Received November 27, 2018, accepted December 13, 2018, date of publication January 8, 2019, date of current version January 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2888882

Sound Classification Using Convolutional Neural Network and Tensor Deep Stacking Network

ADITYA KHAMPARIA[1], DEEPAK GUPTA[2], NHU GIA NGUYEN[3], ASHISH KHANNA[2], BABITA PANDEY[4], AND PRAYAG TIWARI[5]

[1]School of Computer Science and Engineering, Lovely Professional University, Phagwara 144401, India
[2]Maharaja Agrasen Institute of Technology, New Delhi 110086, India
[3]Graduate School, Computer Science, Duy Tan University, Da Nang 550000, Vietnam
[4]Department of Computer and Information Technology, Babasaheb Bhimrao Ambedkar University, Lucknow 226025, India
[5]Department of Information Engineering, University of Padova, I-35131 Padua, Italy

Corresponding author: Nhu Gia Nguyen (nguyengianhu@duytan.edu.vn)

This work was supported in part by the Duy Tan University. The authors would like to thank the reviewers in advance for their comments and suggestions.

ABSTRACT In every aspect of human life, sound plays an important role. From personal security to critical surveillance, sound is a key element to develop the automated systems for these fields. Few systems are already in the market, but their efficiency is a point of concern for their implementation in real-life scenarios. The learning capabilities of the deep learning architectures can be used to develop the sound classification systems to overcome efficiency issues of the traditional systems. Our aim, in this paper, is to use the deep learning networks for classifying the environmental sounds based on the generated spectrograms of these sounds. We used the spectrogram images of environmental sounds to train the convolutional neural network (CNN) and the tensor deep stacking network (TDSN). We used two datasets for our experiment: ESC-10 and ESC-50. Both systems were trained on these datasets, and the achieved accuracy was 77% and 49% in CNN and 56% in TDSN trained on the ESC-10. From this experiment, it is concluded that the proposed approach for sound classification using the spectrogram images of sounds can be efficiently used to develop the sound classification and recognition systems.

INDEX TERMS Deep learning, convolutional neural network, tensor deep stacking networks, spectrograms.

I. INTRODUCTION

In recent years, research on automatic sound recognition has gained momentum and has been used in multidisciplinary fields like multimedia [1], bioacoustics monitoring [2], intruder detection in wildlife areas [3], audio surveillance [4] and environmental sounds [5]. Sound recognition problem consists of three different stages as pre-processing of signals, extraction of specific features and their classification. Signal pre-processing divides the input signal to different segments which used for extracting related features. Feature extraction reduces the size of data and represent the complex data as feature vectors. Crossing rate, pitch and frame features used in speech recognition applications were classified using various classifiers like decision trees, random forest and k nearest neighbor. Spectrogram image features (SIF), Stabilized auditory image (SAI) and Linear prediction coefficients (LPC) are used widely in recent years. Moreover, usage of different machine learning and soft computing techniques like Hidden and Gaussian mixture model, random forest, multi-layer perceptron and emerging deep learning networks in sound recognition system resulted in performance enhancement of sound recognition and classification systems.

In recent years SIF generates sound waves which provides more accurate results in noisy conditions. These sound waves are made up of high pressure and low-pressure regions moving through a medium. Such high- and low-pressure regions forms a specific type of pattern to every distinguish sound. These waves have few characteristics like wavelength, frequency, wave speed and time periods [6]. These characteristics are used to classify the sounds into different categories like humans do. As shown in Fig. 1, a spectrogram is a way to visualize the frequency spectrum of the sound wave. In simple words, it is a photograph of the frequency spectrum present in the sound wave [7].

The generated spectrogram of the sound signal is infrequent so that noise intensity is found in lower region and strong components are found in higher region of the generated spectrogram. The generated spectrogram images can be used together with various machine learning classifiers. In the study of Sun *et al.* [8] they proposed an integrated

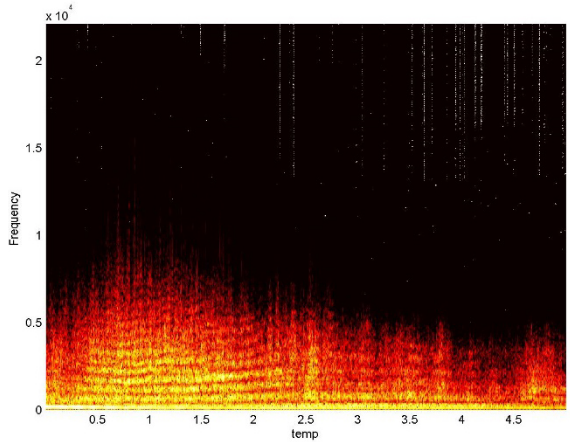

FIGURE 1. Generated spectrogram of a sound wave.

deep learning autoencoder based technique with extreme machine learning models which detects approaches that integrate extreme learning for detection of abstract signal representations using unsupervised learning. Baig *et al.* [9] proposed an Ada-boosting-based method which uses non-linear activation functions having single layer multi perceptron model. Various emerging applications like head and pose estimation utilizes unsupervised autoencoder deep networks to identify search driven engine and associated non-linear mapping. Stacked discriminative sparse autoencoder used to provide semantic granular level representations of the satellite images. A survey on Deep learning models was prepared by Liu *et al.* [10], which demonstrated that Convolutional Neural Network (CNN) outperformed other models in images and video data. CNNs are used for object detection in high resolution remote sensing images [11] but they not able to address object rotation problems, to overcome this rotation invariant CNN had been proposed for object detection in sensory images. As a spectrogram image is the visual representation of the frequency spectrum of a signal, deep learning methods used to perform feature extraction and classification from spectrogram images. Sound signals are less frequent, weak locality and generate different pattern representations in spectrogram. However, CNN is gaining popularity in computer vision and audio processing which are insensitive to the pattern position on the generated spectrogram image and recognized as suitable technique for classifying spectrogram image features. The first framework for CNN was built in the early 90s. LeNet-5 was the first Convolutional Neural Network developed to classify handwritten digits [12]. The performance of LeNet-5 was much better than the existing techniques at that time [13], [14]. The first layer in CNN is a convolutional layer, which tries to learn the underlying features of the image. The next layer is pooling layer which tries to reduce the dimensionality of the feature map. The pooling layer gets the feature map from the convolutional layer. As shown in Fig. 2, there can be multiple sets of the convolutional layers and pooling layers based on the complexity of the dataset. The last layer in a convolutional neural network is the classification or prediction layer. The success of the convolutional neural network is because of three important properties [15]. These are Local Receptive Fields; Shared Weight and Spatial Sub-sampling. Local receptive fields mean the response of a neuron is influenced by a specific 2D portion of an image. Shared weights are the plus points for the convolutional neural network as with these weights the overall number of parameters in the network is reduced [16]. Sub-sampling is used to reduce the resolution of the feature map. This solves the problem of distortion and shifts in the final output. In proposed approach, to recognize sound event in different frequency ranges which are insensitive to spectrogram images and provide more accurate image positioning Tensor Deep Stack network is considered. In recent times, there has been vast need for effective feature extraction in many fields. For example: In IoT, a paper uses the effective feature with the intention of classifying big data by means of social IoT [17]. Other applications for IoT is presented Keswania *et al.* [18] and Lakshmanaprabhu *et al.* [19]. In e-commerce market, a paper uses ideal features for ranking analysis of online customer product reviews using opinion mining with clustering [20]. The hierarchical software usability model has been designed using fuzzy expert system [21]–[23] to predict the usability of software development life cycle models Gupta and Ahlawat [24] and live auction portal [25], [26]. The datasets for SDLC and Live auction have been discussed by Gupta and Khanna [27]. Various Bio inspired algorithms have been used for software usability models [28] like modified binary bat algorithm, modified crow search algorithm [29], modified whale optimization algorithm [30]. Optimal features have been selected for thyroid disease [31].

