# Audio Classification using Braided Convolutional Neural Networks

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

*Harsh Sinha*[1]*, Vinayak Awasthi*[2]*, Pawan K Ajmera*[2*]

*[1] Department of Computer Science and Information Systems, Birla Institute of Technology and Science Pilani, Pilani, 333031, India*
*[2] Department of Electrical and Electronics Engineering, Birla Institute of Technology and Science Pilani, Pilani, 333031, India*
** E-mail: pawan.ajmera@pilani.bits-pilani.ac.in*

**Abstract:**Convolutional Neural Networks (CNNs) work surprisingly well and has helped drastically enhance the state-of-the-art techniques in the domain of image classification. The unprecedented success motivated the application of CNNs to the domain of auditory data. Recent publications suggest Hidden Markov Models (HMMs) and Deep Neural Networks (DNNs) for audio classification. This paper aims to achieve audio classification by representing audio as spectrogram images and then use a CNN-based architecture for classification. The paper presents an innovative strategy for a CNN-based neural architecture that learns a sparse representation imitating the receptive neurons in primary auditory cortex in mammals. The feasibility of the proposed CNN-based neural architecture is assessed for audio classification task on standard benchmark datasets such as Google Speech Commands datasets (GSCv1 and GSCv2) and UrbanSound8K dataset (US8K). The proposed CNN architecture referred to as Braided Convolutional Neural Network (BCNN) achieves 97.15%, 95% and 91.9% average recognition accuracy on GSCv1, GSCv2 and US8K datasets respectively outperforming other deep learning architectures.

## 1 Introduction

Content-based classification refers to inspection of a data stream for useful information. In context to audio signals, the term is specifically used for recognizing sounds or voice commands in an audio stream. This research area is focused on two major applications namely speech/voice recognition and environmental sound recognition. The difference in the two domains lies in the fact that human speech (for example music and verbal speech) involves classification of strongly structured and organized audio samples whereas environmental sound classification involves semi-structured audio samples.

In perspective of speech recognition, Google offers smart assistants and Keyword spotting (KWS) systems on mobile devices allowing its users to search by voice [1]. There are also speech recognition applications for the disabled community [2]. On the other hand applications for environmental sound classification range from surveillance [3], acoustic event analysis [4], health, hygiene and smart homes [5].

Researchers have used Hidden Markov Model (HMM) [6], matrix factorization [7], Hough Transform [8] and Radon Transform [9] to the domain of audio classification. Such methods learn simple representations of data and they require task-specific modifications.

Developments in parallel processing such as advent of GPUs lead to burgeoning interest in the field of deep learning which aims at progressively extracting more complex, higher level representations from raw input. Among deep neural architectures, Convolutional Neural Networks (CNNs) have been one of the most successful architectures, especially in computer vision [10]. The primary cause which lead to proliferation of CNNs across various domains is its agility in reducing variations and extracting spatial correlations for large scale image recognition [11–14].

Motivated by magnificent achievements of CNNs, this paper investigates whether CNNs can be used for audio classification. This work explores the capability of CNNs to learn spectral correlations and to reduce spectral variations ultimately achieving accurate content-based audio classification.

As the deep-learning methods are pivoted to learn feature hierarchies, the potency of deep learning can be exploited by increasing its size i.e. depth (number of stacked layers) and width (number of layers at the same level) [15]. In theory, increasing the number of layers should not increase generalization error as the redundant layers should learn an identity mapping [14]. Thus, empirically deep networks emulate shallower counterparts to learn an optimal non-linear mapping. Such an approach results in high classification accuracy. However, the ability of a deep neural architecture to learn discriminative features by directly mapping input to output subsists on quantity and quality of data [16]. Inadequate amount of data would make searching optimal kernels for a deep architecture a cumbersome task. Training deep neural networks by iterating over limited data often leads to poor generalization. Recent publications have addressed the problem of degradation using regularization [17], weight normalization [18] and residual connections [14].

Integrating sparsity in the learning algorithm can fundamentally solve the problem of learning an optimal representation [19]. The primary auditory cortex in mammals have a sparse architecture [20]. The architecture is localized, oriented, sparsely associated, and systematically organized. Imitating natural sparse architecture of auditory cortex in mammals, it can be postulated that learning a sparse representation can efficiently process audio signals. This work is based on arriving at a optimal sparse architecture modeled using dense convolutional components.

