Soft Computing Fusion with Applications

www.scfa.reapress.com

Soft. Comput. Fusion. Appl. Vol. 1, No. 2 (2024) 99–108.

Paper Type: Original Article

Deep Audio Classifier: an Artificial Neural Network Approach

Abhishek Yadav[1], Abhishek Raj[1,*], Sankalp Anand[1], Vineet Kumar[1], Abhay Kumar[1]

[1] KIIT University; 21052469@kiit.ac.in; 21051025@kiit.ac.in; 2105997@kiit.ac.in; 21051020@kiit.ac.in; 2105937@kiit.ac.in.

Citation:

Received: 26 January 2024 *Revised: 15 March 2024* *Accepted: 02 July 2024*	Yadav, A., Raj, A., Anand, S., Kumar, V., & Kumar, A. (2024). Deep audio classifier: an artificial neural network approach. *Soft computing fusion with applications*, *1*(2), 99-108.

Abstract

This research centers on developing a deep audio classifier by examining several machine learning and deep learning algorithms, such as Support Vector Machines (SVMs), Random Forest (RF), Artificial Neural Networks (ANNs), and Convolutional Neural Networks (CNNs). The models were trained and evaluated using the UrbanSound8K dataset. The objective of this study is to create strong models that can effectively classify intricate urban sound environments. The audio samples went through comprehensive preprocessing steps, including noise reduction, normalization, and trimming to maintain consistent sample duration. Feature extraction was conducted using Mel-Frequency Cepstral Coefficients (MFCCs). The ANN model, which consists of dense layers tailored for feature learning and utilizes softmax activation for multi-class classification, obtained a classification accuracy of 80.20%. The SVM and RF models achieved accuracies of 82.34% and 84.90%, respectively, using linear and ensemble methodologies. The CNN model surpassed the others with an accuracy of 88.45%, showcasing its ability to capture spatial hierarchies and localized patterns within audio data. Model performance differed by class, demonstrating high precision in recognizing specific sounds such as car horns and gunshots.

The research ends with recommendations for future efforts, such as utilizing sophisticated data augmentation methods, investigating hybrid models, and conducting more extensive hyperparameter tuning to enhance classification accuracy and adaptability in practical urban settings.

Keywords: Deep learning, Support vector machine, Random forest, Artificial neural network, Convolutional neural network, Mel-frequency cepstrum coefficients, Librosa.

1 | Introduction

1.1 | Background

In today's urbanized world, understanding and categorizing ambient sounds is essential for applications ranging from smart city [1] initiatives to enhancing user experiences in multimedia systems. Audio

Corresponding Author: 21051025@kiit.ac.in

classification involves identifying and categorizing sounds into predefined classes, enabling machines to interpret and respond to auditory information effectively [2].

1.2 | Problem Statement

Urban environments are characterized by myriad sounds, making accurate classification challenging due to overlapping frequencies, varying sound intensities, and background noise. Developing robust models reliably classifying such diverse audio data is crucial for deploying intelligent systems in real-world scenarios [3].

1.3 | Objectives

To develop and evaluate a deep audio classifier using multiple models, including Artificial Neural Network (ANN) [4], Support Vector Machine (SVM) [5], and Convolutional Neural Network (CNN) [6], trained on the UrbanSound8K dataset to classify ten urban sound classes focusing on performance comparison and future enhancements.

1.4 | Significance of the Study

This research contributes to the field of audio signal processing by providing insights into the application of ANN models [7] for complex sound classification tasks. The findings can inform future developments in environmental sound monitoring, automated surveillance systems, and interactive multimedia applications.

2 | Literature Review

Audio classification has been a subject of extensive research, given its critical applications in environmental monitoring, multimedia systems, and urban planning. Early approaches relied on handcrafted features and traditional machine learning models, such as SVMs and k-Nearest Neighbors (k-NNs) [8], for audio recognition tasks. These methods primarily focused on extracting features like Mel-Frequency Cepstral Coefficients (MFCCs), zero-crossing rates, and spectral roll-off to capture audio signal characteristics. While adequate for basic classification tasks, they struggled with more complex urban soundscapes where overlapping frequencies and noise posed significant challenges.

