the volunteers talking loudly were recorded when they engaged in (1) conversation individually in groups of 2, 3, or 4 people and (2) conversation concurrently with others in groups of 2, 3, 4, or 5 people over 2–9 min.

Sounds of the volunteers screaming were recorded when they screamed individually or in groups. When the participants screamed individually, they screamed once, twice, or thrice. However, when the participants screamed as a group, they screamed once in groups of 2, 3, 4, or 17 (i.e., all participants together). The screams ranged from 0.5 to 7 s in duration.

Sounds of the volunteers quarreling were recorded in a similar manner to sounds of them talking loudly and screaming. The sounds of quarreling lasted for 1–4 min.

We collected the following secondary data from the following sources: 186 recordings of people screaming, 7 recordings of people talking loudly, and 18 recordings of people quarreling [20]. The vocal signals were down-sampled to 6000 Hz and then filtered using a pre-emphasis technique at a coefficient of 0.97. This down-sampling frequency was selected according to the maximum frequency of the screaming sounds [21] because screams tend to have the highest frequency among the considered vocalizations.

B. DECISION-MAKING FRAMING-PADDING APPROACH

Our decision-making framing–padding algorithm was designed to handle multidimensional signals in a dataset; it discards signals with voices that are softer than a given threshold (the level of human audibility in the present study). This algorithm has several nested loops (comprising if and for conditions) that govern whether it proceeds to a subsequent stage; thus, the algorithm is named the decision-making framing–padding algorithm. The inputs of this algorithm are audio signals, a reference power level, and a desired sample size. In the present study, the desired sample size was the dimension of the generated signal or frame and was set as 24 000 samples (equivalent to 4 s). The reference power was set as 1×10^{-12}. The output of the algorithm is a generated signal and its label. The algorithm proceeds as follows. In general, the algorithm determines the length of a signal and labels the signal. First, the algorithm determines whether the signal length is of a higher-than-desired sample size. If the aforementioned condition is achieved, then the ratio (N) of the desired sample size to the signal length is calculated. Subsequently, the *ceil()* and *tile()* functions are applied in sequence. The *ceil* function is used to round up N, and the *tile* function [30] repeats the entire current sample of audio signal N times until the desired sample size is attained. Once the desired sample size is reached, the signal power (in decibels) is calculated, as described in (1). Subsequently, the algorithm determines whether the signal power is above a given threshold (in this study, this threshold is the level of human audibility), and a signal is stored in a final dataset if and only if the signal power is higher than this threshold. Specifically, in this study, screaming and loud-talking sounds had to have power values of $\geq$100 and $\geq$90 dB, respectively.

Our description in the previous paragraph is for a situation in which the signal length is smaller than the desired sample size. Now, we describe how the algorithm processes signals of a length greater than the desired sample size. For these signals, the ratio ($N-1$), the signal length to the desired sample size, is calculated. Subsequently, the algorithm applies the *modf()* function to determine the integer and decimal part of $N-1$, each of which is processed separately. The integer part determines how many of the generated signals and their labels will be obtained. For example, if the integer is 2, it means two signals and their labels will be generated. Moreover, for each generated signal, the algorithm tests against the threshold, and then signals and their labels are generated. The decimal part indicates the remaining signal samples that need to be included. The remaining signal samples are usually less than the desired sample size, so we do the same as in the first stage, where the signal length was less than the desired sample size. After that, the algorithm again tests against the threshold, and finally, signals and their labels are generated. The application of this algorithm yields a set of signals of uniform sample size and acceptable power. The algorithm flowchart is illustrated in Fig. 1.

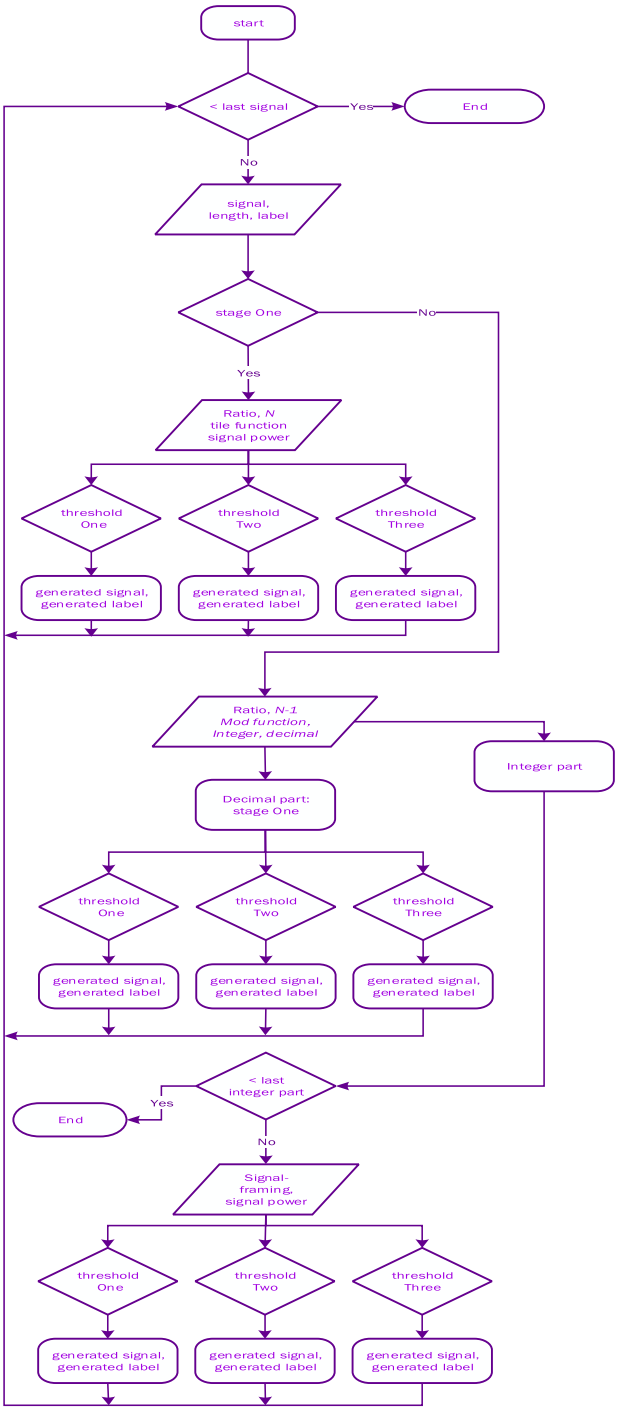

FIGURE 1. Proposed decision-making framing–padding algorithm.

