Mansoura Engineering Journal

Volume 48 | Issue 2 Article 12

2023

Optimized Deep Learning Audio Tagging Approach

Fatma S. El-metwally
Department of Computer Engineering and Control Systems,Faculty of Engineering, Mansoura University, fatema.shaban19@gmail.com

Ali I. Eldesouky
Computer Engineering Department, University of Mansoura, Mansoura

Sally M. Elghamrawy
Communications & Computer Engineering Department, Misr Higher Institute for Engineering and Technology, sally@mans.edu.eg

ORIGINAL STUDY

Optimized Deep Learning Audio Tagging Approach

Fatma S. El-metwally [a], Ali I. ElDesouky, Sally M. Elghamrawy [b,*]

[a] Department of Computer Engineering and Control Systems, Faculty of Engineering, Mansoura University, Egypt
[b] Computer Engineering Department, Misr Higher Institute for Engineering and Technology, Mansoura, Egypt

Abstract

Audio signal processing is a method for applying powerful algorithms and techniques to record, improve, save, and transmit audio content signals. Audio Tagging (AT) is a challenge that requires predicting the tags of audio clips. Developments in deep learning and audio signal processing have resulted in a significant improvement in audio tagging. Many techniques have been used. Several studies have introduced different audio tagging techniques, but the performance of the results obtained from these studies is insufficient. This study proposes an Optimized Deep Learning Audio Tagging (ODLAT) approach to classify and analyze audio tagging. Each input signal is used to extract the different characteristics or features of the audio tagging. Such features are input into a neural network to carry out a multi-label classification for the predicted tags. Adam and Adamax are used as effective optimization methods for learning rate. Many experiments are conducted to test the validity of the Optimized Deep Learning Audio Tagging approach against others. The results obtained have shown the superiority of the proposed approach.

Keywords: Audio tagging, Deep learning, Multi−label classification, Short time Fourier transform

1. Introduction

Audio signal processing is a method for applying powerful methods and techniques to audio signals (Rao, 2008). Devices such as smartphones have been increasingly popular in recent years, and communication remotely via the internet has become the preferred way to connect over face-to-face meetings. However, in any process of communication, auditory noise, distortion, and echo are unavoidable. Due to the widespread use of electronic means of communication, a huge number of multimedia recordings are produced and published on the Internet continuously. These recordings contain multiple media that include music, news broadcasts, television programs, and science articles. Many audio events can be identified and distinguished by humans. But it is a very difficult task for a machine. As a result, more studies are required to develop powerful systems capable of recognizing a variety of acoustic events (Virtanen et al., 2018). Audio signal processing techniques are used in many applications. For example, monitoring of health-related activities (Goetze et al., 2012), robotic systems, eLearning, and intelligent surveillance systems that use audio signals to recognize activities in their environments. Therefore, much more research is required to accurately acknowledge audio scenes and individual audio sources in realistic audio scenes, where there are multiple voices, often at the same time and distorted by environmental noise (Mesaros et al., 2017). The need for analyzing these sounds has grown in recent years because it is useful. Audio tagging is a technique for predicting one or multiple labels in an audio clip. The Detection and Classification of Acoustic Scenes and Events (DCASE) challenge (Giannoulis et al., 2013; Stowell et al., 2015) provide strongly labeled datasets. For the musical tagging task (Pons et al., 2017; Choi et al., 2016), deep learning methods have proved their efficiency. Deep learning-based algorithms have also been utilized for environmental audio tagging, it is a suggested task in the DCASE

Received 12 September 2022; revised 27 October 2022; accepted 12 January 2023.
Available online 30 December 2023

* Corresponding author. Computer Engineering Department, Misr Higher Institute for Engineering and Technology, Mansoura, 35516, Egypt.
E-mail address: sally@mans.edu.eg (S.M. Elghamrawy).

https://doi.org/10.58491/2735-4202.3096

2016 challenge () based on the CHiME-home dataset (Foster et al., 2015). Until now, the majority of audio-related recognition technologies have been employed. The frequency domain of the audio signal is used to extract features (Subramanian et al., 2004), such as Mel frequency cepstral coefficients (MFCCs) (Cakir et al., 2015), log-frequency filter banks (Nadeu et al., 2001), and time-frequency filters (Chu et al., 2009).

The task of audio tagging has been extensively studied. The Gaussian mixture model (GMM) is trained on MFCCs (Yun et al., 2016), convolutional neural networks (CNNs) with inputs from the Mel spectrogram (Cakõr et al., 2016), and deep neural networks with inputs from the Mel Filter Bank (Kong et al., 2016). This study presents an optimized deep learning audio tagging (ODLAT) approach for audio signal classification. First, the features are extracted from the input signal, and then these features are input into a neural network. The learning rate may be the most important hyperparameter when configuring your neural network. An Adam or Adamax optimizer is used to optimize the parameters of the neural network. This study was structured into several sections. Section 2 discusses related work, and Section 3 discusses the Proposed ODLAT approach. Section 4 discusses the experimental results. Finally, in this section, the conclusion and future work are presented.

2. Related work

The DCASE challenge is a set of tasks designed to improve sound classification and detection systems.

