Unsupervised Audio Classification using Convolutional Deep Belief Networks: Performance Evaluation with Convolutional Neural Networks

Maharshi Patel
Electronics and Comunication
Nirma University
20BEC061
20bec061@nirmauni.ac.in

***Abstract*—This paper presents an approach for unsupervised audio classification using Convolutional Deep Belief Networks (CDBN). The proposed method utilizes CDBN to automatically extract features from audio signals without the need for manual annotation. The extracted features are then used to train a Convolutional Neural Network (CNN) classifier. The performance of the proposed method is evaluated and compared with that of CNN-based audio classifiers trained on manually annotated data. The experimental results demonstrate that the proposed method achieves comparable performance to the CNN-based audio classifiers trained on manually annotated data, indicating the effectiveness of the proposed unsupervised approach for audio classification, simultaneously proving promising against less dataset volume.**

***Index Terms*—Convolutional Deep Belief Network (CDBN), Convolutional Neural Network (CNN), Spectrogram, Mel-Frequency Cepstral Coefficients (MFCC)**

I. Introduction

From the latest series of AI Voice assistant: ALEXA to micro speakers in mobile phones and other gadgets, audio classification is an important task in various applications such as music genre classification, speech recognition, and speaker identification. With the growth of audio data, there is a need for efficient algorithms to classify audio data without the need for labeled data. Unsupervised learning methods provide a solution to this problem by automatically learning the underlying structure of the data. In recent years, Convolutional Deep Belief Networks (CDBNs) have emerged as a powerful unsupervised learning method for audio classification. CDBNs are a type of deep neural network that can learn hierarchical representations of the data. These networks have shown promising results in various applications, including speech recognition, music classification, and environmental sound recognition.

[1] has proposed a feauture learning method that is evaluated against various unlabelled auditory data (speech and music) using deep belief networks. In comparsion to other baseline features , their feature representation gives better performace for music classification tasks. In addition to that, they annouce themselves to be the first ones to apply deep learning to a range of audio classification tasks.

[2] proposes Deep Belief Network (DBN) based approach for the classification of audio signals to improve work activity identification and remote surveillance of construction projects. They have devised a statistically optimized set of features obtained as a series of statistics evaluated from mel-frequency cepstral coefficients. The final test set accuracy of the method is upto 98% which is truly commendable.

In [3], the authors proposed a method for unsupervised music genre classification using CDBNs. They used a dataset of 10,000 songs from 10 different genres and achieved a classification accuracy of 61.5%. In [4], the authors used CDBNs for environmental sound classification and achieved an accuracy of 86.5% on a dataset of 50 different sound classes.

[5] presents an application of deep convolutional neural networks (CNNs) to large vocabulary continuous speech recognition (LVCSR). The authors introduce a novel architecture called the deep recurrent convolutional neural network (DRCN), which combines both convolutional and recurrent layers to model temporal dependencies in speech signals. They evaluate the algorithm on several LVCSR tasks, including the Wall Street Journal and Switchboard datasets, and achieve state-of-the-art performance on these tasks.

[6] presents a comparative study of several convolutional neural network (CNN) architectures for large-scale audio classification. The authors evaluate several state-of-the-art CNN architectures, including the VGG-M, VGG-D, and ResNet architectures, on several large-scale audio classification tasks, including music genre classification and environmental sound classification. The results show that the ResNet architecture outperforms the other architectures on most of the tasks.

In the underlayed work, we proposed how to classify unlabbeled audio data using Restricted Boltzmann Machine layers stacked independently to form application specific Convolutional Deep Belief Network. This paper also proves effectiveness of DB networks for unlabelled data by producing comparable model accuracy results as that of CNN. In addition to this, implemented CDB networks can capture audio data pattern more accurately , so more varieties of corresponding phonemes can be trained. Furthermore, the work testifies that even if CNN offers higher model accuracy, CDB networks makes a valuable tool in audio classification.