The average power is calculated as follows:

$$P_{av} = \frac{1}{n} \cdot \sum_{k=1}^{n} s(t_n)^2 \tag{1}$$

The threshold (in decibels) is calculated as follows:

$$threshold = 10 \cdot \log_{10}\left(\frac{P_{av}}{ref}\right) \tag{2}$$

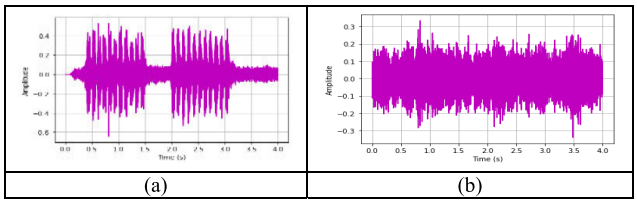

FIGURE 2. Waveforms of noise signals of (a) alert of door closing and (b) train braking.

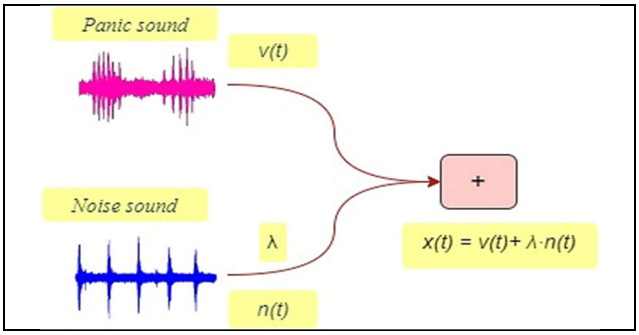

FIGURE 3. Overview of the traditional method of combining audio signals.

where $s(t_n)$ is the signal waveform, n is the signal dimension, and *ref* is the reference power.

In Fig. 1, thresholds one to three are the thresholds for screaming, loud talking, and quarreling, respectively.

C. NOISE AND SOUNDS IN MRT TRAIN CAR

We recorded noise of the following types in an MRT car: (1) trains braking, (2) doors opening and closing, (3) alerts of the door closing, (4) trains speeding up, and (5) train announcements. Recordings of train announcements were made close to and far from a train speaker. The noise signals had the same dimension as the desired sample size (Fig. 2).

D. TECHNIQUE FOR BLENDING AUDIO SIGNALS

The vocal and noise signals were linearly combined as per the traditional technique for combining audio signals (Fig. 3) [22]. This combination is encapsulated in the blending coefficient λ, which is dependent on the signal-to-noise ratio (SNR). The term λ is calculated using (3), where x(t) is the blended signal, v(t) is the vocal signal, n(t) is the noise signal, P_v is the power of the vocal signal, and P_n is the power of the noise signal.

$$\lambda = \sqrt{\frac{P_v}{P_n} \cdot 10^{-\frac{SNR}{10}}} \tag{3}$$

We initially collected a set of 331 signals, which became 1325 signals after being processed by the decision-making framing–padding algorithm (Table 1). These 1325 signals comprised 271 sound clips of people screaming, 393 sound clips of people quarreling, and 661 sound clips of people talking loudly. These signals were mixed with the noise signals, and the total number of signals increased to 7950 in the final

dataset, which comprised 1626 sound clips of people screaming, 2358 sound clips of people quarreling, and 3966 sound clips of people talking loudly. We split the whole dataset into 64%, 16%, and 20% for training, validation, and testing, respectively. So, there were 1590 data in the test dataset.

IV. FEATURE EXTRACTION AND DEEP LEARNING MODELS

A. FEATURE EXTRACTION

The audio signals in our dataset are spectrogram and time-domain signals. The spectrogram signals were from Mel and log-Mel spectrograms, whose formulas are presented in [14] and [23], respectively. The time-domain signals, which are denoted SIG-WAVE, comprised blended vocal and noise signals. Each spectrogram feature was also concatenated with its first and second derivatives. We adopted horizontal [5] rather than vertical concatenation [24], [25], [26], [27] and used a method proposed in our previous study [14], in which multiple image features are combined. The concatenations of Mel and log-Mel spectrogram features with their derivatives are denoted as MEL and log-MEL, respectively. The shape of each concatenated spectrogram feature was (32, 1204), and the shape of each SIG-WAVE feature was (24000).

B. DEEP LEARNING MODELS

We designed four deep multi-scale CNNs (named models 1 to 4). The goal is to choose some better models that outperform those state-of-the-art counterparts. Our CNN model adopted three kernel sizes to capture different regions of input features. These models are illustrated from Fig. 4 to Fig. 7, where blue and brown bars represent convolutional layers and maximum pooling layers, respectively. Maximum pooling is used after every convolution. In general, maximum pooling is advantageous because it results in smaller network sizes for deeper networks. The yellow bar represents activation layers, and the rectified linear unit function is used to boost nonlinearity. Green bars represent the global average pooling layers. Global average pooling is used to globally convert the dimensions of a feature map. Finally, black bars represent merge layers, which are used for concatenation.