In (Hertel et al., 2016), a short-time fourier transform was used to extract features from audio signals that are input into a convolutional neural network to perform multi-label classification for audio tagging. They achieved an overall accuracy of 84.5% and an average equal error rate (EER) of 0.17. In other research (Vu and Wang, 2016), a MFCCs input feature signal was used. Each audio signal was transformed into 13-dimensional MFCCs with frame sizes of 0.04 s and hop sizes of 0.02 s. Recurrent Neural Networks were used for classification. On the evaluation set, they obtained an average EER of 0.21.

In the research (Lidy and Schindler, 2016), convolutional neural networks were used, which were trained on the constant Q shift (CQT) feature of the audio signal. In the evaluation set, they obtained an average EER of 0.166.

In the research (Xu et al., 2016), a MFCCS input feature signal was used. Each audio chunk was preprocessed by segmenting it with an (80 ms) sliding window with a hop size of 40 ms and converting it to 24-Dimensional MFCCs. They obtained an average EER of 0.1785.

In this research (Xu et al., 2017), MFCCS and mel filter bank features were used, and these features were input to a deep neural network with an SGD optimizer. The MFCC Feature achieved results with an average EER of 0.168. The MFB Feature achieved results with an average EER of.157. It turns out that the feature MFB is better than MFCC, so it has taken the MFB feature and conducted it with another network that has a Denosing Autoencoder (DAE) with SGD optimizer and achieved results with an average EER of 0.148.

In (Wei, 2018), a sample mixed data augmentation was explored, which included mixup, sample pairing, mixup with label preserving (mixup lp), and extrapolation. It used a convolutional recurrent neural network (CRNN) with an attention module with a log-scaled mel spectrum as a baseline system. The efficiency of the CRNN neural network model was examined (without and with data augmentation). Without data augmentation, they achieved an ER of 0.13. In the case of data augmentation, it can effectively improve classification (including mixup, sample pairing, and extrapolation). In the case of data mixup, different ratios were used for the mixing approach, and when using 1.5 and 2.0, achieved an average ER of.11. But in the case of sample pairing, they achieved an average ER of.13, but in the case of extrapolation, they achieved an average ER of.12. This implies that the CRNN model with the mixup approach produces better results.

3. The proposed optimized deep learning audio tagging approach

ODLAT approach consists of four layers: signal representations layer, data preprocessing layer, deep learning layer, prediction layer (Fig. 1).

3.1. Signal representations layer

Audio is any waveform whose frequencies are in the human audible range. Audio is produced by the shaking of a body, and that shaking causes the wiggle of air particles, which leads to a change in air pressure. The combination of high and low air pressure causes a wave, and we can represent this wave using wave form. If you take a look at the plotted audio wave in the time domain in Fig. 2, it looks so complex to understand, but nature has

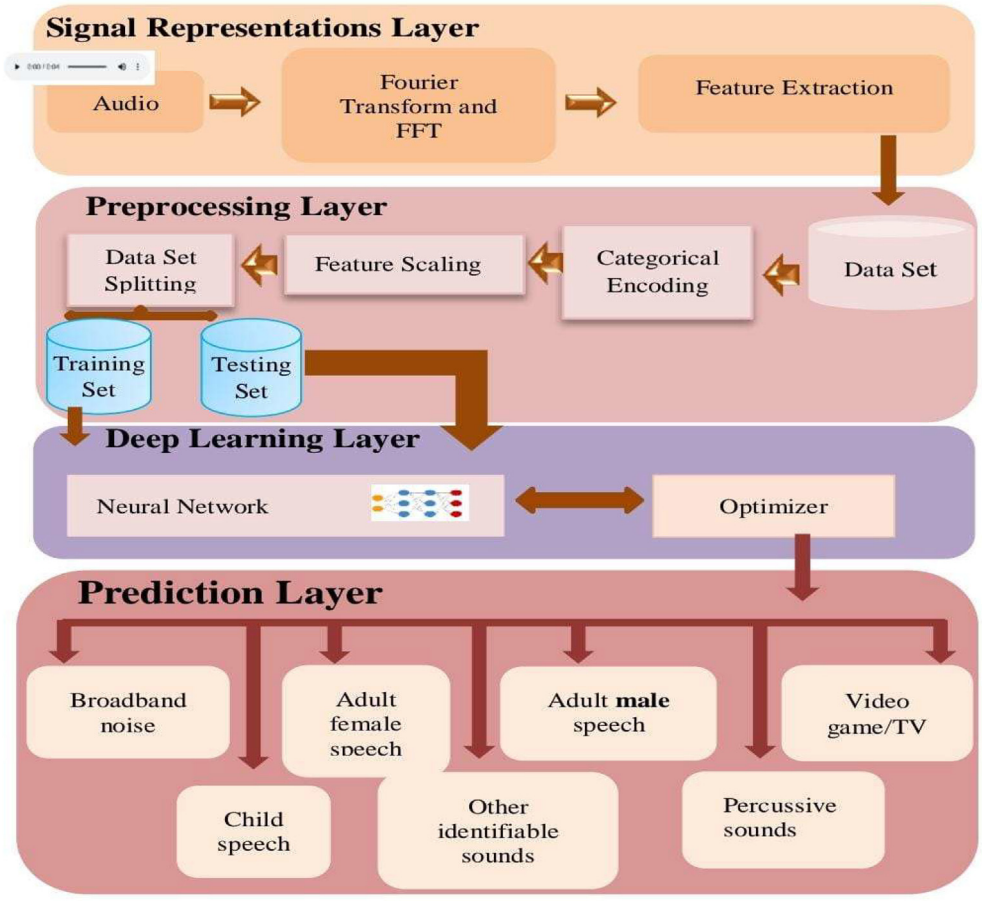

Fig. 1. The proposed optimized deep learning audio tagging (ODLAT) approach.

given us an incredible way of knowing quite a lot about complex sounds, and that's given through a fourier transform (FT). A Fourier transform is used to decompose complex periodic audio into a sum of sine waves oscillating at different frequencies.