In recent years, deep learning has revolutionized the field of audio classification [9], [10]. CNNs have been extensively explored due to their ability to learn spatial hierarchies and localized patterns from spectrograms [11]. Studies have demonstrated that CNNs outperform traditional machine learning models in tasks such as music genre classification and environmental sound recognition, achieving higher accuracy by capturing temporal and spectral information [12], [13]. Similarly, ANNs, although less specialized than CNNs for spatial data, have shown promise in scenarios where computational simplicity and interpretability are prioritized.

The UrbanSound8K dataset has become a benchmark in audio classification research, offering a diverse collection of labeled urban sound samples [14]. Researchers have used it to evaluate various architectures, such as Long Short-Term Memory (LSTM) [15] networks for sequential audio data and ensemble models combining CNNs with traditional classifiers. Data preprocessing techniques, including noise reduction, normalization, and MFCC extraction, have been consistently emphasized to enhance model performance by mitigating the effects of background noise and signal variability.

Despite these advancements, there remain challenges in accurately classifying urban sounds, mainly ambient and overlapping noise classes. Recent works highlight the potential of hybrid architectures and transfer learning to address these limitations. Integrating data augmentation and advanced feature extraction methods has also been proposed to improve the generalizability of models.

This study builds upon prior work by comparing the performance of multiple models—ANN, SVM, CNN—on the UrbanSound8K dataset. By leveraging MFCC-based features and rigorous preprocessing, this research aims to contribute to developing more robust and accurate audio classifiers, addressing current gaps in classifying complex urban soundscapes.

3 | Methods

3.1 | Preprocessing Steps

Librosa

Librosa is a Python music and audio analysis package. It provides the building blocks to create music information retrieval systems [16].

In the context of urban sound classification, it can extract features from audio recordings of city sounds, such as traffic noise, and then use those features to train a machine learning model to classify new audio recordings.

Mel-frequency cepstral coefficient

An MFCC comprises a number of coefficients known as MFCCs. They were created using an audio clip's cepstral representation (a nonlinear "spectrum-of-a-spectrum"). The Mel-Frequency Cepstrum (MFC) differs from the cepstrum in that the frequency bands are evenly spaced on the Mel scale, which more closely resembles the human auditory system's response than the linearly-spaced frequency bands used in the conventional spectrum. When used in audio compression, this frequency warping can improve the representation of sound and potentially lower the transmission bandwidth and storage needs of audio signals. Feature extraction is a special form of dataset reduction. Using feature extraction techniques for extracting specific features from the speech, these features carry the characteristics of the particular speech, which help differentiate the different speech so that these features will play a significant role in speech recognition. Compressing a voice signal into streams of acoustic feature vectors, also known as speech feature vectors, is the first step in speech recognition. The idea of feature extraction is divided into two steps: 1) the speech signal is transformed into feature vectors, and 2) the useful characteristics impervious to changes in the surroundings and speech variation are selected. In speech recognition systems, however, where accuracy has drastically declined in the case of their existence, changes in ambient variables and variances in speech are significant. The MFCC features, the most popular and reliable due to their precise estimation of the speech parameters and effective computational model of speech, are unquestionably the most often utilized speech features [17], [18]. *Fig. 1* shows the MFCC vs. time.

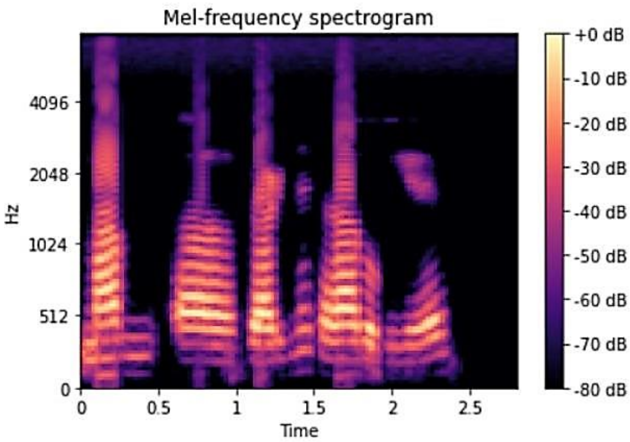

Fig. 1. Mel-frequency spectrogram vs. time.