Model 1 contains two sub-models: models 1a and 1b. Model 1a contains several 2D layers and two parallel branches, as illustrated in Fig. 4(a). The inputs for the two branches are MEL and log-MEL, respectively. Each branch then branches off further into three parallel streams. Each stream contains three convolutional layers, three maximum pooling layers, one activation layer, and one global average pooling layer. The convolutional layers kernel size for the first, second, and third streams were 3×3, 5×5, and 7×7, respectively. The number of filters in convolutional layers for each stream was set in the order of 8, 16, and 32, respectively. The maximum pooling size is 2×2. The global average pooling layer is used to convert 2D feature maps into 1D feature maps. Because two parallel branches are present, two merge layers are formed. These two merge layers are further concatenated for the overall feature maps to be generated. The extracted feature is then passed to the fully connected (dense) layer and dropout layer before being classified at the output layer, where the SoftMax function is used as the activation function. The number of units in the dense layer was 100, and a dropout of 50% was used in this study to avoid overfitting during training.

Model 1b receives signal waveforms as input; these waveforms are then fed to three parallel streams. These three parallel streams have an architecture that is identical to that of the parallel streams for one branch of model 1a in all respects except for the dimensionality of the layers. Model 1b contains 1D layers. The outputs of the parallel streams are then concatenated, and the resulting feature map is transferred to the fully connected layer and output layer, as illustrated in Fig. 4(b).

Model 2 is a consolidated model that is functionally similar to the combination of models 1a and 1b. The feature extracted from the consolidated structure is transferred to the fully connected layer and finally classified at the output layer (Fig. 5).

Model 3 is identical to model 2 except for the following point of difference. In model 2, the combined features formed through the combination of models 1a and 1b—are fed to the fully connected layer. However, in model 3, the combined features are first reshaped to allow them to be fed to three consecutive 1D CNNs. Each of these CNNs contains 16, 32, 64 filters, respectively, and has a kernel size of 10, 10, and 3, respectively. The padding in model 3 is the same as that in model 2. The output feature from the convolutional layer is then passed to the global average pooling layer, dense layer, and output layer (Fig. 6).

Model 4 is based on a modification of models 1a and 1b. Specifically, we removed the fully connected layer and output layers of models 1a and 1b and added one reshape layer and three consecutive 1D CNNs followed by a global average pooling layer, dense layer, and output layer to each of the two models (Fig. 7).

1) PROPOSED ENSEMBLE TECHNIQUE FOR MODEL 1

Ensemble techniques are used to improve classification performance. We trained models 1a and 1b simultaneously and fused their predicted probabilities. Our ensemble technique is illustrated in Fig. 8 and described in Table 2. First, the combined predicted probabilities for models 1a and 1b are represented as zero matrices (P_a and P_b, respectively). Subsequently, a combined predicted probability matrix (P) is constructed, where the element in row k and column $i$of P is the greater-value element between P_a and P_b in row k and column i.

2) EXPERIMENTAL SETUP AND EVALUATION METRICS

The experimental setup and performance metrics were similar to those in our previous study [14]. In addition, training ended after 50 epochs, the categorical cross-entropy function was

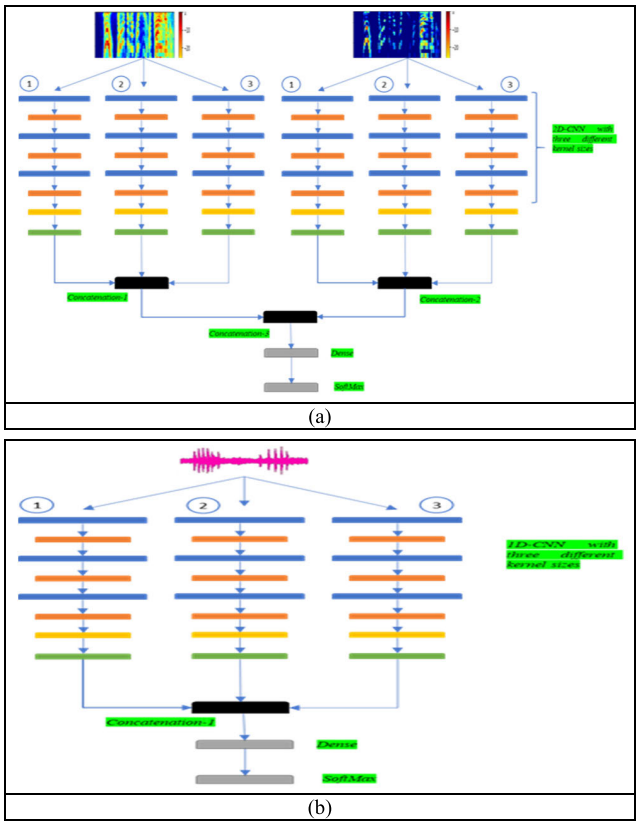

FIGURE 4. (a) Model 1a and (b) model 1b.

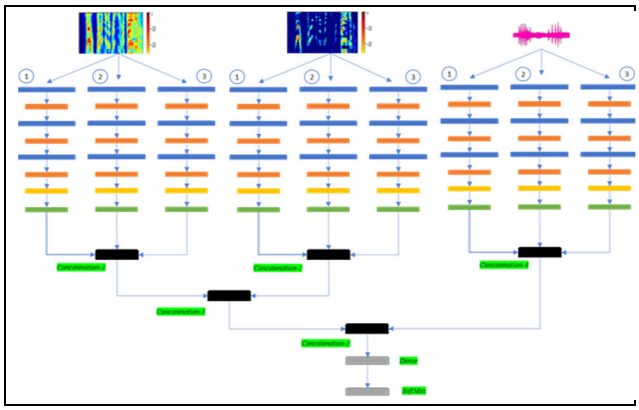

FIGURE 5. Model 2.

used as the loss function, and Adam was used as the optimizer.

$$\text{F1 score} = \frac{TP}{TP + 0.5(FP + FN)} \times w_i \tag{4}$$

$$\text{MCC} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{5}$$

where TP denotes the number of true positives, TN is the number of true negatives, FP represents the number of false positives, FN is the number false negatives, and w_i denotes the weight ratio of class i.