The 'Fast Fourier Transform' (FFT) is a significant analytical technique in the field of audio. It decomposes a signal into its spectral components and offers information about the signal's frequency.

Feature extraction is the procedure for extracting features in order to use them in analysis. Each audio signal has a variety of features. So we need to extract the features (Elghamrawy et al., 2022) related to the problem we are trying to solve.

Short-time fourier transform (STFT) is an effective audio signal processing tool that aims to mimic human perception, such as the recognition of auditory scenes or automatic music transcription. It computes several FFT at different intervals, preserves time information, and gives a spectrogram (time + frequency + magnitude) as shown in Fig. 3 (Müller, 2015).

3.2. Preprocessing layer

Data preprocessing is a crucial stage in machine learning, as the quality of the data influences the learning model's performance. Most datasets contain noise, insufficient variables, and are inaccurate (contain errors). Therefore, they cannot be used directly for machine learning. For this reason, it is very important that our data be preprocessed before being fed into our model. Data preprocessing refers to the procedures that must be followed to transform or encode data in order for it to be easily recognized by a machine.

Categorical Encoding: categorical data is information that has distinct categories within a data set. Machine learning models work primarily with numerical data. Categorical data is meaningless to a computer, but it is useful to us. To make the dataset useable, we must convert these categorical variables into numerical representations. Label encoding is a popular encoding method for dealing with categorical variables (Hancock and Khoshgoftaar, 2020).

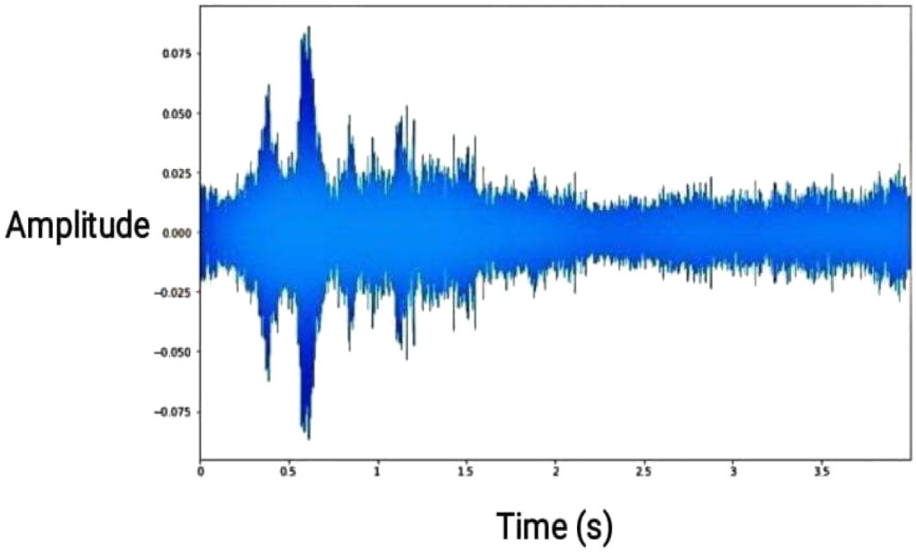

Fig. 2. Plot audio wave in time domain.

Feature scaling: it is an important step in the data processing part. The goal of this technique is to ensure that all features are on the same scale. So we need to perform feature scaling so that one large number does not have an impact on the model simply because it is too large. A standardization scale is used in which features are transformed by subtracting them from the mean and dividing them by the standard deviation. In the case of scaling the feature, all data is on the same scale as shown in Fig. 4. In the case of without scaling the feature, all data is not on the same scale as shown in Fig. 5 (Ali et al., 2014).

Here's the formula for standardization value:

$$\text{Standardized value} = \frac{x - \mu}{\sigma} \quad (1)$$

Here, σ is the standard deviation, σ the mean.

Data set splitting: a data set splitting strategy is required for building a model with good generalization performance, as well as for model validation, k-folds Cross validation was used. The sample data is divided into k parts, k-1 parts are used for training, and 1 for testing repeat the procedure five times, and rotate the test group. Determine expected performance based on iteration results.

3.3. Deep learning layer

The DNN (deep neural network) is a nonlinear multi-layer model for extracting characteristics linked to a specific classification (Hinton et al., 2012) or regression (Xu et al., 2014) task.

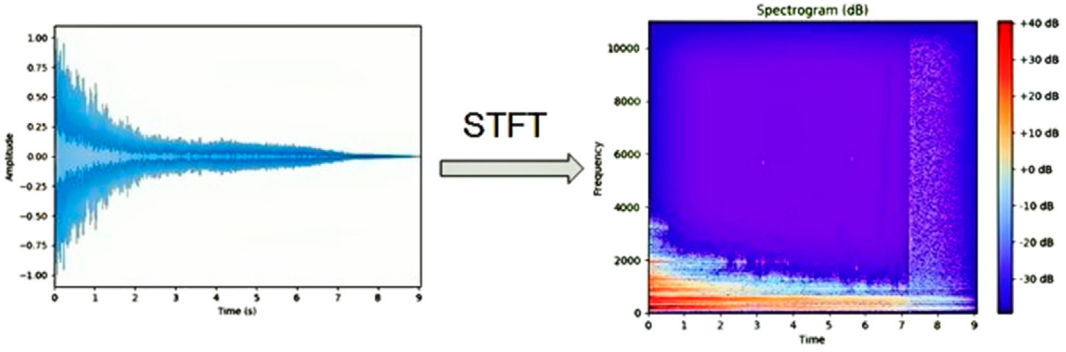

Fig. 3. Plot short-time fourier transform of the audio.