Genetic Algorithm (GA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) are widely known meta-heuristic optimization techniques for feature selection. The Grey Wolf Optimizer (GWO) is a recent algorithm, which simulates the grey wolves leadership and hunting manner in nature. Classification of protein structure using improved grey wolf optimization is presented in [32]. In Particle Swarm Optimization (PSO), the particles are divided into swarms that interact with each other. Some other evolutionary algorithms are Crow Search Algorithm (CSA) is optimized for the Parkinson's disease diagnosis at an early stage [33], [34]. For example, the paper (Gupta *et al.* [35] presented the early diagnosis of Parkinson's disease by using cuttlefish algorithm and Tiwari *et al.* [36] presents detection of blood cell type using deep learning [37]. Tiwari and Melucci [50]–[52] and Di Buccio [53] proposed a classification model inspired by quantum mechanics which can provide high recall and precision depend upon the need.

The Tensor Deep Stacking Network is an extension to the Deep stacking network. These architectures are the sub-classes of Deep Generative Architectures [38]. In these graphical models, the modules are stacked over one another to reach the final prediction. Sometimes the original input vector is also concatenated with the intermediate output of the hidden

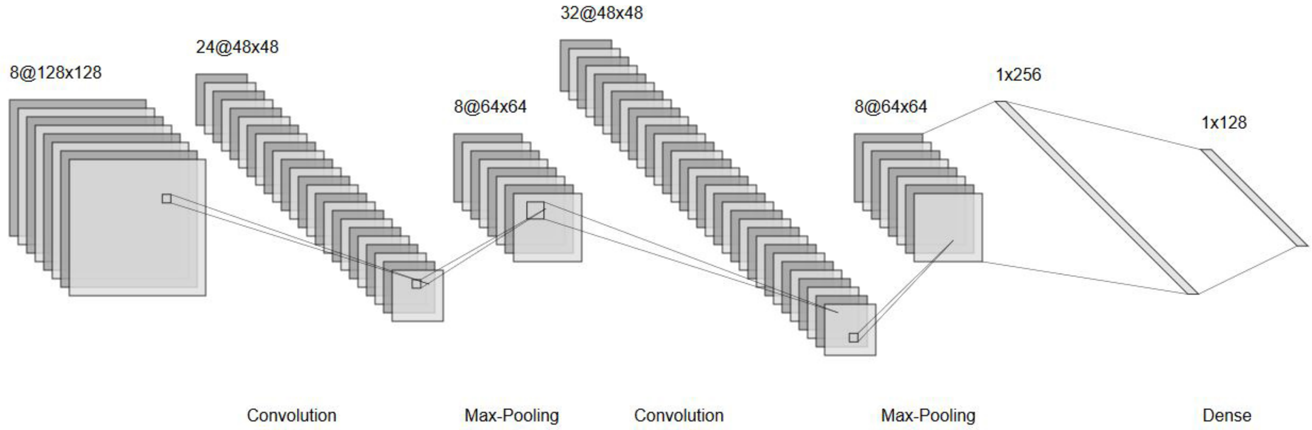

FIGURE 2. **Convolutional neural network; as shown there are two layers of convolutional and pooling layer and a final dense layer.**

layer to achieve more accuracy than the previous layer. Deep stacking network is different from other architectures because the here instead of using gradient descent approach, it works on the principle of mean square error between the current module's prediction and final prediction value [39].

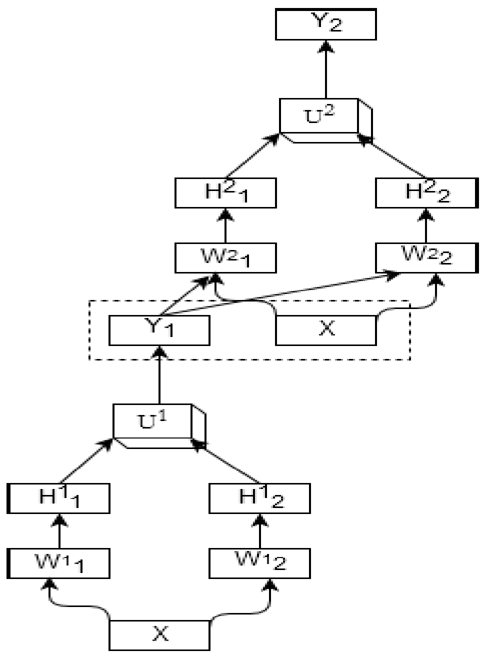

FIGURE 3. **Tensor deep stacking network with two stacked modules.**

As shown in Fig. 3, Tensor-Deep Stacking Network is similar to Deep Stacking Network but instead of having sequential hidden layers in each module, it has two parallel hidden layers in each module. These two parallel hidden layer units will provide an ability to capture the higher order feature interaction through the use of cross products. In, tensor notation the operation will be:

$$y = \mu \left(h_{(1)}, h_{(2)}\right) \cong (\mu \times_1 h_1) \times_2 h_2 \tag{1}$$

Here $\times_i$ denotes the multiplication of respective hidden layer with the i^{th} dimension of the tensor μ of 3rd order [40].

X represents the input vector, W_i^j represents the weight from input to i^{th} hidden layer of j^{th} block of the architecture, H_i^j represents the i^{th} hidden layer of j^{th} block, U^jrepresents the 3rd order weight tensor to combine output of two hidden layer for final prediction.

These two parallel hidden layers (H1,1 and H1,2) will produce two different representations of the input data and a third order tensor (U) in each module is used to produce bilinear mapping of these representations to give a prediction for each module [41]. The concatenation of original input vector X with prediction Y(1) of current layer will guarantee a better generalization in next layer prediction. In the proposed work, after conversion of spectrograms into quantized images, classification performance over different sound categories are compared with CNN and TDSN. These enhancement in proposed deep learning approaches provide better performance on ESC10 and ESC50 dataset in comparison to other methods proposed by Piczak [42] in different frequency domain conditions. The remaining part of the articles are organized as following. Materials and Experiment including hardware details are mentioned in Section 2; Results and discussion is given in Section 3; Section 4 describes the conclusion and future work.