Prior works have bifurcated the task of audio classification into two different domains of speech and environment acoustic events. This work focuses on learning an optimal cross-domain sparse network that can successfully be applied for audio classification in general. The proposed Braided Convolutional Neural Network (BCNN) outperforms other deep neural architectures without any modification in the architecture with respect to domain of audio signals.

The rest of the paper is organized as follows. Section 2 discusses the existing approaches for audio classification. In Section 3, an overview of the proposed methodology and CNN architectures is presented. Section 3.5 describes in detail the proposed BCNN architecture. The experimental setup, the results and comparative performance measures are described in Section 4. Section 5 includes a discussion of the observed results. Section 6 concludes the paper providing a brief summary of the work.

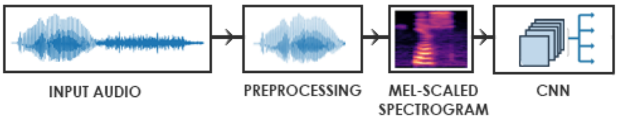

**Fig. 1**: Schematic diagram representing CNN-based audio classification

## 2 Related Work

Identification of speech commands in short audio segments has a wide range of applications especially with wide acceptance of speech-driven user interfaces. Apart from smart devices, the applications span to developing an OpenKWS system, content based search in conversations, sung-word recognition [21] and audio database indexing [22]. With development of IoT, researchers are using wireless sensors to analyze environmental sounds particularly for bird species recognition [23], obtain obstacle information for visually-impaired people [24], audio surveillance [25], whale sound categorisation [26] and automatic snore detection [27].

In past years, numerous researchers have applied different methods to achieve robust audio classification. To overcome primary obstacle of environmental and demographic variations researchers have used techniques such as Hidden Markov Model (HMM) [6], matrix factorization [7], i-vector [28, 29], Hough Transform [8], Radon Transform [9], Restricted Botlzmann Machines (RBM) [29], Deep Neural Networks (DNNs) [30, 31] and CNNs [32, 33] to the domain of audio classification.

Researchers have also focused on representing audio in the form of spectrograms for its classification. Costa et al. [34] extracted local binary patterns from time-frequency spectrograms. Nanni et al. [35] extracted visual features from local windows of a spectrogram generated by Mel-scale zoning with an ensemble of SVM classifiers. In their subsequent work [36], they combined visual and acoustic features which boosted the classification accuracy. Researchers have also used state-based models such as Dynamic Time Warping [37] and Hidden Markov Models [30]. However, rather than employing a time-variant approach, the audio was represented as a sequence of spectrogram images.

GMM-HMMs [38] have been used extensively for automatic speech recognition. In theory, GMM-HMMs can model probabilistic distribution to complete gamut of precision [30]. Consequently, there had been exclusive focus on constraining GMMs, in order to enhance their speed and accuracy. With advent of high-performance computing systems, DNNs have proven to perform exceptionally better than GMMs in robust modeling of audio recognition systems, especially in terms of implementation, evaluation time, latency and memory footprint [31]. DNNs provide more flexibility in feature representation and they tend to perform more efficiently than GMMs especially with large datasets and large vocabulary [32]. In addition, DNN model size can be appropriately constrained so that it can be deployed directly on end-devices.

However, DNNs have significant drawbacks. First of all, DNNs disregard the spatial topology of the input [39]. Audio consists of strong correlations in structure with respect to frequency domain. The spatial correlations are not utilized by DNNs as they inherently don't model topology of the input. Moreover, DNNs are not invariant to translational variances in audio signals. Although, adequate number of parameters in DNN architecture and sufficient training time would allow the network to achieve translational invariance, the network would be dense. Hence, it would dramatically increase computation and complexity [39].

Therefore, recent works [40, 41] have revolved around using CNNs with spectrograms for audio classification. CNNs have shown improved efficiency as they account for the spatial differences in the input by using a sparse locally connected structure [42]. They can model time and frequency components between adjacent audio samples. Thus, CNNs have outperformed DNNs in modeling audio classification systems. Depth is of prime importance for deep CNNs to extract relevant high-dimensional features. However, the results presented by Sainath and Parada [1] in context to speech recognition demonstrate that increasing depth unnecessarily may degrade the performance of learning algorithm. He et al. [14] tackles the problem of learning very deep neural architectures by introducing shortcut connections. It can be inferred that it is important to progressively extract complex but also coherent feature representations. Deep architectures suffer from vanishing gradients as it reaches the end of neural architecture for classification [43]. Inspired by tremendous success of CNNs [11, 13], this paper investigates the ability of deep CNNs to model spectral correlations and to reduce spectral variations for audio classification. The proposed model proposes braided-connectivity of convolutional layers to push features extracted from the various layers for efficient classification.