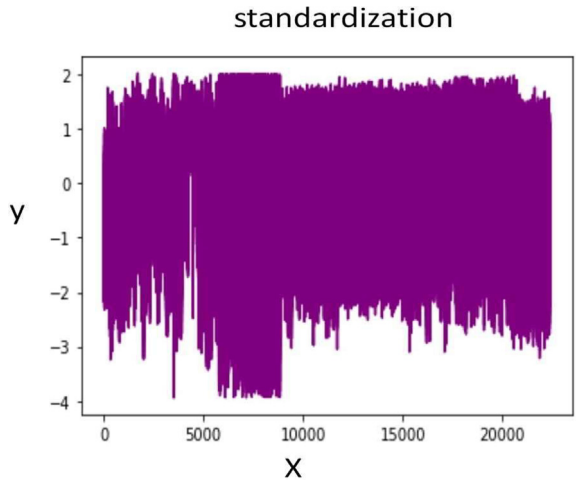

Fig. 4. Impact of the feature scaling process.

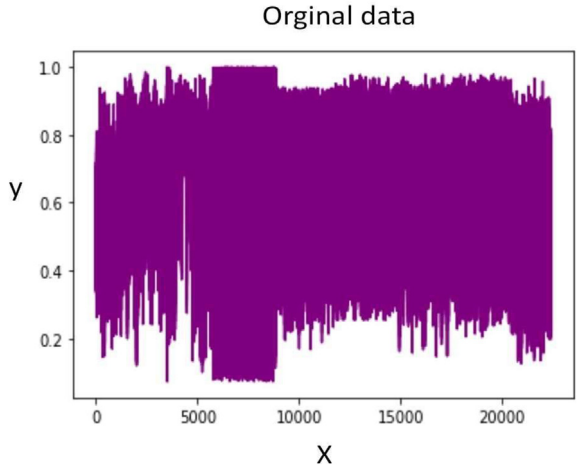

Fig. 5. Impact without the feature scaling process.

3.3.1. Deep neural network structure

A DNN structure consists of three layers: input, hidden, and output. Fig. 6 shows the DNN structure. The hidden layer may have two or more layers, whereas the input and output layers are single layers. The hidden layer contains a set of neurons The parameters used in a neural network are shown in Table 1.

The input layer receives data features. After processing in the hidden layers, prediction values are produced from the output layer.

Due to the enormous number of variables in neural networks, they are prone to over-fitting. Dropout is a technique used in neural networks to prevent over-fitting. Dropout indicates the removal of units (both hidden and visible) from a neural network, as well as all of its inbound and outgoing connections. In the original method, during each training iteration, each neuron in a neural network is removed with a probability of 0.5, with all neurons being included during testing (Srivastava et al., 2014). The dropout rates for hidden layers are 0.1 and 0.2, respectively.

3.3.2. Optimizer

When applying the deep learning technique, we have the notion of loss, which informs us of how poorly the model is performing right now. Now we need to use this loss to train our network so that it works better. So basically, what we need to do is take the loss and try to reduce it. Because a lower loss indicates that our model will perform better. The process of minimizing (or maximizing) is called optimization. Optimizers are techniques that modify

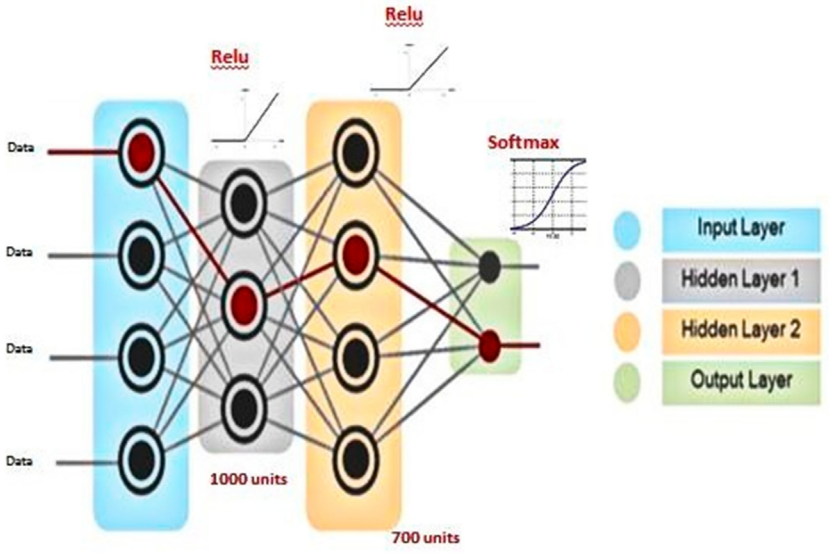

Fig. 6. Deep neural network structure.

Table 1. Parameters of a neural network.