II. MATERIAL AND EXPERIMENT

A. DATASETS

Datasets used in this experiment are much different from other audio datasets available. Here in this, we used environmental sound datasets, not speech datasets. The environmental sound datasets are very limited which is a huge problem to develop a good system for sound classification. We used two available datasets ESC-10 AND ESC-50 for this experiment [43]. The ESC-10 dataset contains total 400 environmental sound recordings from 10 categories. These categories are; dog barking, firecrackers, rain, rooster, baby cries, sneezing, sea waves, chainsaw, helicopter and clock sound.

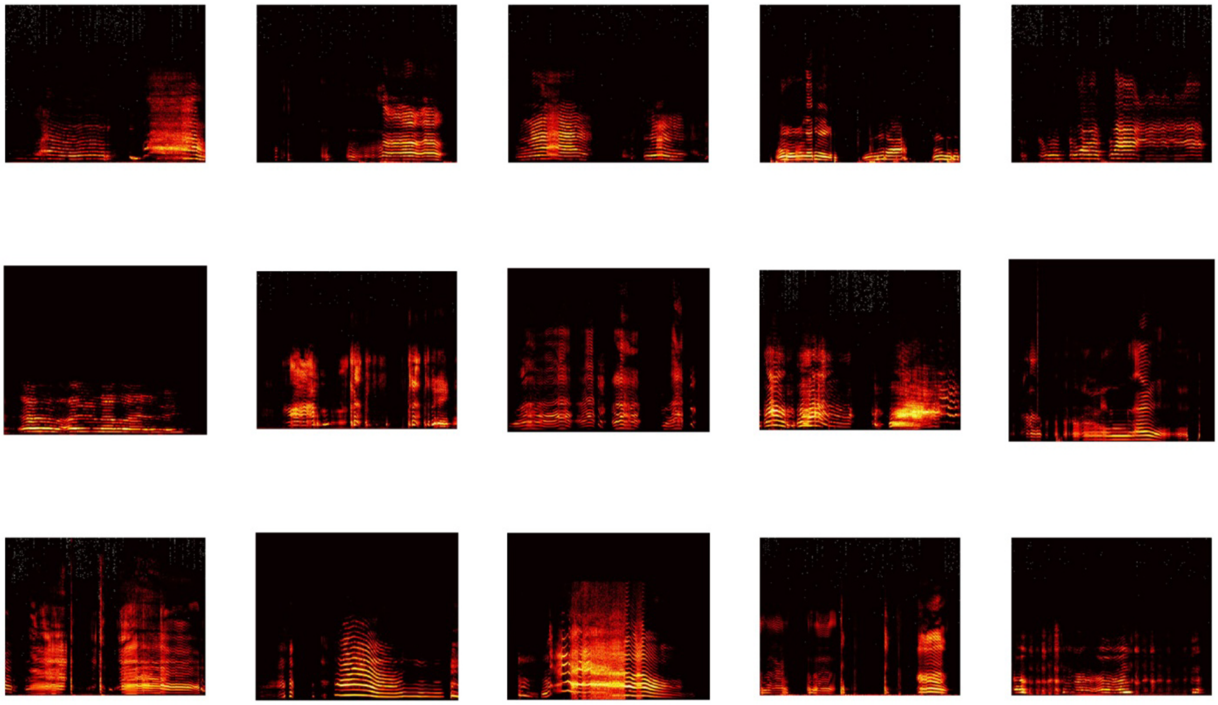

FIGURE 4. Spectrograms for baby crying class.

The ESC-50 dataset is more complex dataset than the ESC-10 dataset. It contains 2000 environmental sound recordings of 50 categories. This dataset is prearranged in 5 folds for cross-validation.

B. HARDWARE USED FOR THIS EXPERIMENT

For this experiment, we used Asus ROG Zephyrus GX501 laptop. Complete specifications of this system are as follow; Processor used is Intel Core i7, Graphics card in this system is Nvidia GeForce GTX 1080 which has 8GB GDDR5X VRAM, total RAM of this system is16GB.

C. SOFTWARE USED

Various software, API and libraries were used in this experiment to build and train convolutional neural network and Tensor deep stacking network.

MATLAB: The main task of this work was to generate spectrograms of sounds present in datasets. This was done by using the built-in function 'spectrogram ()' of the MATLAB to generate a spectrogram of the audio signal. These generated spectrograms were saved using the 'saveas (gcf, 'name'. format)' function over a loop iteration equal to the number of samples present in the dataset. Spectrograms generated by this procedure for baby crying class are shown in Fig. 4.

1) ANACONDA

It is an open source distribution for python which contains a number of machine learning packages. It is very easy to create a virtual environment in Windows using this software.

2) TDSN TOOLKIT

It is an open source toolkit for implementing tensor deep stacking network [44]. It contains almost all libraries required to run this toolkit. This toolkit provides a number of functions those are used to train and test the tensor deep stacking network.

3) KERAS

It is an API specifically designed to support the deep neural network architectures [45]. In this experiment, we used Keras on top of TensorFlow. Keras is used in the experiment to build the convolutional neural network. Keras contains a number of activations and optimizers those can be used very easily in the model. Apart from these various other libraries like Numpy and Scikit-Learn [46] were also used in this setup.

D. EXPERIMENT

1) CONVOLUTIONAL NEURAL NETWORK

Using the Keras library running over TensorFlow, we built a sequential model with the following specification. The convolutional neural network here was a 2-layer deep architecture with a final fully connected layer and an output prediction layer as shown in Fig. 5. The code snippet of proposed implementation method is shown in Fig. 6. Complete workflow procedure of CNN is described in Fig. 7. The first convolutional layer contained 32 filters of 3×3 size with ReLU activation [47]. Fig. 10 shows 32 feature maps generated by an intermediate layer. Max pooling of size 2×2 was used to reduce the dimensionality of the data and filter