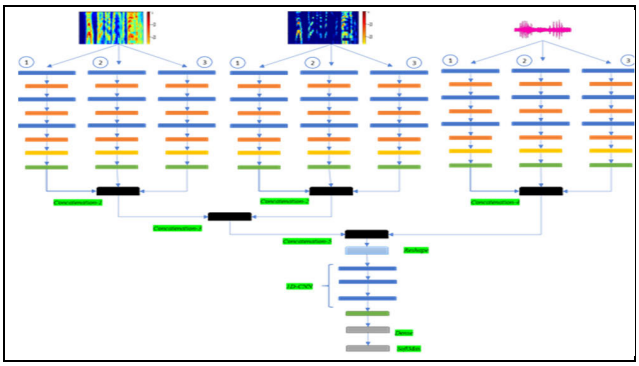

FIGURE 6. Model 3.

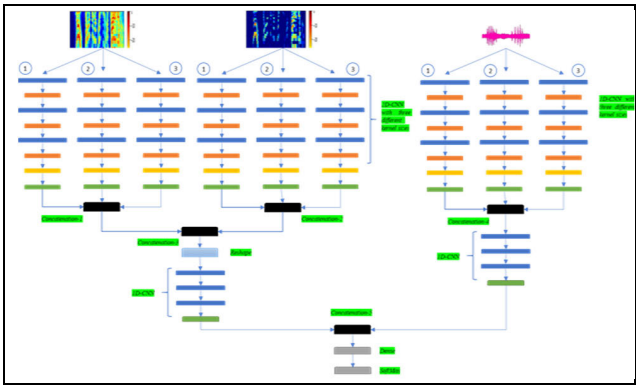

FIGURE 7. Model 4.

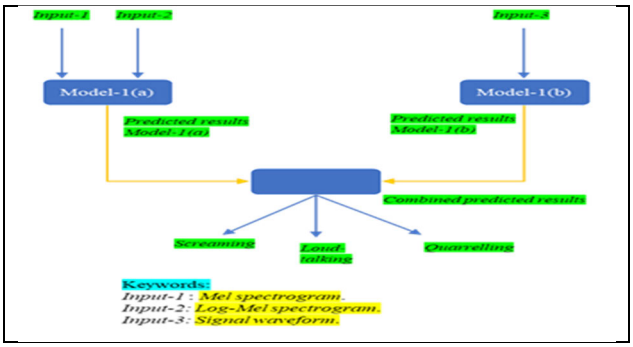

FIGURE 8. Proposed structure of an ensemble technique of model-1.

V. EVALUATION EXPERIMENTS

A. DATA COLLECTION FOR DETECTION

Data for the detection of human exclamations were collected and preprocessed in the same manner as that for classification data. Specifically, we recorded an initial sample of 15 audio clips; 7 clips were of volunteers screaming, 5 were of volunteers talking loudly, and 3 were of volunteers quarrelling. This sample expanded to contain 167 audio clips after being preprocessed with the decision-making framing–padding algorithm. These 167 clips comprised 37, 57, and 73 clips

TABLE 2. Code procedure for fusing predicted probability of model-1.

Input: predicted probabilities.
Output: combined predicted probability.
Initialization: combined predicted probability zero matrix.

```
For k in range (predprob1.shape [0])
    For i in range (predprob1.shape [1])
        If predprob1^b [k, i] > predprob2^c [k, i]
            combprob^a [k, i] = predprob1^b [k, i]
        else
            combprob^a [k, i] = predprob2^c [k, i]
```

combprob[a] = combined predicted probability,
predprob1[b] = predicted probability of model-1a,
predprob2[c] = predicted probability of model-1b.

of volunteers screaming, talking loudly, and quarrelling, respectively.

B. PROPOSED TECHNIQUE FOR FUSING CATEGORIES

Categories can be fused, and performance on some combination of categories can be compared with performance on one or some other combination of categories. We fused categories by combining the predicted probabilities of two categories and comparing the combined probability with the predicted probability of the third category (Table 3).

TABLE 3. Fusion of categories on the basis of predicted probabilities.

Input: predicted probabilities from two categories.
Output: a new combined predicted probability.
Initialization: a new combined probability with shape (*N, M*) array.
0, 1, 2 represent columns for three categories, respectively.

1. ***for*** k in range of N
2. ***for*** i in range of M
3. *if* $\boldsymbol{i} == 0$
4. $P_{ab}[k,i] = P[k,0] + P[k,1] - P[k,0] \cdot P[k,1]$
5. *if* $\boldsymbol{i} == 1$
6. $P_{ab}[k,i] = P[k,2]$

VI. RESULTS AND DISCUSSION

A. CLASSIFICATION RESULTS OF MODEL 1

The classification output by each model were also fused using our ensemble technique. Table 4 presents the classification results of model 1.

Model 1a exhibited its best performance when SNR = 15 ($F1$ score and Mathew correlation (MCC) of 99.24% and 0.9878, respectively) and its worst performance when SNR = −15 (i.e., high noise; $F1$ score and MCC of 77.20% and 0.6316, respectively). When SNR = 15, model 1a labeled 327 (actual number = 325), 476 (actual number = 468), and 787 (actual number = 785) clips as clips of screaming, quarrelling, and loud talking, respectively.

Model 1b exhibited its best performance when SNR = 15 ($F1$ score and MCC of 98.05% and 0.9686, respectively) and its worst performance when SNR = −15 ($F1$ score and MCC of 54.86% and 0.4850, respectively).

When the ensemble technique was applied, model 1 exhibited its best performance when SNR = 15 ($F1$ score and MCC of 99.49% and 0.9919, respectively) and its worst performance when SNR = −15 ($F1$ score and MCC of 77.95% and 0.6434, respectively). According to the confusion matrix of model 1 (Fig. 9), when SNR = 15, the model labeled 325, 472, and 793 clips as clips of screaming, quarrelling, and loud talking, respectively.