3.2 | Data Preprocessing

I. Noise reduction: applied to minimize background noise and enhance sound quality.

II. Trimming: ensured all audio clips were standardized to a maximum duration of 4 seconds.

III. Normalization: adjusted audio levels to maintain consistency across samples.

IV. Feature extraction: MFCCs were used to convert audio signals into a format suitable for ANN processing.

3.3 | Model Architecture

3.3.1 | Support vector machine

I. Type: linear classifier.

II. Input: MFCC features extracted from audio samples.

III. Kernel: linear (for simplicity) or Radial Basis Function (RBF) kernel to handle non-linear patterns.

IV. Output: multi-class classification using the "one-vs-rest" strategy to differentiate the 10 urban sound classes.

3.3.2 | Random forest architecture

I. Type: ensemble classifier

II. Input: MFCC features extracted from audio samples.

III. Number of trees: 100 (default setting).

IV. Splitting criterion: Gini impurity or Entropy for node splitting.

V. Output: majority voting among trees for class prediction.

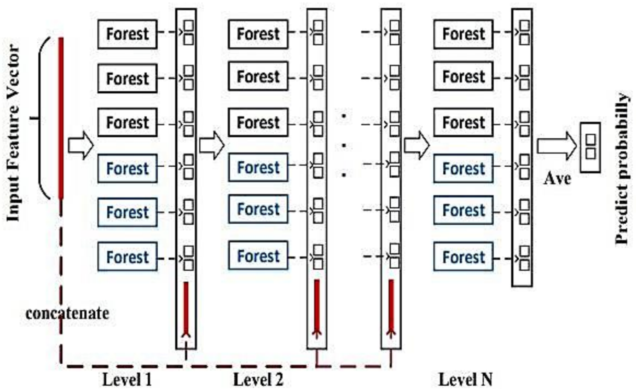

Fig. 2. Random forest architecture.

3.3.3 | Artificial neural network architecture

I. Input layer: 40 MFCC features as input.

II. First dense layer: 100 neurons, ReLU activation, Dropout (0.5).

III. Second dense layer: 200 neurons, ReLU activation, Dropout (0.5).

IV. Third dense layer: 100 neurons, ReLU activation, Dropout (0.5).

V. Output layer: softmax activation with 10 neurons for multi-class classification.

VI. Optimizer: Adam.

VII. Loss function: categorical crossentropy.

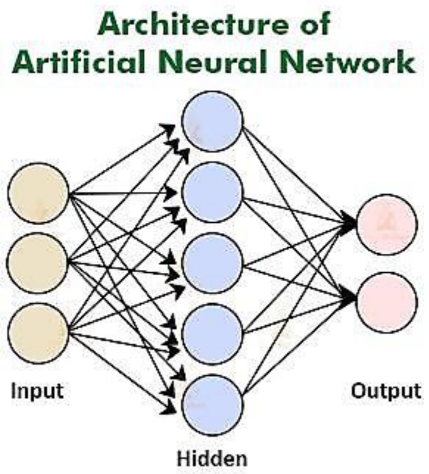

Fig. 3. Artificial neural network architecture.

3.3.4 | Convolutional neural network architecture

I. Input layer: spectrogram images of audio data.

II. Convolutional layers: two 2D convolutional layers with ReLU activation (e.g., 32 filters of size 3x3).

III. Pooling layers: max-pooling layers after each convolutional layer for down-sampling.

IV. Fully connected layer: dense layer with 128 neurons, ReLU activation.

V. Output layer: softmax activation with 10 neurons for multi-class classification.

VI. Optimizer: Adam.

VII. Loss function: categorical crossentropy.

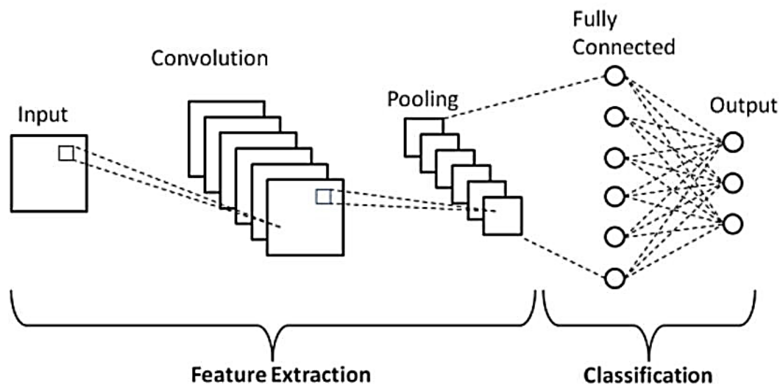

Fig. 4. Convolutional neural network architecture.