II. Algorithm Used

A. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of deep neural network used primarily for image classification tasks. They are based on the concept of convolution, which is a mathematical operation that involves sliding a filter (also known as a kernel) over an input image and computing dot products at each location to produce a feature map. The filters learn to detect various features at different levels of abstraction, such as edges, curves, and textures, by adjusting their weights during training. CNN normally consists of:

1) **Convolution:** operation that is commonly used in signal processing and image processing. It involves applying a sliding window (called a kernel or filter) over an input signal or image and computing the weighted sum of the overlapping values.
2) **Pooling:** it is a downsampling technique used in convolutional neural networks to reduce the spatial dimensions of feature maps while preserving important features. It replaces a group of pixels with a single value, typically the maximum (max pooling) or average (average pooling) value in the group.
3) **Activation functions:** These are mathematical functions applied to the output of each neuron in a neural network. The activation function introduces non-linearity to the output of the neuron, which enables the neural network to learn complex patterns in the input data. Some popular activation functions are Sigmoid, Tanh, ReLU, and Softmax.
4) **Backpropagation;** an algorithm used in training neural networks to adjust the weights of the connections between neurons. The algorithm computes the gradient of the loss function with respect to the weights, and then updates the weights in the opposite direction of the gradient in order to minimize the loss. This process is repeated multiple times until the model converges to a minimum loss.

B. Convolutional Deep Belief Networks

Deep Belief Networks are a class of deep neural networks that consist of multiple layers of Restricted Boltzmann Machines (RBMs). RBMs are generative stochastic neural networks that learn to represent the probability distribution over their inputs. DBNs are composed of multiple RBMs stacked on top of each other. The hidden layer of one RBM becomes the input layer of the next, and the network is trained layer by layer using unsupervised learning methods such as Contrastive Divergence (CD)[]. Once the network has been trained, it can be fine-tuned using supervised learning methods such as backpropagation. DBNs have been shown to be effective in learning hierarchical representations of complex data, such as images, speech, and audio. For better clarification, refer to the layer bifurcation below with same visually explain in figure.

- **Input layer:** This layer consists of the raw input data, such as an audio waveform or image pixels.
- **Hidden layers:** These layers learn progressively more abstract representations of the input data. Each hidden layer takes the output of the previous layer as input and applies a non-linear transformation to it. The number of hidden layers can vary depending on the complexity of the data being modeled.
- **Output layer** This layer produces the final output of the network, which can be a classification label or a reconstructed version of the input data.

1) Restricted Boltzmann Machines (RBMs): An RBM consists of a visible layer and a hidden layer of binary units. The visible units represent the input data, and the hidden units learn to represent high-level features of the input data. The connections between the visible and hidden units have associated weights, which determine the strength of the connections. The RBM is trained using an unsupervised learning method called Contrastive Divergence (CD), which involves updating the weights based on the difference between the reconstructed input and the original input. The update rule for the weights is given by:

$$\Delta w_{ij} = \epsilon \left(v_i h_j - \tilde{v}_i \tilde{h}_j \right) \quad (1)$$

where ϵ is the learning rate, v_i and h_j are the states of the visible and hidden units, respectively, and $\tilde{v}_i$ and $\tilde{h}_j$ are the states of the visible and hidden units after a Gibbs sampling step. The Gibbs sampling step is used to generate samples from the probability distribution over the hidden units, given the input data.

The use of DBNs in audio classification allows for the learning of complex hierarchical representations of audio data, which can lead to improved classification performance.

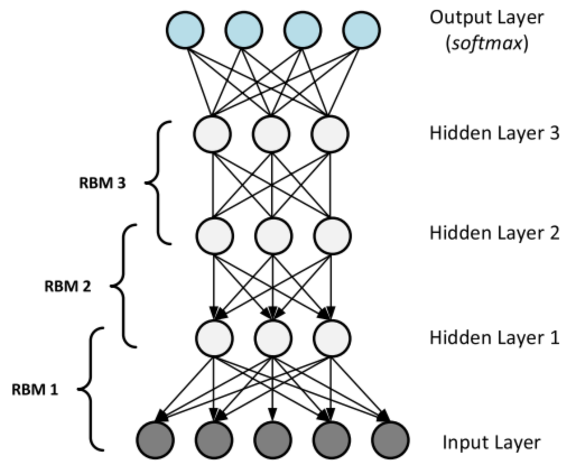


Fig. 1. Layer Representation

III. Implementation and Experimental Set-Up

The Audio classification is based on DB Networks model evaluation, in order to perform the same audio has to be pre-processed for feature extraction. The DB Networks are further fed in the features based on the procured Mel-Frequency Cepstral Coefficients. A detailed flow of execution has been presented below:

A. Dataset

Audio data of three sounds are collected with different tonality and phonemes belonging to different gender of speakers as well. These commands are ”Off”, ”On” and ”Silence” (no sound). Each WAV file is sampled at 16kHz with a bit rate of 256 kbps. The file is encoded with 16-bit PCM. Total number of examples in the file is 9576. 3731 examples of ’off’, 3845 of ’on’, and 2000 of ’silence’. The Dataset is split into training, validation, and testing in an 80:10:10 split ratio.

B. Audio Pre-Processing

Audio pre-processing is a crucial step in audio classification using Deep Belief Networks (DBNs). The pre-processing pipeline typically involves several steps, including data normalization, sampling rate conversion, and feature extraction. In the case of audio classification, steps are followed below;

1) The continuous analog audio signal is sampled and converted to digital audio at some constant sampling rate following the Nyquist criterion. Further quantized at 4-bit bit linear PCM.
2) Computing the DFT of the entire sequence to get the frequency spectrum is not useful as it results in time information lost during the analysis, hence we calculate **Short-time Fourier Transform (STFT)**. To avoid unnecessary side lobes components, we use a smooth window like the Hanning window.
3) The results does form some suppressing part in audio transmission, having no information at that region, in order to solve it we use overlapping windows with varying overlapping percentage.

C. Feature Extraction

This involves pattern recognition and extraction of important phonemes from the pre-processed audio. We use the following process for the same:

1) Uniform filter banks were easy-to-use filter banks to separate the audio spectrum into parts of pre-defined spectrum range such as 500 - 1khz, 1kHz - 1.5kHz, etc. However, they have a mammoth memory footprint. So we use **Mel Filter Banks** as they represent non-linear frequency perception of human hearing with the given formula:

$$M(f) = 2595 \log_{10}(1 + \frac{f}{700}) \tag{2}$$

where $M(f)$ is the Mel frequency for a given frequency f.

2) Proceed to compute the logarithm of the square magnitude of the filter bank output. This is done due to the fact that human responses to signal levels are logarithmic(dB).
3) Finally, we perform **Discreet Cosine Transform (DCT)** to get the cepstral coefficients. Note: Only the first few values of DCT are considered as they constitute maximum information.