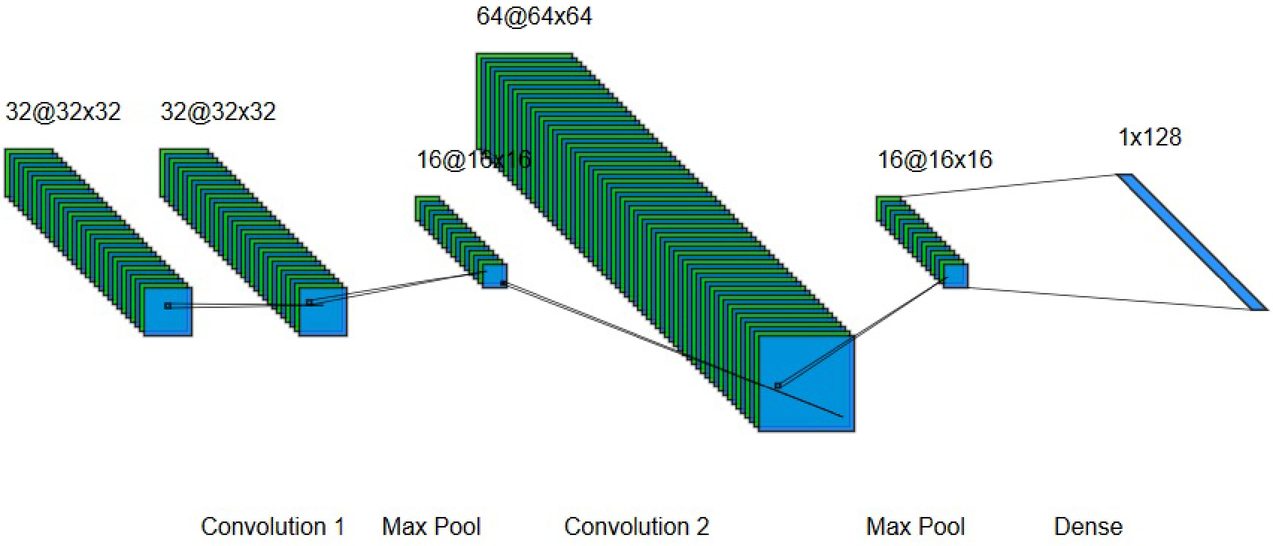

FIGURE 5. Schematic CNN.

```
# Defining the model
SequentialModel (Convolution2D, Pooling, Dense, Activation)
        Input image shape for different convolutions
        Add 2D convolutions with 32 filters and size 3*3 until filter reaches 64
        Perform sigmoid and relu activations till dense reached 64 layers.
        Involve mask of size 2*2 through max pooling operations until dense layer not reached
        Hyper-tune model with drop rate of 0.5 to avoid overfitting
        Add fully connected layer with Dense classes
        Perform Softmax activation through SGD optimizer to avoid model decay
        Define the accuracy metric and compile the model
# Viewing model_configuration
SequentialModel (get_config, get_weights)
        View generated model configuration
        View model input and output shape
        Assign the weights to individual layers
        Model is ready to train once weights get updated
# Training
        Perform model fitting using batch size and epochs
        Validation of data also tested till training completed
#Visualizing losses and accuracy
        Visualize model loss and validation loss with obtained accuracy until epochs reaches 100
```

FIGURE 6. Pseudocode of proposed work.

out the unnecessary data in the feature maps. The next layer contained the 64 filters of 3×3 size with ReLU activation. Max pooling of size 2×2 was again used here in second hidden layer. To avoid the over fitting of data, Dropout learning with 50% dropout probability was used to create high co-adaption among hidden layer units [48]. Before passing the information to fully connected layer, we flattened the features to form a one-dimensional feature vector. A fully connected layer with 128 ReLU activations is used to process the features vector. The prediction layer was a softmax layer to predict the final class. Network was trained using Keras implementation of rmsprop instead of mini batch stochastic gradient decent approach. The loss function used in the model was categorical cross entropy [49].

Spectrograms were generated from the both datasets and were resized to 180×180 pixels to reduce the system load

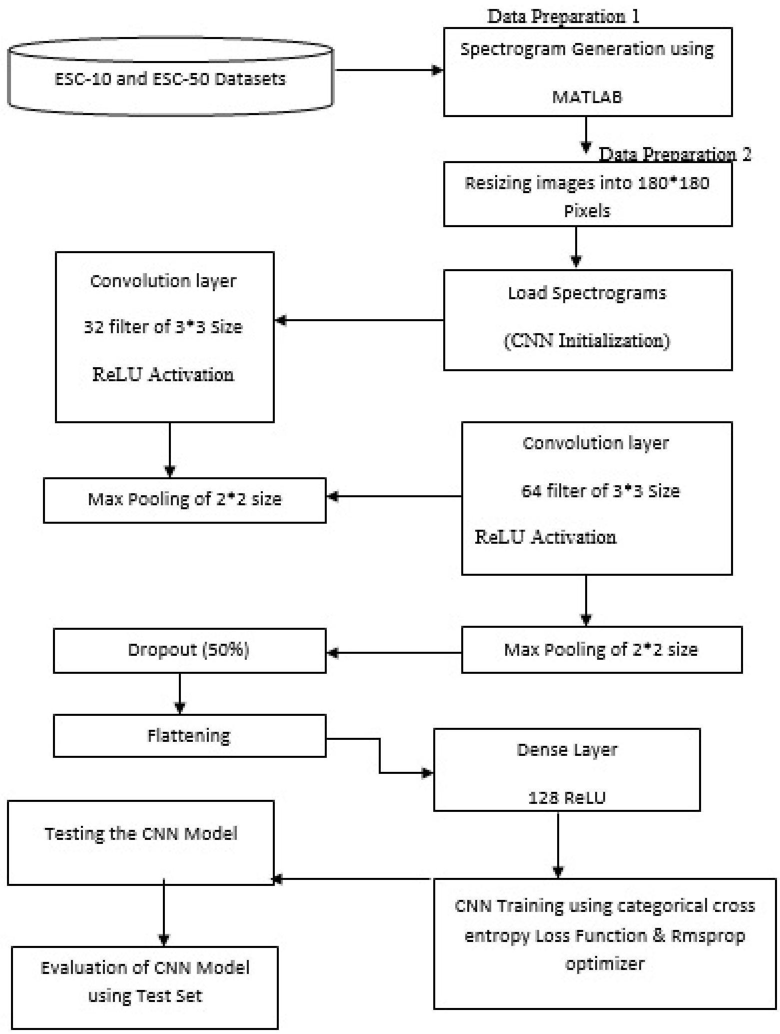

FIGURE 7. Workflow procedure.

and speed up the training process. These resized imaged were then used to train the CNN network.

2) TENSOR DEEP STACKING NETWORK

TDSN network was designed for this experiment with the help of TDSN toolkit. This toolkit is very easy to operate because almost all the operations are built-in in the toolkit.

- This network training was divided into three subgroups. First, we trained the network using 2 stacked blocks, then with 3 stacked blocks and finally with four stacked blocks.
- The number of hidden units in parallel layers were 90 units in each.
- Before given the data for trained we used Scikit-Learn using Keras to flatten the spectrogram images into 1D feature vectors.
- The feature vectors were further converted into dense binary file format and the target matrix was converted into sparse binary file format.

III. RESULTS AND DISCUSSIONS

In this section, the performance of proposed CNN and TDSN based sound event classification system is compared with other reported systems and effectiveness of proposed approach is observed which further evaluated with reference to different parameters. CNN trained on both datasets. Table 1 shows the results comparison of some previous studies conducted using spectrogram image features for sound event classification with proposed CNN and TDSN approach. It is seen from Table 1 that our approach (CNN) provided 38.9%, 37.4%, 34%, 32.2%, 29.76% and TDSN provided 21.9%, 22.4%, 17%, 15.2% and 12.76% better performance in comparison to other systems.