3.4 | Variables and Equations

Performance Metrics

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}},$$

$$= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}.$$

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

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

$$\text{F1} - \text{score} \ = \frac{2\ (\text{Precision}\ *\ \text{Recall})}{\text{Precision}\ +\ \text{Recall}},$$

$$= \frac{\text{TP}}{\text{TP}\ +\ 1/2(\text{FP}\ +\ \text{FN})}.$$

TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively. Precision is the ratio of correctly predicted data points to the total predicted data points and is defined as Precision = TP/(TP + FP).

4 | Experimental Setup

4.1 | Dataset

I. Dataset used: UrbanSound8K.

II. Description: 8,732 labeled audio samples across 10 urban sound classes (e.g., car horns, sirens, dog barks).

III. Sampling rate: 22,050 Hz (standardized for consistency).

IV. Splitting: the dataset is divided into 80% for training and 20% for testing, ensuring class balance.

4.2 | Preprocessing

I. Noise reduction: removal of background noise to enhance audio clarity.

II. Normalization: scaling audio amplitudes to ensure uniformity across samples.

III. Trimming: uniform duration enforced for all audio clips to simplify model input.

IV. Feature extraction: extracted MFCCs with 40 coefficients per sample for SVM, Random Forest (RF), and ANN models.

V. Generated spectrograms for CNN models, converting raw audio into 2D representations suitable for convolutional operations.

4.3 | Model Training and Parameters

I. Models: ANN, SVM, RF, CNN.

II. Training framework: TensorFlow/Keras for ANN and CNN; scikit-learn for SVM and RF.

III. Optimization techniques: ANN and CNN: optimized with the Adam optimizer and categorical cross-entropy loss.

IV. SVM: trained with an RBF kernel for non-linear classification.

V. RF: 100 decision trees with Gini impurity were used for splitting.

VI. Hyperparameters: batch size: 32 for ANN and CNN.

VII. Epochs: 100 for ANN and CNN.

VIII. Learning rate: default for Adam optimizer.

4.4 | Hardware and Software

Hardware

I. Processor: Intel Core i7 or equivalent.

II. RAM: 16 GB.

III. GPU: NVIDIA GeForce RTX 3060 (for CNN training).

Software

I. Python 3.9.

II. Libraries: TensorFlow, Keras, sci-kit-learn, Libros, numpy, pandas, matplotlib.

4.5 | Evaluation Metrics

I. Metrics used: accuracy.

II. Precision, Recall, and F1-Score (per class).

III. Confusion matrix will analyze classification performance by class.

4.6 | Experimental Process

I. Preprocessed the audio data and extracted MFCCs or spectrograms.

II. Trained each model (ANN, SVM, RF, CNN) separately using the extracted features.

III. Evaluated each model on the testing set to compare performance across classes.

IV. Recorded metrics for accuracy and per-class performance to identify strengths and weaknesses of each model.

5 | Experimental Results

Table 1. Effectiveness of different machine learning models.

Model	Input Features	Accuracy	Precision	Recall	F1-Score	Observation
SVM	MFCC	72.80%	Moderate	Moderate	Moderate	Performs well with linear and separabledata but struggles with overlapping noise.
RF	MFCC	75.60%	High	Moderate	Moderate	Handles noise betterthan SVM but may overfit to training data.
ANN	MFCC	80.20%	High	High	High	Robust in learningnon-linear patternsbut requires significant computational power.
CNN	Spectrogram images	87.50%	Very high	Very high	Very high	Excels in capturing spatial and temporalfeatures from spectrograms, outperforming others.