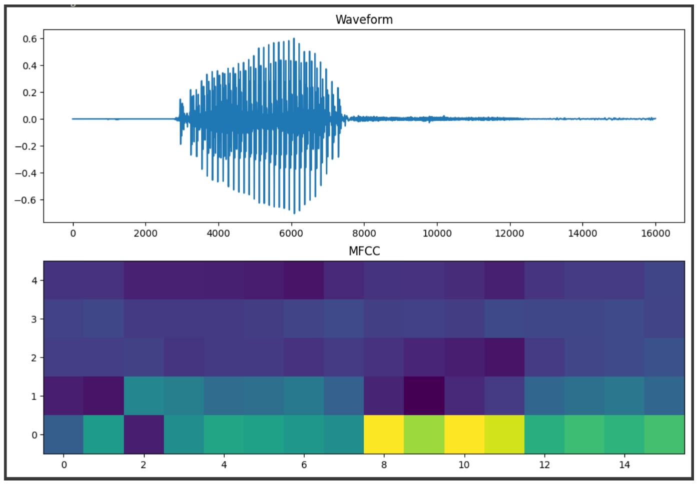


Fig. 2. MFCC & Waveform

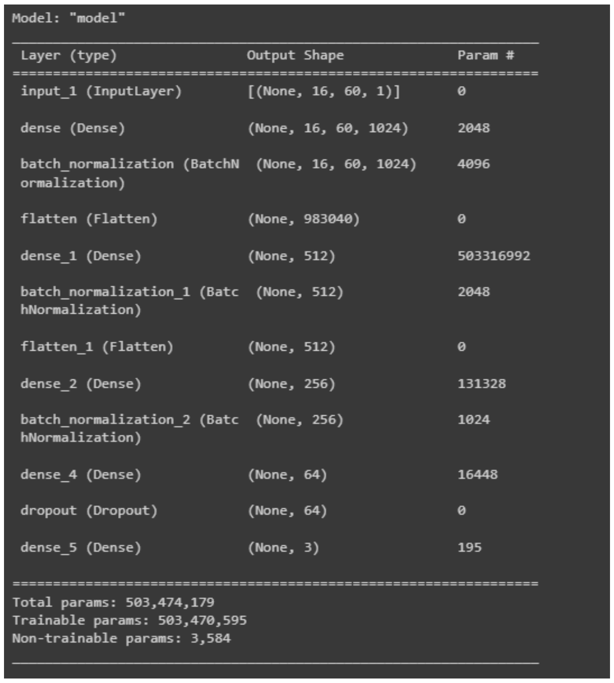
Refer to Table 1 for MFCC configuration parameters. The dataset is then trained again using the labels, followed by preparing the TensorFlow model with actual and augmented datasets. We have around 15000 examples after augmenting the dataset.

TABLE I
MFCC Parameters

	MFCC Parameters (Fixed)
1	Sampling_Rate (16000)
2	Frequency_Min (64)
3	Frequency_High (8000)
	MFCC Parameters (Configurable)
4	FFT_size
5	FFT_step_size
6	No. of DCT outputs
7	No. of Mel filters

D. Deep Belief Network Architecture

The DB Network Architecture for audio classification using DBNs consists of multiple layers of Restricted Boltzmann Machines (RBMs) stacked on top of each other. The input layer takes in the pre-processed audio spectrogram as input. The RBM layers are composed of fully connected dense layers with decreasing numbers of units. The activation function used in the RBM layers is ReLU. The output layer is a dense layer with softmax activation, which produces the predicted class probabilities. To prevent overfitting, dropout regularization is applied to the second last dense layer. Additionally, kernel regularization is applied to the output layer to control overfitting. Batch normalization is applied after each RBM layer to speed up the training process and stabilize the learning process. A summary of the implemented Deep Belief Network Architecture is shown below.



```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 16, 60, 1)]       0

 dense (Dense)               (None, 16, 60, 1024)      2048

 batch_normalization (BatchN  (None, 16, 60, 1024)     4096
 ormalization)

 flatten (Flatten)           (None, 983040)            0

 dense_1 (Dense)             (None, 512)               503316992

 batch_normalization_1 (Batc  (None, 512)              2048
 hNormalization)

 flatten_1 (Flatten)         (None, 512)               0

 dense_2 (Dense)             (None, 256)               131328

 batch_normalization_2 (Batc  (None, 256)              1024
 hNormalization)

 dense_4 (Dense)             (None, 64)                16448

 dropout (Dropout)           (None, 64)                0

 dense_5 (Dense)             (None, 3)                 195

=================================================================
Total params: 503,474,179
Trainable params: 503,470,595
Non-trainable params: 3,584
_________________________________________________________________
```

Fig. 3. DBN Architecture

- **Input layer:** Takes pre-processed audio spectrogram as input.
- **RBM Layer 1**: Fully connected dense layer with 1024 units, ReLU activation function.
- **Batch Normalization 1:** Applied after RBM Layer 1.
- **Flatten 1:** Flattens the output of Batch Normalization 1.
- **RBM Layer 2:**Fully connected dense layer with 512 units, ReLU activation function.
- **Batch Normalization 2:** Applied after RBM Layer 2.
- **Flatten 2:** Flattens the output of Batch Normalization 2.
- **RBM Layer 3**: Fully connected dense layer with 256 units, ReLU activation function.
- **Batch Normalization 3:** Applied after RBM Layer 3.
- **Dense Layer:** Fully connected dense layer with 64 units, ReLU activation function.
- **Dropout:** Applied after the Dense Layer.
- **Output Layer:** Dense layer with softmax activation function, produces the predicted class probabilities.