CNN Training is done with two activations functions; Tanh and ReLU. Tanh activation function is similar to logistic sigmoid activation function but its performance is better. The main advantage of using Tanh function was due to its capability to map negative inputs to a strongly negative region and all the nearby zero inputs are mapped near to zero.

TABLE 1. Performance comparison of proposed approach with other systems.

Spectrogram driven Sound Systems	Techniques/Architectures Used	Performance (Accuracy %)
MFCC-SVM	Support Vector Machine (SVM)	34.1 %
MPEG-7	Decision Trees	33.6 %
Gabor	Random Forest	39.0%
GTCC	K-Nearest Neigbor	40.8%
MFCC-MP	Multi layer Perceptron	43.24%
CNN (Proposed)	**Convolutional Neural Network**	**73%**
TDSN (Proposed)	**Tensor Deep Stack Network**	**56%**

ReLU activation function performed best in the experiment because of its half-rectified nature. One disadvantage of using ReLU is that it maps all the negative inputs to zero which cause a hindrance in training the network.

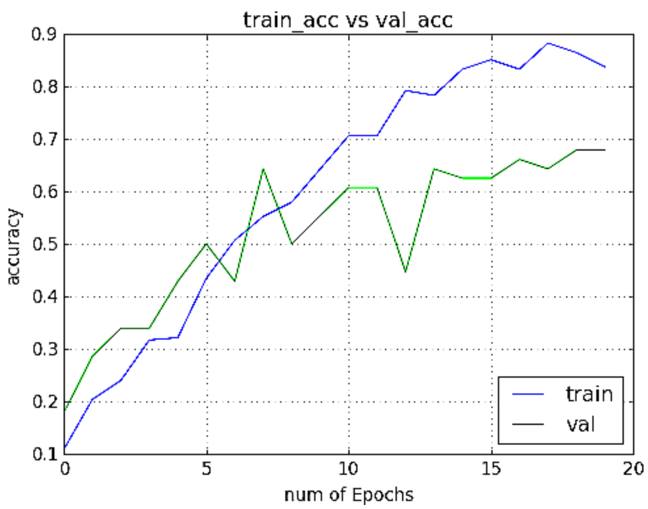

FIGURE 8. Training accuracy vs validation accuracy for ReLU.

As shown in Fig. 8, the training accuracy is increasing with number of epochs going up. The system was tested with multiple segments of the datasets. CNN was performing better with more data provided in the training. Curve shown in Figure 8 was plotted using 450 samples drawn from the ESC-50 data set with training stopped after 20 epochs.

With more data provided to the training process in CNN, the training and validation loss curve shows a promising results. As shown in Fig. 9, training loss was decreased to below 0.5 when trained with 450 samples as compared to 200 samples drawn from ESC-50 dataset. The reason for this performance is the ability of CNN architecture to learn more features from the large datasets as compared to small sample size.

FIGURE 9. Training loss vs validation loss for ReLU.

TABLE 2. CNN performance with different filter size.

Iteration / Batch Size	32	64	128	Training Time
500	60.23	63.25	65.88	50 min 16 sec
1000	62.12	67.82	68.06	1 hr. 58 sec
3000	65.95	69.22	71.46	3 hr. 10 sec
50000	**69.17**	**73. 47**	**77.00**	**5 hr. 43 sec**

TABLE 3. TDSN performance with different filter size.

Iteration/ Batch Size	32	64	128	Training Time
500	37.88	38.62	40.90	48 min 05 sec
1000	38.54	42.39	44.18	1 hr. 26 sec
3000	45.55	47.10	49.40	3 hr. 22 sec
50000	**49.03**	**52.84**	**56**	**5 hr. 52 sec**

Fig. 11 shows the testing accuracy when CNN was trained on ESC-10 dataset up to 100 epochs and 150 epochs. As compared to old classification techniques like HMM and ANN models for sound classification, CNN performed better to learn features and predicting the final class. One disadvantage of large dataset is over-fitting of data which was solved in this experiment using dropout with 50% probability. Drop Connect might be used instead of dropout to handle the over fitting in better way. Drop connect is considered as a generalization of dropout because it generates more possible models as compared to dropout. Dropout dropped the output of randomly selected units to zero but drop connect sets the selected weights to zero. There are always more weights as compared to number of nodes in the network. Therefore, drop connect provides more chances to find a better model for efficient training.

Fig. 12 shows the testing accuracy of CNN model trained on ESC-50 dataset and number of epochs are 200. ReLU activation function performed better even with large dataset of 2000 spectrogram images of 180 × 180 pixels.