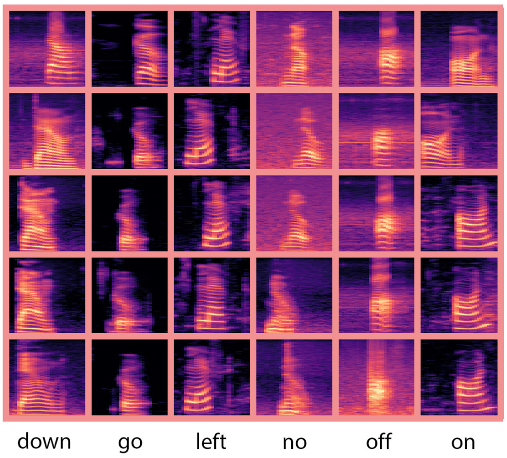

**Fig. 2**: Illustration of mel-spectrogram from the audio files in GSCv1 dataset for various speech commands exhibiting intra-class variations and inter-subject similarity.

## 3 Proposed Methodology

Figure 1 represents the schematic of the proposed methodology. The proposed methodology is represented as three major components: preprocessing, mel-scaled spectrogram generation and classification.

### *3.1 Preprocessing*

The input audio signal is re-sampled to 8 kHz at the pre-processing step. Re-sampling is applied to reduce dimensionality of the input signal. In addition, every sample is padded with zeros to guarantee uniformity in input data. Zero padding preserves spatial size without influencing learning algorithm in a biased way.

### *3.2 Spectrogram Generation*

The pre-processed raw audio waveform is transformed to a 2-dimensional image known as a spectrogram. A spectrogram can be understood as a 2-dimensional feature map representing frequencies with respect to time [9]. The human ear perceives frequencies on a logarithmic scale. Hence, the frequency scale is changed to mel-scale thereby converting a regular spectrogram to mel-spectrogram.

The obtained spectrogram is resized to (96 × 96) before feeding for classification as reducing the dimension of the input before spatial aggregation leads to faster training without much loss of spatial representation [44]. Figure 2 depicts an illustration of mel-spectrogram images generated from the audio samples for different classes such as down, go, left, no, off and on.

### *3.3 Standard CNN architecture*

The initial layer of a CNN represents the input image, $I\epsilon\mathbb{R}^{s\times s\times c}$ where $s$, $c$ are image size and number of channels respectively. A kernel $K\epsilon\mathbb{R}^{m\times m\times n}$ is convolved with initial layer $I$ to generate $k$ feature maps $F\epsilon\mathbb{R}^{(s-m+1)\,\times\,(s-m+1)\,\times\,k}$. A kernel (or filter) is shared across patches of previous layer giving rise to a locally connected structure leading to translational invariance. Each convolutional layer is succeeded by a subsampling or pooling layer which extracts important information while reducing spatial resolution leading to a compact representation of data. To ensure that the output cannot be reproduced from an affine transformation of data, a non-linear activation is applied. After several alternating convolutional and sub-sampling layers, a fully connected layer is employed to predict the output based on posterior probabilities. The goal is to learn suitable kernels using back-propagation to reduce the difference between predicted outputs and ground truth.

### *3.4 Motivation and Considerations*

As explained in Section 3.3, CNNs consist of several alternating convolutional and sub-sampling layers, and a fully connected layer to predict the posterior probabilities. Thus, CNNs form a generalized linear model (GLM) for the underlying feature maps. However, the abstraction can be improved using a "micro-network" [45] replacing GLM. The "micro-network" emulates a general non-linear function enhancing the abstraction obtained by GLM. The proposed BCNN utilizes the same idea by using kernels (3 × 3, 5 × 5) at the same level and repeating the block (referred to as bead in Section 3.5) sequentially.

DNNs use fully-connected layersat all levels which leads to a dramatic increase in the computational cost. A locally-connected structure can be efficiently used to alleviate the issue of computational cost [13]. CNNs use locally-connected shared kernels for convolutions, allowing us to learn a sparse representation. This hypothesis is based on theoretical results proven by Arora et al. [46] indicating that correlated inputs would concentrate in small local regions. The results show that an optimal network for accurate classification can be constructed if an over-specified neural network is used to learn the probability distribution of the dataset. Moreover, the use of ReLU non-linear activation function leads to sparse feature maps naturally [47]. The proposed BCNN approximates optimal sparse structure by utilizing available dense computations with uniformity of architecture, non-linear activation function (ReLU) and a large number of filters.