The results indicated that (1) the models performed better at higher SNRs, (2) model 1a outperformed model 1b at almost all SNRs, and (3) the ensemble method yielded a slight improvement in classification performance.

Actual Label		Predicted Label		
		Loud-talking	Quarrelling	Screaming
	Loud-talking	789	4	0
	Quarrelling	4	468	0
	Screaming	0	0	325

FIGURE 9. Confusion matrix of model 1.

TABLE 4. Classification performance of model 1 under different SNRs.

SNR	Model 1a		Model 1b		Model 1	
	F1 score (%)	MCC	F1 score (%)	MCC	F1 score (%)	MCC
15	99.24	0.9878	98.05	0.9686	99.49	0.9919
10	98.42	0.9746	97.74	0.9637	99.18	0.9868
5	97.35	0.9575	96.93	0.9507	98.23	0.9716
0	94.63	0.9138	94.96	0.9192	96.34	0.9412
-5	92.04	0.8719	87.73	0.8016	93.41	0.8938
-10	85.02	0.7612	81.79	0.7099	86.54	0.7826
-15	77.20	0.6316	54.86	0.4850	77.95	0.6434

B. CLASSIFICATION RESULTS OF MODEL 2, 3, AND 4

Table 5 presents the classification results of models 2, 3, and 4 when SNR = 15. Model 2 exhibited its best performance when SNR = 15 (F1 score and MCC of 99.37% and 0.9898, respectively) and its worst (but still satisfactory) performance when SNR = −15 (F1 score and MCC of 80.56% and 0.6870, respectively). Despite model 2 performed well in distinguishing between loud-talking and quarrelling sounds when SNR = 15; it misclassified four and six signals of quarreling and loud talking, respectively, as belonging to the other category.

Similarly, model 3 exhibited its best performance when SNR = 15 (F1 score and MCC of 99.56% and 0.9929, respectively) and its worst performance when SNR = −15 (F1 score

and MCC of 67.38% and 0.5054, respectively). Model 3 was also good in distinguishing between sounds of quarrelling and loud talking; it misclassified two and five signals of quarrelling and loud talking, respectively, as belonging to the other category.

Model 4 outperformed models 2 and 3. Model 4 exhibited its best performance when SNR = 15 (F1 score and MCC of 99.74% and 0.9959, respectively) and its worst performance when SNR = −15 (F1 score and MCC of 76.22% and 0.6402, respectively).

TABLE 5. Classification performance of models 2, 3, and 4 under different SNRs.

SNR	Model 2		Model 3		Model 4	
	F1 score (%)	MCC	F1 score (%)	MCC	F1 score (%)	MCC
15	99.37	0.9898	99.56	0.9929	99.74	0.9959
10	99.37	0.9898	99.30	0.9889	99.62	0.9939
5	98.42	0.9746	97.61	0.9617	98.73	0.9797
0	96.16	0.9384	94.75	0.9160	98.04	0.9686
-5	92.14	0.8745	88.43	0.8124	94.07	0.9049
-10	85.54	0.7668	82.85	0.7247	83.21	0.7313
-15	80.56	0.6870	67.38	0.5054	76.22	0.6402

C. COMPARISON OF CLASSIFICATION PERFORMANCE OF MODELS 1-4

Among the four models, model 4 performed the best overall, and models 1 and 2 performed the best under high-noise conditions. Specifically, if we set F1 score to be 80%, then all four models met the threshold under higher noise of SNR = −10 and model 2 can even sustain under SNR = −15.

Furthermore, model 4 had the highest number of trainable parameters (213715), followed by model 2 (185683), and model 1b had the lowest number of trainable parameters (58579). Note that these figures were indicated later in Table 6. Higher numbers of trainable parameters are more favorable.

D. COMPARISON OF THE PROPOSED MODELS WITH THEIR STATE-OF-ART COUNTERPARTS IN CLASSIFICATION

The proposed models were compared with their state-of-the-art counterparts at SNR values of 15 and 10 (Table 6).

The features used by the state-of-the-art methods differ from those in our dataset. We used only those features of Sharma and Kaul [28] that were feasible to implement when running their method. The method of Sharma and Kaul [28] achieved an $F1$ score and MCC of 95.54% and 0.9281, respectively, when SNR = 15. Moreover, their method achieved an $F1$ score and MCC of 93.60% and 0.8972, respectively, when SNR = 10. The use of SIG-WAVE features in the model 1b led to our models outperforming counterpart [28] in terms of $F1$ score (2.51% and 4.14% higher at SNR = 15 and 10, respectively).

For the method of Saeed et al. [22], we used the mean coefficients as a feature from the Mel-frequency cepstrum and combined them with the other remaining features. The method of Saeed et al. [22] achieved an $F1$ score and MCC of 95.09% and 0.9210, respectively, when SNR = 15. Moreover, this method achieved an $F1$ score and MCC of 93.77% and 0.8997, respectively, when SNR = 10. The use of SIG-WAVE features in the model 1b led to our models outperforming counterpart [22] in terms of $F1$ score (2.96% and 3.97% higher at SNR = 15 and 10, respectively).

We trained the model of Liu et al. [19] in the following manner. Because the blocks in the system of Liu et al. differ in their dimensions, we adopted one convolutional layer with a filter and a kernel size of 1 [29] to enable skip connections to be connected in the following block. We adopted a batch size of 4 and adopted 64 units in the fully connected layer. SIG-WAVE was used as the input. The method of Liu et al. achieved an $F1$ score and MCC of 96.38% and 0.9424, respectively, when SNR = 15. Moreover, this method achieved an $F1$ score and MCC of 90% and 0.8503, respectively, when SNR = 10.