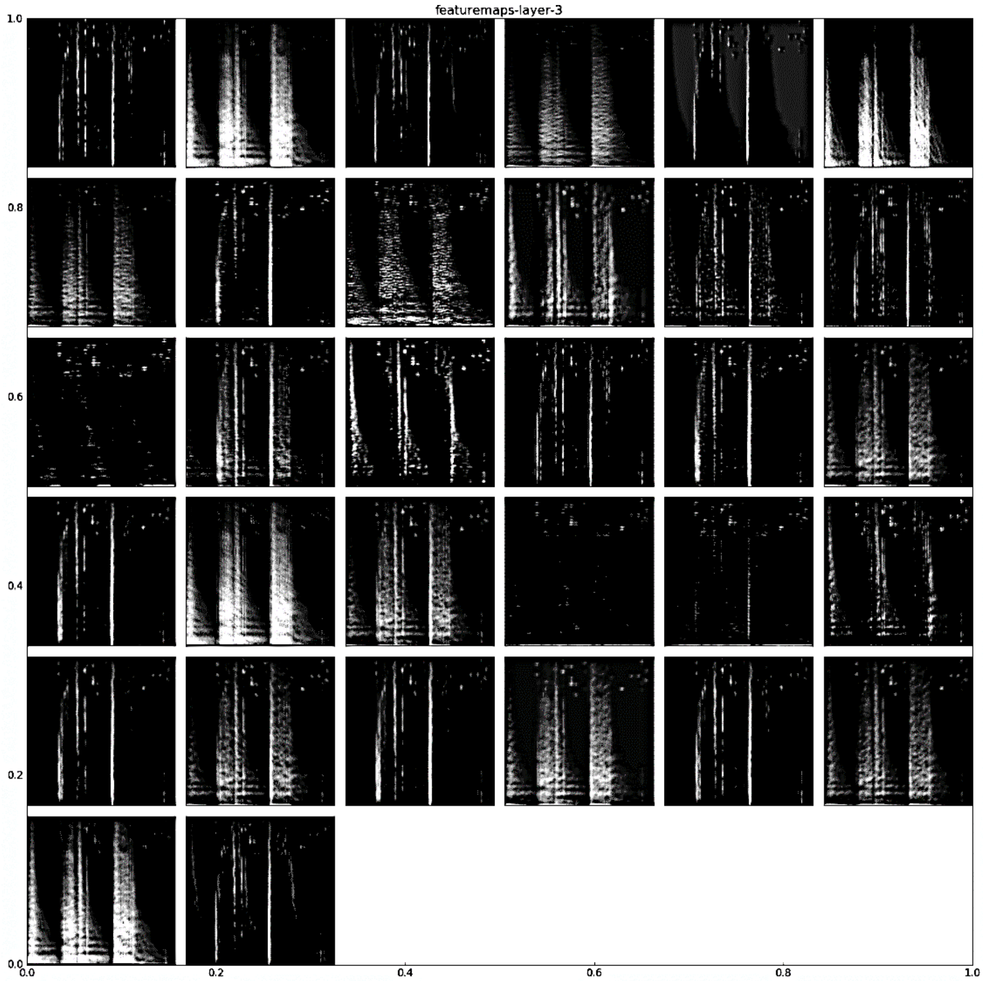

FIGURE 10. Feature maps generated by intermediate layer of CNN.

TABLE 4. Overall Comparison of our approaches with base implementations.

Technique	Dataset Used	Accuracy
CNN (K. Piczak, 2015)	ESC-10	73%
CNN (Our Approach)	ESC-10	77%
CNN (K. Piczak, 2015)	ESC-50	44%
CNN (Our Approach)	ESC-50	49%
TDSN (B. Hutchinson et.al, 2013)	ESC-10	53%
TDSN (Our Approach)	ESC-10	56%

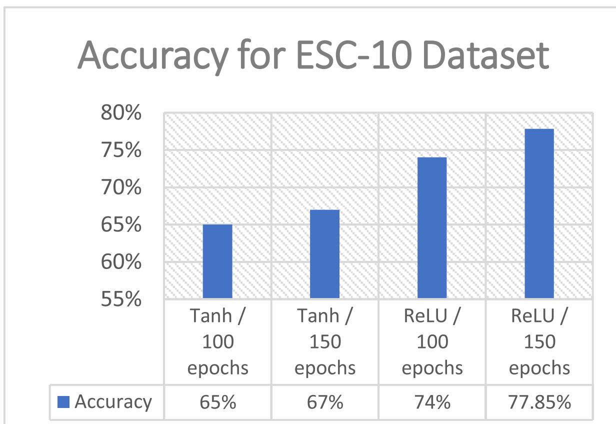

FIGURE 11. Testing accuracy of CNN with ESC-10 dataset.

Fig. 13 shows the testing accuracy of TDSN model trained on ESC-10 dataset. Due to the lack of resources TDSN was not tested on ESC-50 dataset. Even on ESC-10 dataset this approach of sound classification using Spectrograms of sound waves outperformed the original implementation for speech classification (Hutchinson *et al.*, 2013). In the baseline implementation of TDSN as shown in Table 4, data given to the network was directly compressed into sparse and dense binary formats. In this implementation, the images were flattened using Sklearn library to convert the image feature map to one dimensional feature map.

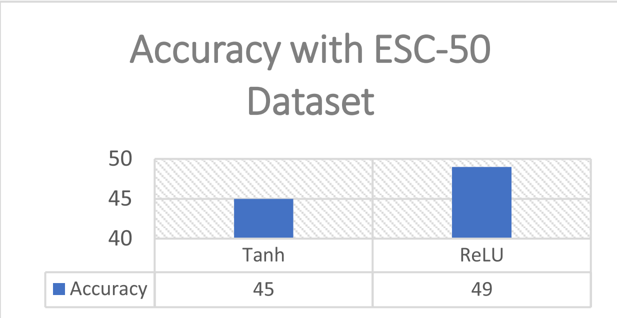

FIGURE 12. Testing accuracy of CNN with ESC-50 dataset.

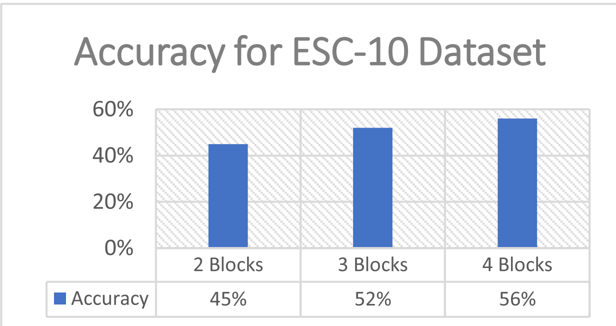

FIGURE 13. Testing accuracy of TDSN with ESC-10 dataset.

Table 2 and 3 demonstrated the performance and training time required by CNN and TDSN applied to ESC-10 dataset with different batch sizes and iterations numbers respectively.

IV. CONCLUSION

The goal of this paper was to evaluate the use of CNN and TDSN architectures to classify the sound signals using the spectrograms of the sound spectrum. Convolutional Neural Network is mostly applied to image classification problems. This paper shows that these deep neural architectures can be applied to sound classification. This approach using CNN and TDSN for sound classification using spectrograms reduced the number of trainable parameters as compared to direct sound classification. In the experimental tests, which conducted using CNN and TDSN, we obtained classification accuracy success rate of 77%, 49% and 56% compared to other existing methods. From the experiment, this can be evaluated that this approach shows promising results for the development of sound classification system in the critical areas. The possible question for future work is whether tensor deep stacking network could be efficiently used with CNN to classify the sound signals. The power of tensors can be utilized to train the network on high definition images instead of compressed images.

CONFLICT OF INTEREST

The authors do not have financial and personal relationships with other people or organizations that could inappropriately influence (bias) their work.

REFERENCES

[1] E. Wold, T. Blum, D. Keislar, and J. Wheaten, "Content-based classification, search, and retrieval of audio," *IEEE Multimedia*, vol. 3, no. 3, pp. 27–36, Jun. 1996.

[2] F. Weninger and B. Schuller, "Audio recognition in the wild: Static and dynamic classification on a real-world database of animal vocalizations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 337–340.

[3] M. V. Ghiurcau, C. Rusu, R. C. Bilcu, and J. Astola, "Audio based solutions for detecting intruders in wild areas," *Signal Process.*, vol. 92, no. 3, pp. 829–840, 2012.

[4] A. Rabaoui, M. Davy, S. Rossignol, and N. Ellouze, "Using one-class SVMs and wavelets for audio surveillance," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 4, pp. 763–775, Dec. 2008.

[5] S. Chu, S. Narayanan, and C.-C. J. Kuo, "Environmental sound recognition with time–frequency audio features," *IEEE Trans. Audio, Speech, Language Process.*, vol. 17, no. 6, pp. 1142–1158, Aug. 2009.