**Table 1** The table summarizes the convolutional neural architecture of a bead. The proposed Braided Convolutional Neural Network (BCNN) architecture consists of 4 such beads connected sequentially.

| *Layer Name* | *Layer Type* | *Patch Size* | *Linked to* |
|---|---|---|---|
| *Layer* 1 | *Convolution* | 1 × 1 | *Input* |
| | *Convolution* | 3 × 3 | *Layer* 1 |
| *Layer* 2 | *Convolution* | 1 × 1 | *Input* |
| | *Convolution* | 5 × 5 | *Layer* 2 |
| *Layer* 3 | *Max − Pooling* | 3 × 3 | *Input* |
| | *Convolution* | 1 × 1 | *Layer* 3 |
| *Append* 1 | *Concatenate* | | *Layer* 1 |
| | | | *Layer* 2 |
| *Layer* 4 | *Convolution* | 1 × 1 | *Append* 1 |
| | *Convolution* | 5 × 5 | *Layer* 4 |
| *Layer* 5 | *Convolution* | 1 × 1 | *Append* 1 |
| | *Convolution* | 3 × 3 | *Layer* 5 |
| *Append* 2 | *Concatenate* | | *Layer* 3 |
| | | | *Layer* 5 |
| *Layer* 6 | *Max − Pooling* | 3 × 3 | *Append* 2 |
| | *Convolution* | 1 × 1 | *Layer* 6 |
| *Append* 3 | *Concatenate* | | *Layer* 4 |
| | | | *Layer* 6 |

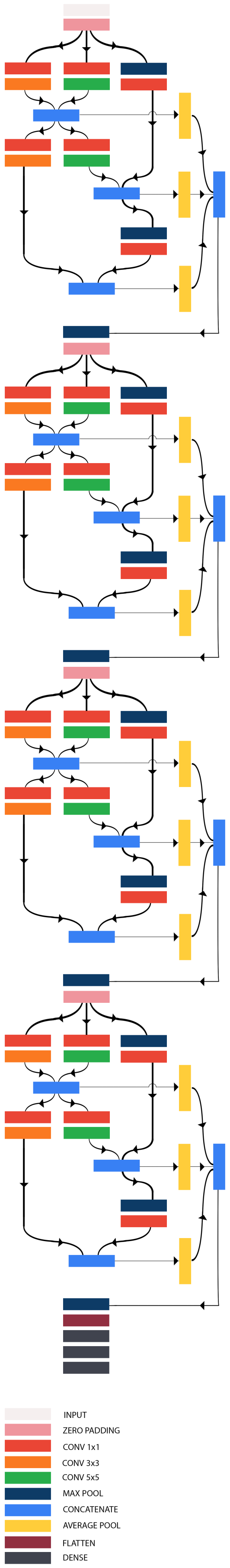

**Fig. 3**: Proposed CNN (BCNN) architecture. The different blocks represent different convolutional and max-pooling layers as shown in the colour map (legend)

Thus, integrating sparsity in the learning algorithm can fundamentally solve the problem of learning an optimal representation [19]. The primary auditory cortex in mammals have a sparse architecture [20]. Imitating the natural sparse architecture of auditory cortex in mammals, by learning a sparse representation can efficiently process audio signals.

Even though depth is very important for deep neural architectures, the results presented by Sainath and Parada [1] in context to speech recognition demonstrate that increasing depth unnecessarily may further degrade the performance of learning algorithm. He et al. [14] tackles the problem of learning very deep neural architectures by introducing shortcut connections. Thus, it can be inferred that it is important to progressively extract complex but also coherent feature representations.

### *3.5 Implementation Details*

The proposed Braided Convolutional Neural Network (BCNN) architecture consists of 4 similar structures (referred to as a bead) connected sequentially as shown in Figure 3. The architecture of a single bead is summarized in Table 1. Each bead involves several convolutional and max-pooling layers connected in a braided fashion.

As explained in Section 3.4, learning algorithm utilizes multiple dense connections of standard kernel size $3 \times 3$ and $5 \times 5$. Each bead makes an effective use of kernel size $1 \times 1$ to reduce computational complexity. SigOpt API was used for defining the best hyperparameters such as the number of filters and the number of layers. SigOpt is an AutoML solution which uses a bayesian method to construct a feedback mechanism between model output and different values for hyperparameters. Thus, the model can be tuned by selecting the best network parameters to maiximize performance [48].