NN parameter	Values
Classifier	Sequential
Number of Hidden layer	2
Hidden activation function	Relue
Number of Neuron in first hidden layer	1000
Number of Neuron in second hidden layer	700
Output activation function	Softmax
Optimizer	Adam - Adamax
loss function	Categorical cross entropy
Batch Size	100
Number of Epoch s	100
Learning rate	0.005
Momentum	0.9

the neural network's properties, such as its weights and learning rate, Optimizers are used to solve optimization problems by minimizing the function. As a result, the primary aim of the optimizer is to find the optimal value for the neural network weights to minimize the objective function (loss/cost function). Adam or Adamax is used as an optimization technique.

Adam Optimizer: Adam is a deep neural network training-specific adaptive learning rate optimization algorithm that calculates individual learning rates for various parameters.

The first momentum is obtained by

$$m_i = \beta_1 m_{i-1} + (1-\beta_1)\frac{\partial C}{\partial W} \tag{2}$$

The second momentum is obtained by

$$V_i = \beta_2 V_{i-1} + (1-\beta_2) \tag{3}$$

$$W_{i+1} = W_i - \eta \frac{\widehat{m}_i}{\sqrt{\widehat{V}_i + \varepsilon}} \tag{4}$$

$$u_i = max\left(\beta_2 . u_{i-1}, \left|\frac{\partial C}{\partial W}(W_i)\right|\right) \tag{5}$$

$$W_{i+1} = W_i - \eta \frac{\widehat{m}_i}{u_i} \tag{6}$$

Where $\widehat{m}_i = m_i/(1\text{-}\beta_1)$, $\widehat{V}_i = \frac{V_i}{1-\beta_2}$, η = Learning rate, $\beta_1, \beta_2 \varepsilon\ [0,1]$ And C (W) denote Cost function with parameters w, In comparison to Adam, Adamax is superior (Yi et al., 2020).

3.4. Prediction layer

This layer presents the whole results obtained from the proposed framework.

4. Experiment results and discussion

Before starting the stage of performing the model, there are some limitations that must be taken into consideration in order to avoid the model from falling into any type of problems that affects the accuracy of the model.

4.1. Selection neural network

The deep learning neural network was selected in this research because the data used in this research is audio recordings, which is unstructured data, and deep learning deals better with structured data. Solve challenging issues like audio processing and eliminate the need for manual feature extraction. Models can be trained on massive amounts of data, and the model improves as more data. Automated tasks that use Keras and Tensorflow can make predictions in less time.

4.2. Optimal feature extraction

Each audio signal has a lot of features, so we must choose the feature that best fits the problem that we want to solve. Audio signal processing algorithms analyze signals, extract their features, and detect the presence of any pattern in the signal.

4.3. Specify the labels used in the research

In a DSCASE challenge, for example, the audio has 9 labels, as shown in Table 2, but the set of labels that are allowed is 7. Any subset of labels can be assigned to an audio clip by an author. Except for the labels S and U, they can only be set separately.

4.4. Data set DCASE2016 for audio tagging

A deep learning model is applied to the DCASE 2016 audio tagging challenge's CHIME-HOME

Table 2. Labels of the audio DSCASE data set.

Label	Description
c	Child speech
m	Adult male speech
f	Adult female speech
v	Video game /TV
p	Percussive sounds, e.g. crash, bang, knock, footsteps
b	Broadband noise, e.g. household appliances
o	Other identifiable sounds
s	Silence / background noise only
u	Flag chunk (unidentifiable sounds, not sure how to label)

dataset. The audio recordings were created in a domestic environment (Christensen et al., 2010). The goal is to classify 4 s audio chunks using several labels at a sampling rate of 16 kHz. There are seven labels that appear in audio segments, as shown in Table 3. Besides, sounds issued from outside the house.

A STFT is used to extract features from an audio signal. Each audio clip is converted into 13 dimensions, with a window length of 320 and a hop size of 160.

4.5. Evaluation metrics

Evaluation metrics are used to assess the model's performance, such as the confusion matrix shown in Table 4, which is a summarized table of a classifier's correct and incorrect predictions and which is widely utilized to offer a variety of classification metrics and performance evaluation parameters.

True Positive (TP): is a term that refers to accurate positive forecasts. False Negative (FN) is a term that refers to inaccurate negative forecasts. False Positive (FP): positive forecasts that are inaccurate. True Negative (TN): negative forecasts that are accurate. Evaluation metrics that are driven from the confusion matrix.

$$\text{Accuracy (Acc)} = \frac{TP + TN}{(TP + FN + FP + TN)} \quad (7)$$

$$\text{Equal Error Rate (ERR)} = \frac{FP + +FN}{(TP + TN + FN + FP)} \quad (8)$$

A model with a lower EER is regarded as more accurate, whereas a model with a higher accuracy coefficient (ACC) is regarded as superior (Saito and Rehmsmeier, 2015).

Table 3. Audio data set labels.

Label/Audio events	Event Description
Event “b”	Broadband noise
Event “c”	Child speech
Event “F”	Adult female speech
Event “m”	Adult male speech
Event “o”	Other identifiable sounds
Event “p”	Percussive sound events
Event “v”	Video game / TV

Table 4. Confusion matrix.