For the method of Jiang et al. [9], we extended their architecture to contain four parallel branches. Four-scale, coarse-grained signals were then generated and fed to parallel branches. The method of Jiang et al. [9] achieved an $F1$ score and MCC of 99.56% and 0.9929, respectively, when SNR = 15. Moreover, this method achieved an $F1$ score and MCC of 99.87% and 0.9979, respectively, when SNR = 10.

TABLE 6. Classification performance of the proposed models and their state-of-the-art counterparts.

Method	Parameter	Feature set/model	SNR	F1 score (%)	MCC
Liu et al. [19]	1,421,641	SIG-WAVE	15	96.38	0.9424
			10	90.00	0.8503
Sharma et al. [28]	-	MFCC, spectral features	15	95.54	0.9281
			10	93.60	0.8972
Saeed et al. [22]	-	MFCC, ZCR, SC, RMSE	15	95.09	0.9210
			10	93.77	0.8997
Jiang et al. [9]	248,259	SIG-WAVE	15	99.56	0.9929
			10	99.87	0.9979
Gong et al. [17]	3,703	SIG-WAVE	15	92.13	0.8742
			10	91.19	0.8589
Ours-model 1a	127,507	MEL, Log-MEL	15	99.24	0.9878
			10	98.42	0.9746
Ours-model 1b	58,579	SIG-WAVE	15	98.05	0.9686
			10	97.74	0.9637
Ours-model 1	-	-	15	99.49	0.9919
			10	99.18	0.9868
Ours-model 2	185, 683	MEL, Log-MEL, SIG-WAVE	15	99.37	0.9898
			10	99.37	0.9898
Ours-model 4	213,715	MEL, Log-MEL, SIG-WAVE	15	99.74	0.9959
			10	99.62	0.9939

MFCC: Mel-frequency cepstral coefficient. Spectral features: centroid, roll-off, flatness, bandwidth (bandwidth is used instead of flux). ZCR = zero-crossing rate, SC = spectral centroid, RMSE = root mean square error.

TABLE 7. Detection results of model 1 for three categories.

SNR	Model	Category	F1 score (%)
15	Model 1a	Loud-talking	45.05
		Quarrelling	46.62
		Screaming	80.00
	Model 1b	Loud-talking	71.35
		Quarrelling	23.38
		Screaming	81.40
	Model 1	Loud-talking	50.42
		Quarrelling	47.24
		Screaming	81.82
10	Model 1a	Loud-talking	61.54
		Quarrelling	36.00
		Screaming	89.74
	Model 1b	Loud-talking	77.85
		Quarrelling	34.21
		Screaming	66.06
	Model 1	Loud-talking	65.79
		Quarrelling	34.78
		Screaming	77.78

TABLE 8. Detection results of model 1 for screaming and non-screaming.

SNR	Model	Class	F1 score (%)
15	Model 1a	Non-screaming	92.62
		Screaming	80.00
	Model 1b	Non-screaming	93.55
		Screaming	81.40
	Model	Non-screaming	93.50
		Screaming	81.82
10	Model 1a	Non-screaming	96.88
		Screaming	89.74
	Model 1b	Non-screaming	84.07
		Screaming	66.67
	Model	Non-screaming	91.80
		Screaming	77.78

TABLE 9. Detection results of models 2, 3, and 4 for three categories.

Model	SNR	Class	F1 score (%)
Model 2	15	Loud-talking	64.94
		Quarrelling	49.06
		Screaming	94.59
	10	Loud-talking	52.41
		Quarrelling	38.60
		Screaming	93.33
Model 3	15	Loud-talking	47.46
		Quarrelling	37.74
		Screaming	65.45
	10	Loud-talking	47.37
		Quarrelling	48.74
		Screaming	67.33
Model 4	15	Loud-talking	46.15
		Quarrelling	46.62
		Screaming	92.96
	10	Loud-talking	32.20
		Quarrelling	42.76
		Screaming	92.96

Among the state-of-the-art methods, the method of Jiang et al. [9] exhibited the best performance.

For the method of Gong et al. [17], we combined their 1D multiscale filter bank with the fully connected layer of our models. This method achieved an $F1$ score and MCC of 92.13% and 0.8742, respectively, when SNR = 15. Furthermore, it achieved an $F1$ score and MCC of 91.19% and 0.8589, respectively, when SNR = 10.

TABLE 10. Detection results of models 2, 3, and 4 for screaming and non-screaming.

Model	SNR	Class	F1-score (%)
Model 2	15	Non-screaming	98.46
		Screaming	94.59
	10	Non-screaming	98.07
		Screaming	93.33
Model 3	15	Non-screaming	83.04
		Screaming	65.45
	10	Non-screaming	86.32
		Screaming	68.00
Model 4	15	Non-screaming	98.10
		Screaming	92.96
	10	Non-screaming	98.10
		Screaming	92.96

TABLE 11. Detection results of the proposed models and their state-of-the-art counterparts for screaming and non-screaming.

Method	Parameter	Feature set/model	SNR	F1 score (%)	MCC
Liu et al. [19]	1,421,641	SIG-WAVE	15	51.09	0.3778
			10	55.38	0.4468
Sharma et al. [28]	-	MFCC, spectral features	15	88.87	0.6877
			10	90.18	0.7361
Saeed et al. [22]	-	MFCC, ZCR, SC, RMSE	15	85.75	0.6271
			10	83.26	0.5424
Jiang et al. [9]	248,259	SIG-WAVE	15	86.50	0.6692
			10	89.77	0.7419
Gong et al. [17]	3,703	SIG-WAVE	15	78.06	0.2354
			10	80.75	0.2429
Ours-model 1a	127,507	MEL, Log-Mel	15	89.82	0.7514
			10	95.29	0.8682
Ours-model 1b	58,579	SIG-WAVE	15	90.85	0.7645
			10	80.21	0.5911
Ours-model 1	-	-	15	81.81	0.7732
			10	77.78	0.7204
Ours-model 2	185, 683	MEL, Log-Mel, SIG-WAVE	15	97.60	0.9305
			10	97.02	0.9141
Ours-model 4	213,715	MEL, Log-Mel, SIG-WAVE	15	92.95	0.9119
			10	92.95	0.9119

E. DETECTION RESULTS OF MODEL 1

The developed models detected sounds from all three categories (screaming, talking loudly, and quarrelling) poorly when SNR = 10 (Tables 7). Specifically, model 1b achieved F1 scores of 77.85% and 34.21% for detecting loud talking and quarrelling, respectively. However, model 1a performed well in detecting screaming and non-screaming, with its F1 scores being 89.74% and 96.88%, respectively (Tables 8).