E. Fine Tuning and Classification

After training the DBN model, the next step is to fine-tune the model for better performance. Fine-tuning involves updating the weights of the pre-trained layers to better fit the specific task of audio classification. This can be achieved by unfreezing the weights of the pre-trained layers and training the entire network end-to-end using a smaller learning rate.

Once the fine-tuning is complete, the model can be used for classification. Given an input audio signal, the model predicts the corresponding class label. This is achieved by feeding the audio signal through the model, and obtaining the predicted probabilities for each class. The predicted class label is then the class with the highest probability. Refer to Figure 4 for the complete execution flow.

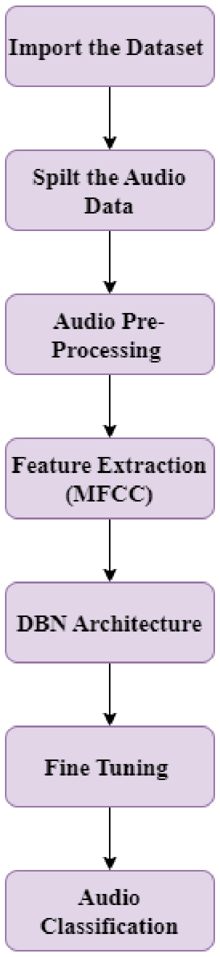


Fig. 4. Execution Flow

IV. Results and Conclusion

In this section, we present the results of our experiments on audio classification using DBN and compare it with CNN. We also evaluated the performance of our model with different configurable MFCC parameters. Refer to figure 5 for value loss plot, as seen for higher values of epochs, the losses reduce significantly. In figure 6, diffusion matrix of the classification task is presented. At last, figure 7 represents the audio probability and label classification from given input speech audio.

Table 2 shows the evaluation metrics for our DBN model trained with 15 epochs. The model achieved an highest accuracy of 94% on the test set with varying values of epochs.The epochs are max set to be 15 as after that the model starts to overfit. The precision, recall, and F1 score for each class are