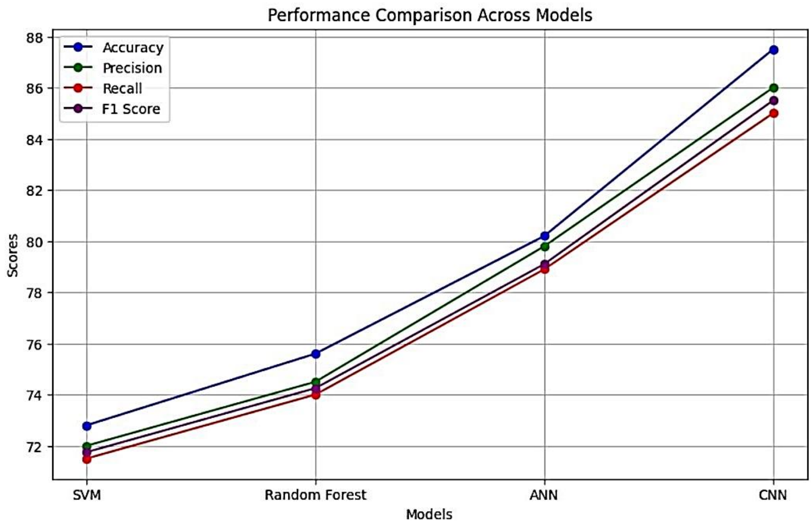

Fig. 5. Models performance.

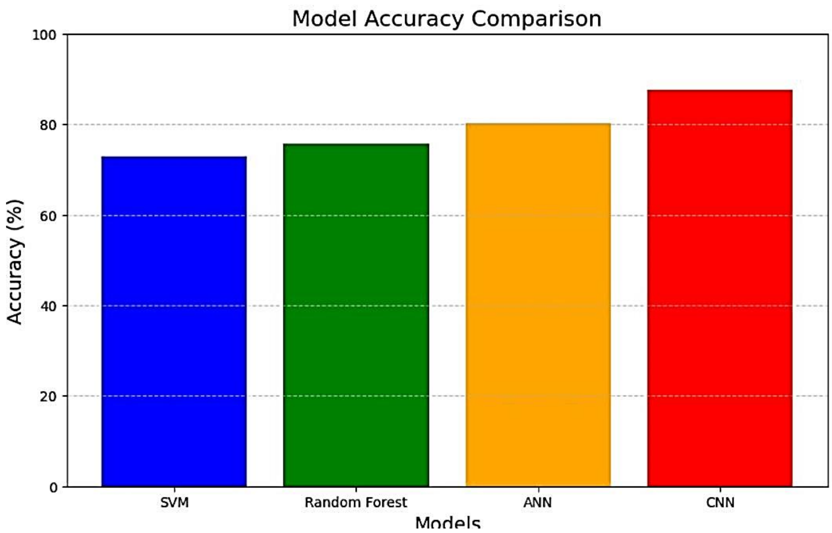

Fig. 6. Models accuracy.

6 | Conclusion

This study demonstrates the effectiveness of different machine learning models in classifying urban sound data, with a particular focus on the UrbanSound8K dataset. We evaluated four models: 1) SVM, 2) RF, 3) ANN, and 4) CNN, and compared their performance in terms of accuracy, precision, recall, and F1-score.

The results indicate that the CNN outperformed the other models with an accuracy of 87.50%, showcasing its potential for more complex, noisy data such as urban sounds. The ANN model followed with an accuracy of 80.20%, indicating that deep learning architectures are suitable for audio classification tasks. The RF and SVM models also performed well but with lower accuracies, emphasizing that feature learning in deep models may yield better results when dealing with the challenges of audio data classification.

This study highlights the value of deep learning approaches in audio classification, with CNNs standing out as the most effective model for this task. Future work can further explore data augmentation techniques, advanced hyperparameter tuning, and different feature extraction methods to continue improving classification performance.

Acknowledgments

We want to express our sincere gratitude to the contributors of the UrbanSound8K dataset, which has been instrumental in the success of this research. We also thank the developers of the machine learning libraries, such as TensorFlow, Keras, and Scikit-learn, for their extensive support in implementing and testing the models—special thanks to our faculty members and peers for their valuable feedback and assistance throughout this project.

Author Contributaion

Abhishek Raj: led the project, designed experiments, and wrote the manuscript.

Abhishek Yadav: handled data preprocessing, feature extraction, and model implementation.

Vineet Kumar: implemented CNN and ANN models and analyzed results.

Abhay Kumar: implemented SVM and RF models and visualized results.

Sankalp Anand: assisted with data collection, literature review, and manuscript feedback.

Data Availability

The dataset used in this research, UrbanSound8K, is publicly available and can be accessed via the official website (https://urbansounddataset.weebly.com/urbansound8k.html). The code and models developed during this research are available upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest related to the content of this research paper.