A substantially deep neural network suffers from performance degradation [14]. Nonetheless, depth is very crucial for a deep neural network. The proposed methodology addresses the problem of degradation by improving information flow between consecutive layers of each bead. Each bead consists of three pairs i.e. (i) $convolution\ 1 \times 1\ convolution\ 3 \times 3$ , (ii) $convolution\ 1 \times 1\ convolution\ 5 \times 5$ and (iii) $max - pooling\ 3 \times 3\ convolution\ 1 \times 1$. The three sets of extracted feature maps are concatenated in different combinations. Braiding feature maps (as shown in Figure 3) preserves and increases the variance of the outputs, encouraging feature reuse. The proposed architecture of a single bead (as shown in Table 1) consists of ${}^3C_2$ combinations of the three different pairs as explained above. The outputs of ${}^3C_2$ combinations of convolutional layers are concatenated using average-pooling before feeding the feature maps to the following bead.

Each bead although have similar structure consist of substantially increasing representation depths to achieve state-of-the-art benefits in terms of classification accuracy [44]. The spatial size is decreased gradually to avoid extreme compression at the penultimate layer to fully connected layers. In order to guarantee proper concatenation of layers all the feature maps are zero-padded to maintain spatial size in consecutive layers. Finally, the resultant feature maps obtained as output of the fourth bead are fed to a fully connected softmax layer for classification.

The aim of proposed deep neural architecture is to learn appropriate kernels (or filters) for accurate audio classification. Adadelta optimizer [49] is used to learn suitable kernels. Adadelta dynamically adjusts with time and uses first order information with least overhead computation loss beyond stochastic gradient descent (SGD). Adadelta is similar to Adagrad as it also aims to adapt learning rate. Agagrad accumulates all the past squared gradients which is very inefficient. Adadelta uses a window of decaying past squared gradients (referred to as moving average). The moving average of squared gradients is defined as in Equation 1.

$$\overline{g^2_{MA}} = \frac{g^2_M + g^2_M + \ldots + g^2_{M-(n-1)}}{n} = \frac{1}{n}\sum_{i=0}^{n-1} g^2_{M-i} \qquad (1)$$

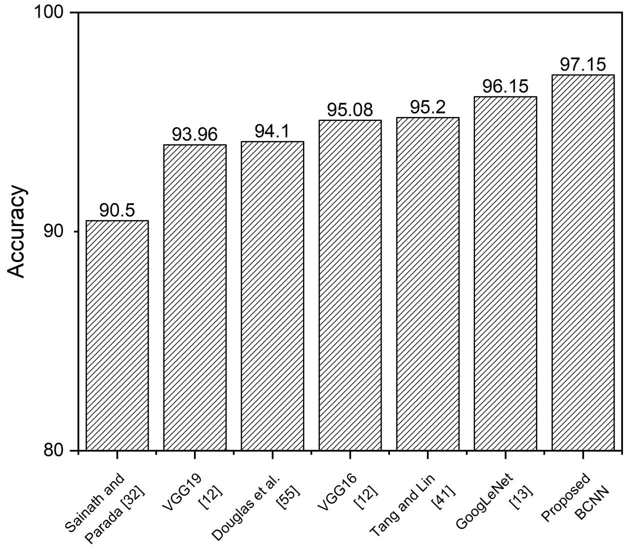

**Fig. 4**: Comparative performance of the proposed BCNN on GSCv1 dataset, in terms of average recognition accuracy.

For every new value, the simple moving average is updated, using Last-in-First-out Scheme (LIFO). The procedure is shown in Equation 2

$$\overline{g^2_{MA}} = \overline{g^2}_{MA,prev} + \frac{g^2_M}{n} - \frac{g^2_{M-n}}{n} \qquad (2)$$

However, Adadelta updates the moving average recursively decaying the average. Storing all the squared past gradients is an inefficient method. Adadelta defines moving average $\overline{g^2_{MA}}$ at step $t$ as in Equation 3.

$$\overline{g^2_{MA,t}} = \gamma \cdot \overline{g^2_{MA,t-1}} + (1-\gamma)\overline{g^2_t} \qquad (3)$$

The term $\gamma$ is analogous to momentum in Stochastic Gradient Descent (SGD). Finally, the parameters $\theta_t$ is updated as shown in Equation 4.

$$\Delta\theta_t = -\frac{\eta}{\sqrt{\overline{g^2_{MA,t}} + \epsilon}} g^2_t \qquad (4)$$

where $\eta$ refers to the learning rate.