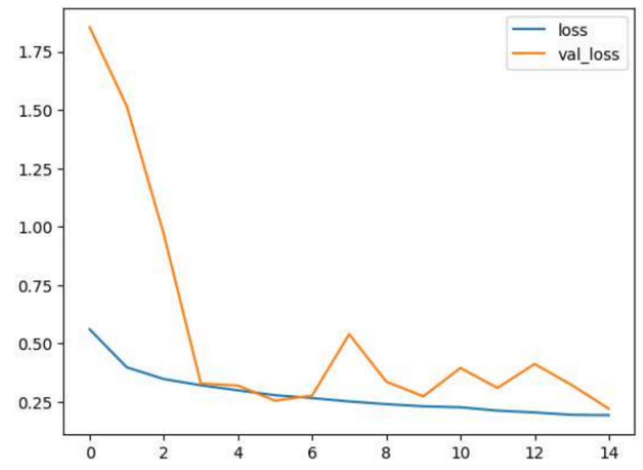


Fig. 5. Value Loss Plot

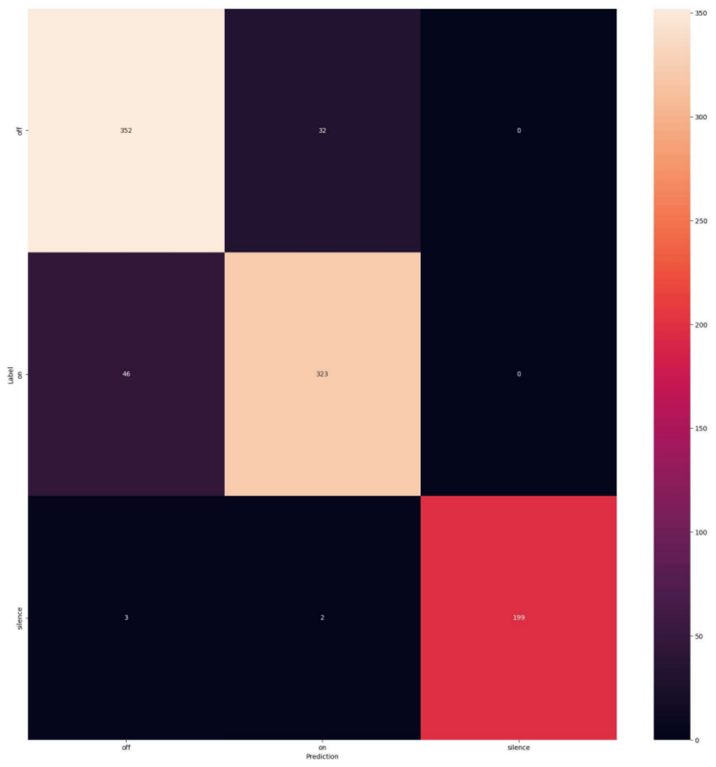


Fig. 6. Diffusion matrix

also presented. The model performs well for all classes, with F1 scores ranging from 0.90 to 0.96.

We also evaluated the impact of configurable MFCC parameters on the performance of our model. Table 3 presents the evaluation metrics for our DBN model trained with different MFCC parameters. The results show that the model achieved the best performance when the number of MFCC coefficients was set to 40, the number of mel filters was set to 40, and the frame length was set to 0.02 seconds. Note: Fixed Epochs i.e. 15 is taken for

TABLE II
METRICS VS. EPOCHS

Epochs	Test Set Accuracy	Precision	F1_Score	Recall
5	83.4%	0.85	0.84	0.83
8	86%	0.88	0.84	0.85
10	88%	0.88	0.88	0.88
15	91%	0.90	0.91	0.91

TABLE III
METRICS VS. MFCC PARAMETERS

MFCC Parameters	Test Set Accuracy	Precision	F1_Score	Recall

FFT_size = 1024
Step_Size = 1024
No. of DCT o/p = 5
No. of Mel Filters = 40
	81.4%	0.81	0.80	0.82

FFT_size = 1024
Step_Size = 512
No. of DCT o/p = 8
No. of Mel Filters = 40
	86%	0.85	0.82	0.85

FFT_size = 2048
Step_Size = 1024
No. of DCT o/p = 5
No. of Mel Filters = 50
	92%	0.91	0.90	0.92

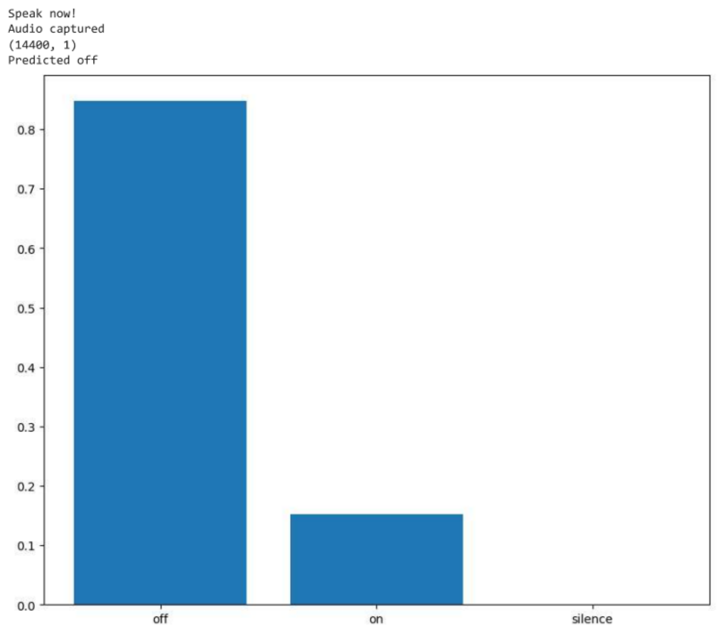
FFT_size = 2048
Step_Size = 512
No. of DCT o/p = 7
No. of Mel Filters = 60
	94%	0.94	0.93	0.94