[6] (2017). *Sound Classification*. [Online]. Available: http://www.paroc.com/knowhow/sound/sound-classification

[7] R. A. Altes, "Detection, estimation, and classification with spectrograms," *J. Acoust. Soc. Amer.*, vol. 67, no. 4, pp. 1232–1246, 1980.

[8] K. Sun, J. Zhang, C. Zhang, and J. Hu, "Generalized extreme learning machine autoencoder and a new deep neural network," *Neurocomputing*, vol. 230, pp. 374–381, Mar. 2017.

[9] M. M. Baig, M. M. Awais, and E.-S. M. El-Alfy, "AdaBoost-based artificial neural network learning," *Neurocomputing*, vol. 248, pp. 120–126, Jul. 2017.

[10] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[11] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.

[12] Y. LeCun, B. Boser, J. S. Denker, and D. Henderson, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 16–19.

[14] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.

[15] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997.

[16] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *Handbook Brain Theory Neural Netw.*, vol. 3361, no. 10, p. 1995, 1995.

[17] R. R. Rodrigues, J. J. P. C. Rodrigues, M. A. A. da Cruz, A. Khanna, and D. Gupta, "An IoT-based automated shower system for smart homes," in *Proc. Int. Conf. Adv. Comput., Commun. Inform. (ICACCI)*, Sep. 2018, pp. 254–258.

[18] B. Keswani *et al.*, "Adapting weather conditions based IoT enabled smart irrigation technique in precision agriculture mechanisms," *Neural Comput. Appl.*, vol. 20, pp. 1–16, Sep. 2018.

[19] S. K. Lakshmanaprabu *et al.*, "Effective features to classify big data using social Internet of Things," *IEEE Access*, vol. 6, pp. 24196–24204, 2018.

[20] S. K. Lakshmanaprabu *et al.*, "Ranking analysis for online customer reviews of products using opinion mining with clustering," *Complexity*, vol. 20, Jun. 2018, Art. no. 3569351.

[21] D. Gupta, J. J. P. C. Rodrigues, S. Sundaram, A. Khanna, V. Korotaev, and V. H. C. de Albuquerque, "Usability feature extraction using modified crow search algorithm: A novel approach," *Neural Comput. Appl.*, pp. 1–11, Aug. 2018, doi: 10.1007/s00521-018-3688-6.

[22] D. Gupta and A. K. Ahlawat, "Taxonomy of GUM and usability prediction using GUM multistage fuzzy expert system," *Int. Arab J. Inf. Technol.*, vol. 16, no. 3, p. 2019, 2017.

[23] D. Gupta, A. Ahlawat, and K. Sagar, "A critical analysis of a hierarchy based usability model," in *Proc. Int. Conf. Contemp. Comput. Inform. (ICI)*, Nov. 2014, pp. 255–260.

[24] D. Gupta and A. K. Ahlawat, "Usability evaluation of live auction portal," *Int. J. Control Theory Appl.*, vol. 9, no. 40, 2016.

[25] D. Gupta, A. K. Ahlawat, and K. Sagar, "Usability prediction & ranking of SDLC models using fuzzy hierarchical usability model," *Open Eng.*, vol. 7, no. 1, pp. 161–168, 2017.
[26] D. Gupta and A. Khanna, "Software usability datasets," *Int. J. Pure Appl. Math.*, vol. 117, no. 15, pp. 1001–1014, 2017.
[27] D. Gupta and A. K. Ahlawat, "Usability determination using multistage fuzzy system," *Procedia Comput. Sci.*, vol. 78, no. 3, pp. 263—270, 2016, doi: 10.1016/j.procs.2016.02.042.
[28] D. Gupta and A. Ahlawat, "Usability prediction of live auction using multistage fuzzy system," *Int. J. Artif. Intell. Appl. Smart Devices*, vol. 5, no. 1, pp. 11–20, 2017.
[29] D. Gupta, S. Sundaram, A. Khanna, A. E. Hassanien, and V. H. C. de Albuquerque, "Improved diagnosis of Parkinson's disease using optimized crow search algorithm," *Comput. Elect. Eng.*, vol. 68, pp. 412–424, May 2018.
[30] R. Jain, D. Gupta, and A. Khanna, "Usability feature optimization using MWOA," in *Proc. Int. Conf. Innov. Comput. Commun. (ICICC)*, vol. 2, 2018, pp. 453–462.
[31] K. Shankar, S. K. Lakshmanaprabu, D. Gupta, A. Maseleno, and V. H. C. de Albuquerque, "Optimal feature-based multi-kernel SVM approach for thyroid disease classification," *J. Supercomput.*, pp. 1–16, Jul. 2018, doi: 10.1007/s11227-018-2469-4.
[32] P. Sharma *et al.*, "The health of things for classification of protein structure using improved grey wolf optimization," *J. Supercomput.*, pp. 1–16, Oct. 2018.
[33] A. Patnaik and D. Gupta, "Unique identification system," *Int. J. Comput. Appl.*, vol. 7, no. 5, pp. 46–51, 2010.
[34] D. Gupta and K. Sagar, "Remote file synchronization single-round algorithms," *Int. J. Comput. Appl.*, vol. 4, no. 1, pp. 32–36, 2010.
[35] D. Gupta *et al.*, "Optimized cuttlefish algorithm for diagnosis of Parkinson's disease," *Cognit. Syst. Res.*, vol. 52, pp. 36–48, Dec. 2018.
[36] P. Tiwari *et al.*, "Detection of subtype blood cells using deep learning," *Cogn. Syst. Res.*, vol. 52, pp. 1036–1044, Dec. 2018.
[37] D. Gupta and A. K. Ahlawat, "Usability feature selection via MBBAT: A novel approach," *J. Comput. Sci.*, vol. 23, pp. 195–203, Nov. 2017.
[38] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
[39] L. Deng, X. He, and J. Gao, "Deep stacking networks for information retrieval," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 3153–3157.
[40] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[41] B. Hutchinson, L. Deng, and D. Yu, "Tensor deep stacking networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1944–1957, Aug. 2013.
[42] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *Proc. IEEE 25th Int. Workshop Mach. Learn. Signal Process.*, Sep. 2015, pp. 1–6.
[43] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 1015–1018.
[44] D. Palzer and B. Hutchinson, "The tensor deep stacking network toolkit," *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–5.
[45] (2018). *Keras Documentation*. [Online]. Available: https://keras.io/
[46] (2001). *SciPy: Open Source Scientific Tools for Python*. [Online]. Available: https://www.scipy.org/
[47] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Workshop Deep Learn. Audio, Speech, Lang. Process. (ICML)*, 2013, p. 3.
[48] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. (2012). "Improving neural networks by preventing co-adaptation of feature detectors." [Online]. Available: https://arxiv.org/abs/1207.0580
[49] Z. Kons and O. Toledo-Ronen, "Audio event classification using deep neural networks," in *Proc. INTERSPEECH*, 2013, pp. 1482–1486.
[50] P. Tiwari and M. Melucci, "Towards a quantum-inspired framework for binary classification," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 1815–1818.
[51] P. Tiwari and M. Melucci. (2018). "Multi-class classification model inspired by quantum detection theory." [Online]. Available: https://arxiv.org/abs/1810.04491
[52] P. Tiwari and M. Melucci, "Binary classifier inspired by quantum theory," in *Proc. AAAI*, Honolulu, HI, USA, 2019, pp. 214–216.
[53] B. E. Di, Q. Li, M. Melucci, and P. Tiwari, "Binary classification model inspired from quantum detection theory," in *Proc. ACM SIGIR Int. Conf. Theory Inf. Retr.*, Sep. 2018, pp. 187–190.