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

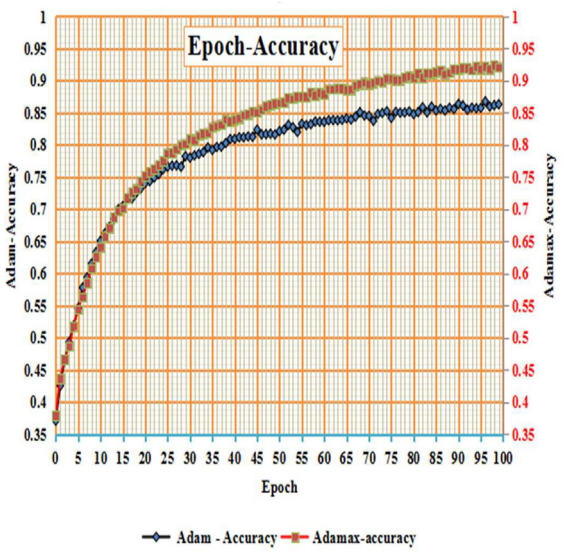

Fig. 7. Adamax accuracy and Adam accuracy by epoch.

4.6. Experiment result number one: the stability of the algorithm with and with the optimizer

4.6.1. Test DCASE 2016 accuracy and loss with optimizer

This experiment is used to test the accuracy and loss of the proposed approach. A DCASE2016 Task 4 data set was used. This data was trained on a DNN consisting of two hidden layers: an input layer and an output layer. The binary cross entropy is used as the cost function, and Adam or Adamax is used as an optimizer. The training process stops after the time period specified (100 epoch).

As shown in Fig. 7, the accuracy of the Adamax optimizer is better than that of the Adam optimizer.

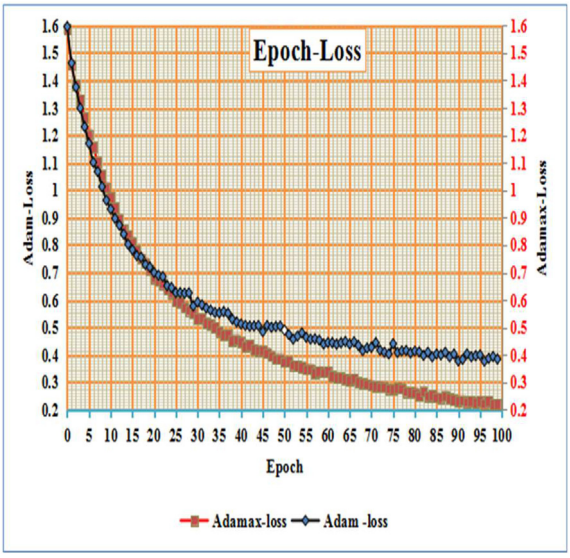

Fig. 8. Adamax loss and Adam loss by epoch.

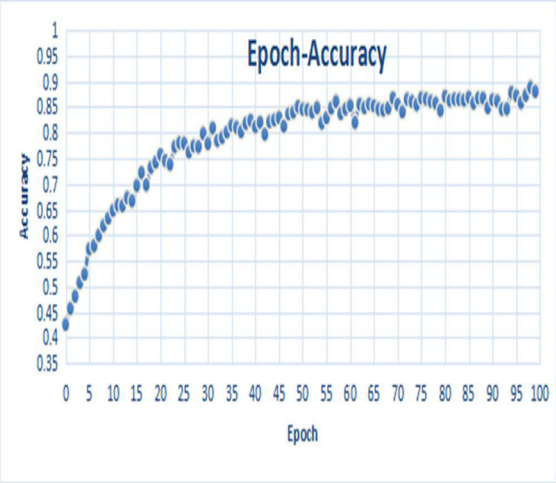

Fig. 9. Epoch by accuracy without optimizer.

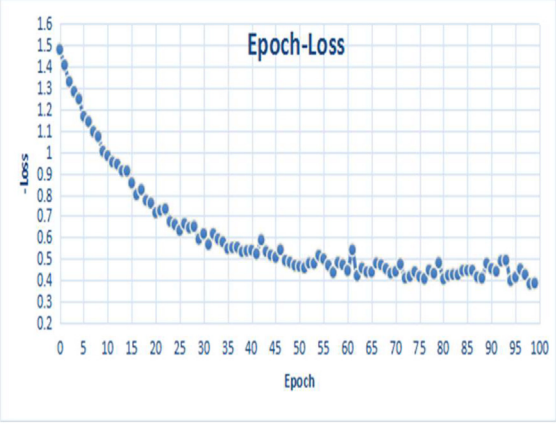

Fig. 10. Epoch by loss without optimizer.

At epoch 100, the accuracy of the Adamax optimizer is 93% and the accuracy of the Adam optimizer is 87%. In Fig. 8, the loss of the.

Adamax optimizer is better than the Adam optimizer. At epoch 100, the loss of the Adamax optimizer is 0.22%. and the loss of the Adam optimizer is 0.38%.

4.6.2. Test DCASE 2016 accuracy and loss without optimizer

Performance optimization tools can be divided into two groups, with each offering a variety of choices. They take a different strategy to reduce the cost function of a neural network and improve the model, producing various results and also fluctuating in speed and complexity, which affects training time and resources.

In Figs. 9 and 10, the accuracy and loss without the optimizer are shown. At epoch 100, the model without an optimizer achieved an accuracy of 85% and a loss of 0.38%, respectively.

When the model is used without an optimizer, it affects the speed of the model's performance, accuracy, and loss.