F. DETECTION RESULTS OF MODELS 2, 3 AND 4

Model 2 exhibited good performance and the best performance among all models when SNR = 15; it achieved F1

scores of 64.94%, 49.06%, and 94.59% in detecting loud talking, quarrelling, and screaming, respectively (Table 9). However, model 2 could not detect quarrelling well and often confused it for talking loudly.

Models 2 and 4 performed excellently in detecting screaming and non-screaming when SNR = 15 (Table 10). Model 2 achieved F1 scores of 94.59% and 98.46% for screaming and non-screaming, respectively, and model 4 achieved F1 scores of 92.96% and 98.10% for screaming and non-screaming, respectively.

G. COMPARISON OF THE PROPOSED MODELS WITH THEIR STATE-OF-ART COUNTERPARTS IN DETECTION

Model 2 performed considerably better than its state-of-the-art counterparts as shown in Table 11. It exhibited F1 score of 97.60% and MCC value of 0.9305 that outperformed their state-of-the-art counterparts by 8.73% $\sim$ 46.51% and 0.2428 $\sim$ 0.5527, respectively, when SNR = 15. Under SNR = 10, model 2 outperformed their state-of-the-art counterparts by 6.84% $\sim$ 41.64% and 0.1780 $\sim$ 0.4673, respectively. Model 4 exhibited the second-best performance, and it had an F1 score and MCC of 92.95% and 0.9119, respectively, when SNR = 15 and 10.

VII. CONCLUSION

In this paper, we propose a system that detects vocal exclamations of panic and distinguishes them from other types of vocal exclamations. We developed a decision-making framing–padding algorithm for preprocessing vocal sounds. Vocal sounds were then mixed with noise signals for data augmentation and for simulating actual sound that may occur in subways.

We created four deep CNN models that use MEL, log-MEL, and SIG-WAVE as inputs. These models differed in their classification performance. Model 1a outperformed model 1b at low and high SNRs. The application of an ensemble technique to model 1 improved its classification performance. Model 4 performed excellently ($F1$ score: 99.74%; MCC: 0.9959) at low noise levels, and model 2 performed satisfactorily (F1 score: 80.56%; MCC: 0.6870) at high noise levels. Model 4 outperformed its state-of-the-art counterparts, in part because of the features that it uses.

With regard to detection, models 2 and 4 exhibited better performance than their state-of-the-art counterparts in distinguishing screaming from non-screaming in both F1 score and MCC.

In future research, we aim to expand our dataset to cover different types of MRT noise, more categories of human vocal exclamations of panic, and to implement our system by using edge computers.

REFERENCES

[1] W. Huang, T. K. Chiew, H. Li, T. S. Kok, and J. Biswas, "Scream detection for home applications," in *Proc. 5th IEEE Conf. Ind. Electron. Appl.*, Taichung, Taiwan, Jun. 2010, pp. 2115–2120.

[2] K. Umapathy, S. Krishnan, and R. K. Rao, "Audio signal feature extraction and classification using local discriminant bases," *IEEE Trans. Audio, Speech Language Process.*, vol. 15, no. 4, pp. 1236–1246, May 2007.

[3] F. Alias, J. C. Socoro, and X. Sevillano, "A review of physical and perceptual feature extraction techniques for speech, music, and environmental sounds," *Appl. Sci.*, vol. 6, no. 5, pp. 1–44, May 2016.

[4] M. Cohen-McFarlane, P. Xi, B. Wallace, K. Habashy, S. Huq, R. Goubran, and F. Knoefel, "Evaluation of respiratory sounds using image-based approaches for health measurement applications," *IEEE Open J. Eng. Med. Biol.*, vol. 3, pp. 134–141, 2022.

[5] N. Peng, A. Chen, G. Zhou, W. Chen, W. Zhang, J. Liu, and F. Ding, "Environment sound classification based on visual multi-feature fusion and GRU-AWS," *IEEE Access*, vol. 8, pp. 191100–191114, 2020.

[6] W. Zhaoxia, L. Fen, Y. Shujuan, and W. Bin, "Motor fault diagnosis based on the vibration signal testing and analysis," in *Proc. 3rd Int. Symp. Intell. Inf. Technol. Appl.*, Nanchang, China, 2009, pp. 433–436.

[7] J. Wang, J. Zhuang, L. Duan, and W. Cheng, "A multi-scale convolution neural network for featureless fault diagnosis," in *Proc. IEEE Int. Symp. Flexible Autom. (ISFA)*, Cleveland, OH, USA Aug. 2016, pp. 65–70.

[8] R. Rasti, H. Rabbani, A. Mehridehnavi, and F. Hajizadeh, "Macular OCT classification using a multi-scale convolutional neural network ensemble," *IEEE Trans. Med. Imag.*, vol. 37, no. 4, pp. 1024–1034, Apr. 2018.

[9] G. Jiang, H. He, J. Yan, and P. Xie, "Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3196–3207, Apr. 2019.

[10] S. Aggarwal and N. Chugh, "Signal processing techniques for motor imagery brain computer interface: A review," *Array*, vols. 1–2, pp. 1–12, Apr. 2019.