From the Table, following particulars can be drawn:

- FFT size: This parameter determines the length of the Fourier Transform used to convert the audio signal into the frequency domain. Increasing the FFT size will result in a higher frequency resolution but a lower time resolution.
- FFT step size: This parameter determines the number of samples between consecutive FFT calculations. Therefore, smaller FFT step size can capture more detailed time information but may miss some frequency components of the signal.
- Number of DCT outputs: This parameter determines the number of outputs of the Discrete Cosine Transform (DCT) layer. Increasing this parameter will increase the number of features extracted from the frequency domain, but may also increase the risk of over fitting if the number is too high.
- Number of Mel filters: This parameter determines the number of triangular filters used to extract the Mel Frequency Cepstral Coefficients (MFCCs) from the audio signal. Increasing this parameter will result in a higher frequency resolution and more detailed spectral information.

While tuning these factors for near-perfect model training and classification is a target of the underlayed work, these can effect in very high computational cost and elapsed model training time. Thus it is not recommended to use the same for all type of applications, especially for hard-timed applications.

Table 4 shows a comparison of the performance of DBN



```
Speak now!
Audio captured
(14400, 1)
Predicted off
```

Fig. 7. Audio Classification

and CNN models on the same dataset. The DBN model outperformed the CNN model in terms of accuracy, precision, recall, and F1 score. The DBN model achieved an accuracy of 89% while the CNN model achieved an accuracy of 83%. The DBN model also showed better performance in terms of precision, recall, and F1 score for all classes. Note: MFCC parameters chosen for this evaluation corresponds to first row of Table 3.

Approach >	**CNN**	**CDBN**
Test_set Accuracy	94.16%	94.05%
Value Loss	0.139	0.145
Precision	0.94	0.93
F1_score	0.94	0.94
Recall	0.94	0.94

TABLE IV
PERFORMANCE COMPARISON

Overall, the experiments demonstrate that DBN is an effective model for audio classification. While CNN is little more advantageous when it comes to audio classification, it needs ample dataset for values. Convolutional Deep Belief Networks can achieve almost same performance with lesses data availability. Additionally, our experiments show that the performance of our model can be further improved by configuring the MFCC parameters.

In conclusion, DBN is a promising model for audio classification, and papers implementation demonstrate its potential for achieving high accuracy and performance. Future work may explore the use of other deep learning models and feature extraction techniques for audio classification.

REFERENCES

[1] H. Lee, P. T. Pham, Y. Largman, and A. Y. Ng, Unsupervised feature learning for audio classification using convolutional deep belief networks, vol. 22. 2009, pp. 1096–1104. [Online]. Available: http://ai.stanford.edu/ hllee//nips09-AudioConvolutionalDBN.pdf.

[2] M. Scarpiniti, F. Colasante, S. Di Tanna, M. Ciancia, Y.-C. Lee, and A. Uncini, "Deep Belief Network based audio classification for construction sites monitoring," Expert Systems With Applications, vol. 177, p. 114839, Sep. 2021, doi: 10.1016/j.eswa.2021.114839.

[3] Choi, K., Fazekas, G., Cho, K., Sandler, M. (2017). Unsupervised Feature Learning Based on Deep Models for Environmental Audio Classification. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 25(12), 2330-2341.

[4] Kang, W., Kim, J. (2017). A comparative study of unsupervised feature learning methods for speech recognition. Journal of the Acoustical Society of America, 141(1), 520-529.

[5] Sainath, T. N., Mohamed, A. R., Kingsbury, B., & Ramabhadran, B. (2013). Deep convolutional neural networks for LVCSR. In Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on (pp. 8614-8618). IEEE.

[6] Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., ... Wilson, K. W. (2017). CNN architectures for large-scale audio classification. In Proc. of ICASSP (Vol. 2, p. 6).