4.6.3. Experiment result number two: test ODLAT's accuracy

This experiment is used to test ODLAT's ACC for seven tags, which measures the number of correct predictions for the approach. The results obtained in Table 5 ACC from the proposed approach work show that ACC in Adamax optimizer is better than Adam optimizer. The average acc increased from 0.972 to 0991 in the development set and from 0.956 to 0.966 in the evaluation set.

Table 5. Accuracy results from the proposed framework.

Audio events	Event 'b'	Event 'c'	Event 'f'	Event 'm'	Event 'o'	Event 'p'	Event 'v'	Average
Development Set								
STFT-DNN (Adam-opt)	0.995	0.961	0.982	0.987	0.961	0.945	0.978	0.972
STFT-DNN (Adamax-opt)	0.998	0.981	0.995	0.996	0.990	0.991	0.989	0.991
Evaluation Set								
STFT-DNN (Adam-opt)	0.991	0.915	0.971	0.981	0.947	0.938	0.949	0.956
STFT-DNN (Adamax-opt)	0.994	0.925	0.981	0.984	0.961	0.964	0.956	0.966

Table 6. Equal error rate comparisons between the results obtained from the proposed framework.

Audio events	Event 'b'	Event 'c'	Event 'f'	Event 'm'	Event 'o'	Event 'p'	Event 'v'	Average
Development Set								
STFT-DNN (Adam-opt)	0.011	0.094	0.037	0.027	0.055	0.072	0.060	0.050
STFT-DNN (Adamax-opt)	0.005	0.080	0.010	0.008	0.030	0.020	0.040	0.027
Evaluation Set								
STFT-DNN (Adam-opt)	0.011	0.095	0.030	0.025	0.054	0.064	0.060	0.048
STFT-DNN (Adamax-opt)	0.003	0.061	0.012	0.009	0.023	0.018	0.039	0.023

Table 4. Summary of the proposed approach with previous studies.

Ref	Year	System characteristics			Equal error rate (Average)		Accuracy
		Features	Classifier	Optimizer	(evaluation dataset)	(development dataset)	
Lars Hertel1 (Hertel et al., 2016)	2016	STFT	Convolutional Neural Networks (CNN)	Adam	0.210	0.170	84.50%
Toan H (Vu and Wang, 2016)	2016	MFCCs	Recurrent Neural Networks (RNN)	ADADELTA	0.210	0.20	
Thomas Lidy (Lidy and Schindler, 2016)	2016	CQT Features	CNN	Stochastic gradient descent (SGD)	0.178	0.166	
Yong Xu (Xu et al., 2016)	2016	MFCCs	Deep Neural Networks (DNN)	SGD		0.1785	
Yong Xu (Xu et al., 2017)	2017	MFCCs	DNN	SGD	0.168	0.151	
		MFBs	DNN	SGD	0.157	0.135	
		MFBs	Denosing Autoencoder (DAE)	SGD	0.148	0.126	
The proposed approach	2022	STFT	DNN	Adam	0.048	0.050	87%
	2022	STFT	DNN	Adamax	0.023	0.027	93%

Table 5. Summary of previous studies.

Ref	Lars Hertel1 [18]	Toan H [19]	Thomas Lidy [20]	Yong Xu [21]	Yong Xu [22]
Broadband noise	0.18	0.26	0.032	0.0868	0.067
Child speech	0.2	0.24	0.21	0.1686	0.124
Adult female speech	0.23	0.11	0.214	0.2409	0.202
Adult male speech	0.06	0.21	0.182	0.1943	0.092
Other identifiable sounds	0.19	0.29	0.32	0.2867	0.231
Percussive sound events	0.11	0.23	0.168	0.2197	0.143
TV sound	0.24	0.06	0.035	0.0530	0.023
Average	0.17	0.20	0.16	0.1785	0.126

4.6.4. Experiment result number three: test ODLAT's equal error rate

This experiment calculates the test ERR. The results obtained in Table 6 EER from the proposed approach for seven tags show that EER in the Adamax optimizer is better than in the Adam optimizer. The average EER decreased from 0.050 to 0.027 in the development set and from 0.048 to 0.023 in the evaluation set.

5. Discussion

In the proposed approach, input signals are processed using the STFT feature to extract the features or characteristic from the audio signal, and these features are entered into the deep neural network with Adam or Adamax optimizer. Table 7 shows the results obtained from the proposed approach ensuring that Adamax outperforms Adam. The comparative study results showed the superiority of the proposed approach to Hertel1 and Phan (Hertel et al., 2016), Toan and Wang (Vu and Wang, 2016), Thomas Lidy and Schindler (Lidy and Schindler, 2016), Yong Xu (Xu et al., 2016), and Yong Xu (Xu et al., 2017). as shown in Tables 7 and 8.

5.1. Conclusion

We provide an ODLAT approach in this study. Experiments were carried out on DCASE 2016 Task 4. In the classification of environmental sound sources, the STFT has been applied. DNN have been demonstrated to be useful for audio tagging and classification. To prevent the neural network from over-fitting, a dropout was also implemented. As an optimization strategy, Adam or Adamax is employed. In future work, this study will be applied to another audio feature to extract features from audio signals, such as MFCCs or CNNs. This may extract more high-level features for the audio tagging task, or different binary optimizers may be tested.

Authors contribution

F S. E-m is responsible for conception of the work, software, and drafting the article. Participates in developing modeling and application programs. S M. E is responsible for project administration and final approval of the version to be published. Participates in preparing the theoretical model and the work of the solution program. A I. E is responsible for the supervision and critical revision of the article. Proposes the research topic and Participates in the analysis of theoretical results.