[11] A. Keerio, B. K. Mitra, P. Birch, R. Young, and C. Chatwin, "On preprocessing of speech signals," *Int. J. Signal Process.*, vol. 5, no. 3, pp. 216–222, Jan. 2009.

[12] G. Monte, "Sensor signal preprocessing techniques for analysis and prediction," in *Proc. 34th Annu. Conf. IEEE Ind. Electron.*, Orlando, FL, USA, Nov. 2008, pp. 1788–1793.

[13] X. Dong, B. Yin, Y. Cong, Z. Du, and X. Huang, "Environment sound event classification with a two-stream convolutional neural network," *IEEE Access*, vol. 8, pp. 125714–125721, 2020.

[14] Y. Huang and R. Mushi, "Classification of cough sounds using spectrogram methods and a parallel-stream one-dimensional deep convolutional neural network," *IEEE Access*, vol. 10, pp. 97089–97100, 2022.

[15] A. Malek. (Jan. 2020). *Signal Framing*. Accessed: Jul. 20, 2022. [Online]. Available: https://superkogito.github.io

[16] P. Laffitte, D. Sodoyer, C. Tatkeu, and L. Girin, "Deep neural networks for automatic detection of screams and shouted speech in subway trains," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 6460–6464.

[17] Z. Gong, P. Zhong, Y. Yu, W. Hu, and S. Li, "A CNN with multiscale convolution and diversified metric for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3599–3618, Jun. 2019.

[18] P. Thuwajit, P. Rangpong, P. Sawangjai, P. Autthasan, R. Chaisaen, N. Banluesombatkul, P. Boonchit, N. Tatsaringkansakul, T. Sudhawiyangkul, and T. Wilaiprasitporn, "EEGWaveNet: Multiscale CNN-based spatiotemporal feature extraction for EEG seizure detection," *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5547–5557, Aug. 2022.

[19] R. Liu, F. Wang, B. Yang, and S. J. Qin, "Multiscale kernel based residual convolutional neural network for motor fault diagnosis under nonstationary conditions," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3797–3806, Jun. 2020.

[20] I. Trowitzsch, J. Mohr, Y. Kashef, and K. Obermayer, "Robust detection of environmental sounds in binaural auditory scenes," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 6, pp. 1344–1356, Jun. 2017.

[21] L. Villazon, "Can a human produced a sound outside the human audible range?" BBC Science Focus Magazine. Accessed: Jun. 30, 2022. [Online]. Available: https://www.sciencefocus.com

[22] F. S. Saeed, A. A. Bashit, V. Viswanathan, and D. Valles, "An initial machine learning-based victim's scream detection analysis for burning sites," *Appl. Sci.*, vol. 11, no. 18, pp. 1–22, Sep. 2021.

[23] T. Yan, H. Meng, S. Liu, E. Parada-Cabaleiro, Z. Ren, and B. W. Schuller, "Convoluational transformer with adaptive position embedding for COVID-19 detection from cough sounds," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Singapore, May 2022, pp. 9092–9096.

[24] Y. Su, K. Zhang, J. Wang, D. Zhou, and K. Madani, "Performance analysis of multiple aggregated acoustic features for environment sound classification," *Appl. Acoust.*, vol. 158, Jan. 2020, Art. no. 107050.
[25] Z. Mushtaq and S.-F. Su, "Efficient classification of environmental sounds through multiple features aggregation and data enhancement techniques for spectrogram images," *Symmetry*, vol. 12, no. 11, pp. 1–34, Nov. 2020.
[26] A. Tjandra, S. Sakti, G. Neubig, T. Toda, M. Adriani, and S. Nakamura, "Combination of two-dimensional cochleogram and spectrogram features for deep learning-based ASR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, South Brisbane, QLD, Australia, Apr. 2015, pp. 4525–4529.
[27] Z. Chi, Y. Li, and C. Chen, "Deep convolutional neural network combined with concatenated spectrogram for environmental sound classification," in *Proc. IEEE 7th Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*, Dalian, China, Oct. 2019, pp. 251–254.
[28] A. Sharma and S. Kaul, "Two-stage supervised learning-based method to detect screams and cries in urban environments," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 2, pp. 290–299, Feb. 2016.
[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

YO-PING HUANG (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Texas Tech University, Lubbock, TX, USA. He is currently the President of the National Penghu University of Science and Technology, Penghu, Taiwan. He is also a Chair Professor with the Department of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan, where he was the General Secretary. He was a Professor and the Dean of Research and Development, the Dean of the College of Electrical Engineering and Computer Science, and the Department Chair of Tatung University, Taipei. His current research interests include deep learning modeling, intelligent control, fuzzy systems design and modeling, and rehabilitation systems design.

He is a fellow of IET, CACS, and TFSA. He received the 2021 Outstanding Research Award from Ministry of Science and Technology (MOST), Taiwan. He serves as the IEEE SMCS VP for Conferences and Meetings and the Chair for the IEEE SMCS Technical Committee on Intelligent Transportation Systems. He was the IEEE SMCS BoG, the President of the Taiwan Association of Systems Science and Engineering, the Chair of the IEEE SMCS Taipei Chapter, the Chair of the IEEE CIS Taipei Chapter, and the CEO of the Joint Commission of Technological and Vocational College Admission Committee, Taiwan.

RICHARD MUSHI received the B.S. degree in electrical engineering from the University of Dar Es Salaam, in 2002, and the M.S. degree in telecommunication engineering from the University of Dodoma, in 2012. He is currently pursuing the Ph.D. degree in electrical engineering and computer science with the National Taipei University of Technology, Taipei, Taiwan.

Since 2009, he has been with the St. Augustine University of Tanzania as a Tutorial Assistant then promoted to an Assistant Lecturer. His research interests include artificial intelligence in healthcare specialized in cough sound analysis, the analysis of various kind of sounds, data mining, time series prediction, artificial intelligence in power systems, and the Internet of Things (IoT).