ADITYA KHAMPARIA received the Ph.D. degree in computer science from Lovely Professional University, India. He has been serving as an Academician and a Research Person for the past five years. He is currently an Assistant Professor of computer science with Lovely Professional University, India. He was invited as a Faculty Resource Person/Session Chair/Reviewer/TPC Member for different FDP, conferences, and journals. He has published more than 35 scientific research publications in reputed international/national journals and conferences, which are indexed in various international databases. His research interests include machine learning, soft computing, educational technologies, the IoT, semantic web, and ontologies. He received the research excellence award, in 2016, 2017, and 2018, from Lovely Professional University for his research contribution during the academic year. He is a member of the CSI, IET, ISTE, IAENG, ACM, and IACSIT. He is also a Reviewer and a member of various renowned national and international conferences/journals.

DEEPAK GUPTA received the B.Tech. degree in IT from GGSIP University, New Delhi, India, the M.E. degree in CTA from the Delhi College of Engineering, and the Ph.D. degree in CSE from the Dr. A.P.J. Abdul Kalam Technical University. He has qualified in GATE. He is a High Spirited Academician and a Researcher with 12 years of teaching experience and two years in industry. He is currently with the Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology, GGSIP University. He is also a Postdoctoral Research Fellow with the Internet of Things Research Lab, Inatel, Brazil. He has authored/edited 31 books with national/international level publishers, including Elsevier, Springer, Wiley, and Katson. He has published 39 scientific research publications in reputed international journals and conferences, including 16 SCI-indexed journals of the IEEE, Elsevier, Springer, and Wiley. His research interests include human–computer interaction, intelligent data analysis, nature-inspired computing, machine learning, and soft computing. He was invited as a Faculty Resource Person/Session Chair/Reviewer/TPC Member for different FDP, conferences, and journals. He is the Convener and the Organizer of the ICICC Springer Conference Series. He has also started a research unit under the banner of Universal Innovator. He was the Guest Editor of nine special issues, including SCI-indexed journals, of the ASoC (Elsevier), NCAA (Springer), and CAEE (Elsevier). He was also appointed as the Editor-in-Chief of the *OA Journal - Computers*. He is also associated with various professional bodies, including the ISTE, IAENG, IACSIT, SCIEI, ICSES, UACEE, the Internet Society, SMEI, IAOP, and IAOIP.

NHU GIA NGUYEN received the Ph.D. degree in computer science from the Hanoi University of Science, Vietnam National University, Vietnam. He is currently the Dean of the Graduate School, Duy Tan University, Vietnam. He has a total academic teaching experience of 18 years with more than 50 publications in reputed international conferences, journals, and online book chapter contributions (indexed by SCI, SCIE, SSCI, Scopus, and DBLP). His research interests include network communication, security and vulnerability, network performance analysis and simulation, cloud computing, and image processing in biomedical. He is currently an Associate Editor of the *International Journal of Synthetic Emotions*.

ASHISH KHANNA received the B.Tech. and M.Tech. degrees from GGSIPU, New Delhi, in 2004 and 2009, respectively, and the Ph.D. degree from the National Institute of Technology, Kurukshetra. He is a highly qualified individual with about 15 years of rich expertise in teaching, entrepreneurship, and research and development with a specialization in computer science engineering subjects. He has been a part of various seminars, paper presentations, research paper reviews, and conferences, as a Convener and the Session Chair, and the Guest Editor of journals. He has co-authored several books with publication houses and papers in national, international journals, and conferences. He has published many research papers in reputed journals and conferences. He has papers in SCI-indexed and Springer journals. He was a Reviewer of some SCI-indexed journals, including *Cluster Computing* (Springer) and the IEEE conferences. He was the Guest Editor of the IEEE Conference IC3TSN-2017, managing a Special Session on Parallel and Distributed Network-Based Computing Systems. He was the Guest Editor of a Springer Conference at ICDMAI-2018, managing a Special Session on Computational Intelligence for Data Science.

He has displayed a vast success in continuously acquiring new knowledge and applying innovative pedagogies and have always aimed to be an effective educator and have a global outlook which is the need of today. He has co-authored 10 text books and edited some books, including *Distributed Systems*, *Java Programming and Website Development*, *Java Programming*, *Computer Graphics*, *Computer Graphics and Multimedia*, *Computer Networks*, *Computer Networks and Data Communication Networks*, *Success Mantra for IT interviews*, and *Fundamental of Computing*. He has also edited a book with Lambert publication. His research interests include distributed computing, distributed systems, cloud computing, vehicular ad hoc networks, and opportunistic networks. He is also a Reviewer and the Session Chair of the IEEE international Conferences ICCCA 2016 and ICCCA 2017. He has designed the syllabus for cloud computing, java programming, and distributed systems for GGSIPU. He has recently successfully managed Smart India Hackathon at MAIT GGSIPU, in 2017, with teams under him winning prices in their respective events.

Dr. Khanna is associated with some Springer and IEEE conferences, managing the special sessions for them and looking forward for some more challenging tasks.

BABITA PANDEY received the Ph.D. degree in computer science from IIT Varanasi, Varanasi, India. She is serving as an Academician and a Research Person for the past 10 years. She is currently an Assistant Professor with Babasaheb Bhimrao Ambedkar University, India. She has published more than 100 scientific research publications in reputed international/national journals and conferences, which are indexed in various international databases. Her research interests include medical informatics, soft computing, educational technologies, expert systems, and data mining. She was invited as a Faculty Resource Person/Session Chair/Reviewer/TPC Member for different FDP, conferences, and journals. She is also a Reviewer and a member of various renowned national and international conferences/journals.

PRAYAG TIWARI received the M.S. degree from NUST MISIS, Moscow. He is currently pursuing the Ph.D. degree with the University of Padova, Italy. He was a Research Assistant with NUST MISIS. He has teaching and industrial work experience. He is also a Marie Sklodowska-Curie Researcher with the University of Padova. He has several publications in journals, book series, and conferences of the IEEE, ACM, Springer, Elsevier, MDPI, Taylor & Francis, IGI-Global, and so on. His research interests include machine learning, deep learning, quantum machine learning, and information retrieval.

• • •