Funding statement

The authors did not receive any financial support of the research authorship and publication of this article.

Conflicts of interest

Declaration of conflicting interests statement: the author declared that there are no potential conflicts of interest with respect to the research authorship or publication of this article.

References

Ali, P.J.M., Faraj, R.H., Koya, E., 2014. Data normalization and standardization: a technical report. Mach. Learn. Tech. Rep. 1, 1–6.

Cakõr, E., Heittola, T., Virtanen, T., 2016. Domestic Audio Tagging with Convolutional Neural Networks', vol. 2016. IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2016), pp. 1–2.

Cakir, E., et al., 2015. Polyphonic Sound Event Detection Using Multi Label Deep Neural networks.' 2015 International Joint Conference on Neural Networks (IJCNN). IEEE.

Choi, K., Fazekas, G., Sandler, M., 2016. Automatic Tagging Using Deep Convolutional Neural Networks, vol. 1606. arXiv preprint arXiv, 00298.

Christensen, H., et al., 2010. The CHiME corpus: a resource and a challenge for computational hearing in multisource environments. In: Eleventh Annual Conference of the International Speech Communication Association.

Chu, S., Narayanan, S., Kuo, C-C Jay, 2009. Environmental sound recognition with time–frequency audio features. IEEE Trans. Audio Speech Lang. Proc. 17, 1142–1158.

Elghamrawy, S.M., Hassanien, A.E., Vasilakos, A.V., 2022. Genetic-based adaptive momentum estimation for predicting mortality risk factors for COVID-19 patients using deep learning. Int. J. Imag. Syst. Technol. 32, 614–628.

Foster, P., et al., 2015. Chime-home: A Dataset for Sound Source Recognition in a Domestic environment.' 2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). IEEE.

Giannoulis, D., et al., 2013. Detection and Classification of Acoustic Scenes and Events: an IEEE AASP challenge.' 2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. IEEE.

Goetze, S., et al., 2012. Acoustic monitoring and localization for social care. J. Comput. Sci. Eng. 6, 40–50.

Hancock, J.T., Khoshgoftaar, T.M., 2020. Survey on categorical data for neural networks. J. Big Data 7, 1–41.

Hertel, L., Phan, H., Mertins, A., 2016. Classifying variable-length audio files with all-convolutional networks and masked global pooling, 1607. arXiv preprint arXiv, 02857.

Hinton, G., et al., 2012. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. 29, 82–97.

Kong, Q., et al., 2016. Deep Neural Network Baseline for DCASE Challenge 2016. University of Surrey.

Lidy, T., Schindler, A., 2016. CQT-based convolutional neural networks for audio scene classification. In: Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016), vol. 90. IEEE Budapest.

Mesaros, A., et al., 2017. Detection and classification of acoustic scenes and events: outcome of the DCASE 2016 challenge. IEEE/ACM Transact. Audio Speech Lang. Proc. 26, 379–393.

Müller, M., 2015. Short-Time Fourier Transform and Chroma Features.

Nadeu, C., Macho, D., Hernando, J., 2001. Time and frequency filtering of filter-bank energies for robust HMM speech recognition. Speech Commun. 34, 93–114.

Pons, J., et al., 2017. End-to-end learning for music audio tagging at scale, 1711. arXiv preprint arXiv, 02520.

Rao, P., 2008. Audio Signal processing.' Speech, Audio, Image and Biomedical Signal Processing Using Neural Networks. Springer, Berlin, Heidelberg, pp. 169–189.

Saito, T., Rehmsmeier, M., 2015. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS One 10, e0118432.

Srivastava, N., et al., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 15, 1929–1958.

Stowell, D., et al., 2015. Detection and classification of acoustic scenes and events. IEEE Trans. Multimed. 17, 1733–1746.

Subramanian, Hariharan, Rao, P., Roy, S.D., 2004. Audio Signal Classification, vol. 2004. EE Dept, IIT Bombay, pp. 1–5.

Virtanen, T., Plumbley, M.D., Ellis, D. (Eds.), 2018. Computational Analysis of Sound Scenes and Events. Springer, Heidelberg.

Vu, T.H., Wang, J.-C., 2016. Acoustic scene and event recognition using recurrent neural networks. Detect. Classif. Acous. Scenes Even. 2016, 1–3.

Wei, S., et al., 2018. Sample Mixed-Based Data Augmentation for Domestic Audio Tagging, 1808. arXiv preprint arXiv, 03883.

Xu, Y., et al., 2014. A regression approach to speech enhancement based on deep neural networks. IEEE/ACM Transac. Audio Speech Lang. Proc. 23, 7–19.

Xu, Y., et al., 2016. Fully DNN-based multi-label regression for audio tagging, 1606. arXiv preprint arXiv, 07695.

Xu, Y., et al., 2017. Unsupervised feature learning based on deep models for environmental audio tagging. IEEE/ACM Transac. Audio Speech Lang. Proc. 25, 1230–1241.

Yi, D., Ahn, J., Ji, S., 2020. An effective optimization method for machine learning based on ADAM. Appl. Sci. 10, 1073.

Yun, S., et al., 2016. Discriminative training of GMM parameters for audio scene classification and audio tagging. IEEE AASP Chall. Detect. Classif. Acoust. Scenes Events 2016, 1–4.