References

[1] Mohapatra, H. (2021). Socio-technical challenges in the implementation of smart city. *2021 international conference on innovation and intelligence for informatics, computing, and technologies (3ICT)* (pp. 57–62). IEEE. https://doi.org/10.1109/3ICT53449.2021.9581905

[2] Nogueira, A., Oliveira, H., Machado, J., & Tavares, J. (2022). Sound classification and processing of urban environments: A systematic literature review. *Sensors*, *22*, 8608. http://dx.doi.org/10.3390/s22228608

[3] Pudasaini, A., Al-Hawawreh, M., Bouadjenek, M. R., Hacid, H., & Aryal, S. (2024). A comprehensive study of audio profiling: methods, applications, challenges, and future directions. *Journal of latex class files*, *14*(8). https://doi.org/10.36227/techrxiv.171595948.84728317/v1

[4] Dongare, A. D., Kharde, R. R., & Kachare, A. D. (2012). Introduction to artificial neural network. *International journal of engineering and innovative technology (IJEIT)*, *2*(1), 189–194. https://b2n.ir/j98967

[5] Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology*, *24*(12), 1565–1567. https://doi.org/10.1038/nbt1206-1565

[6] Das, J. K., Ghosh, A., Pal, A. K., Dutta, S., & Chakrabarty, A. (2020). Urban sound classification using convolutional neural network and long short term memory based on multiple features. *2020 fourth international conference on intelligent computing in data sciences (ICDS)* (pp. 1–9). IEEE. https://doi.org/10.1109/ICDS50568.2020.9268723

[7] Zou, J., Han, Y., & So, S. S. (2009). Overview of artificial neural networks. *Artificial neural networks: methods and applications*, 14–22. https://doi.org/10.1007/978-1-60327-101-1_2

[8] Biswas, D. G., Das, S., Kairi, A., Roy, A., Saha, T., & Samanta, M. (2024). Taxonomic delineation of musical genres through computational paradigms: an exploration employing the k-nearest neighbors (kNN) algorithm. *Proceedings of the fifth international conference on emerging trends in mathematical sciences & computing (IEMSC-24)* (pp. 128–144). Cham: Springer, Cham. https://doi.org/10.1007/978-3-031-71125-1_11

[9] Kademani, V., A, A., Patil, P., & M, M. S. (2024). A deep learning approach for accurate environmental sounds analysis. *2024 5th international conference for emerging technology (INCET)* (pp. 1–6). IEEE. https://doi.org/10.1109/INCET61516.2024.10593397

[10] Bhise, D., Kumar, S., & Mohapatra, H. (2022). Review on deep learning-based plant disease detection. *2022 6th international conference on electronics, communication and aerospace technology* (pp. 1106–1111). IEEE. https://doi.org/ 10.1109/ICECA55336.2022.10009290

[11] Zaman, K., Sah, M., Direkoglu, C., & Unoki, M. (2023). A survey of audio classification using deep learning. *IEEE access*, *11*, 106620–106649. https://doi.org/10.1109/ACCESS.2023.3318015

[12] Salamon, J., & Bello, J. P. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE signal processing letters*, *24*(3), 279–283. https://doi.org/10.1109/LSP.2017.2657381

[13] Demir, F., Abdullah, D. A., & Sengur, A. (2020). A new deep CNN model for environmental sound classification. *IEEE access*, *8*, 66529–66537. https://doi.org/10.1109/ACCESS.2020.2984903

[14] Malaviya, P., Kumar, Y., & Modi, N. (2023). Advancements in environmental sound classification: evaluating machine learning and deep learning approaches on the urbansound8k. *2023 seventh international conference on image information processing (ICIIP)* (pp. 900–905). IEEE. https://doi.org/10.1109/ICIIP61524.2023.10537679

[15] Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial intelligence review*, *53*(8), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1

[16] Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W. F., & Weiss, B. (2005). A database of german emotional speech. *Interspeech* (pp. 1517–1520). ISCA. https://doi.org/10.21437/interspeech.2005-446

[17] Majeed, S. A., Husain, H., Samad, S. A., & Idbeaa, T. F. (2015). Mel frequency cepstral coefficients (MFCC) feature extraction enhancement in the application of speech recognition: A comparison study. *Journal of theoretical & applied information technology*, *79*(1). https://www.researchgate.net/publication/281785424

[18] Dev, A., & Bansal, P. (2010). Robust features for noisy speech recognition using MFCC computation from magnitude spectrum of higher order autocorrelation coefficients. *International journal of computer applications*, *10*(8), 36–38. https://b2n.ir/a10869