## 4 Experimental Results

The following section explains the benchmark datasets, error estimation methods and specifications of parameters used for assessing feasibility of BCNN. Three benchmark datasets containing short audio files have been used for evaluation, namely $(i)$ Google Speech Commands Dataset (GSCv1), $(ii)$ Google Speech Commands Dataset (GSCv2), $(iii)$ Urban Sound 8K (US8K) datasets.

### *4.1 Datasets*

The first set of experiments were performed on the Google Speech Commands Dataset (GSCv1) which consists of 64,727 short audio clips of 30 english words [50]. The goal is to discriminate among speech commands such as yes, no, up, down, left, right, on, off, stop, go and unknown. The remaining 20 auxiliary words are designated as 'unknown'.

Similar to GSCv1 dataset, the Google Speech Commands Dataset (GSCv2) [51] consists of 105,829 one-second long audio of 35 english words. The dataset is used to discriminate among yes, no, up, down, left, right, on, off, stop, go, zero, one, two, three, four,

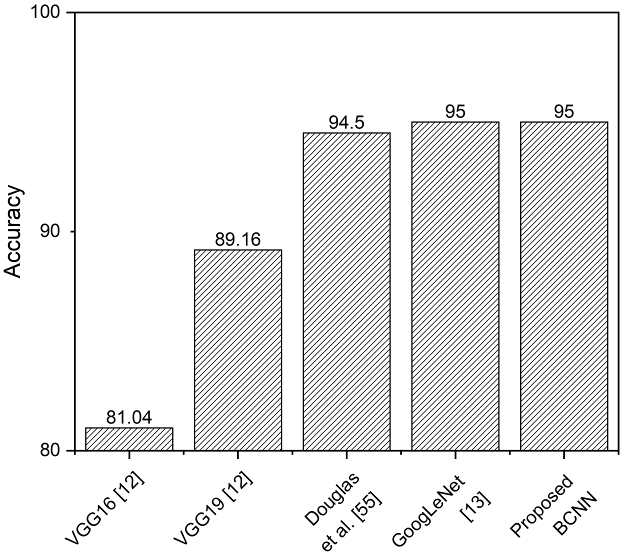

**Fig. 5**: Comparative performance of the proposed BCNN on GSCv2 dataset, in terms of average recognition accuracy.

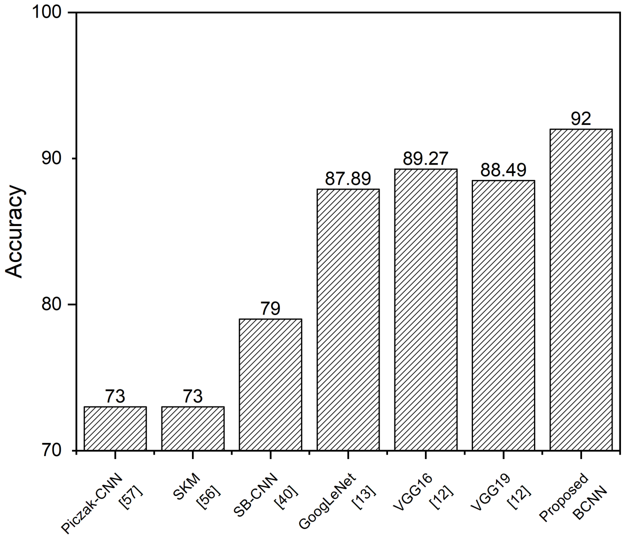

**Fig. 6**: Comparative performance of the proposed BCNN on US8K dataset, in terms of average recognition accuracy.

five, six, seven, eight, nine and unknown. The remaining 15 words are categorized into 'unknown' class.

The UrbanSound8K dataset [52] consists of 8732 sound clips upto 4 seconds in duration. In contrast to GSCv1 and GSCv2 datasets which contain voice commands, US8K consists of short environmental sounds. The task is to discriminate 10 sound classes:air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren and street music.

As discussed in Section 3, the original audio files are zero-padded and re-sampled (to 8 kHz). Further, the preprocessed audio signal is transformed to a mel-spectrogram. The spectrogram images are oblong. Hence, before feeding for classification spectrogram images are resized to 96 × 96. Moreover, reducing the dimension of the spectrogram before spatial aggregation leads to faster training without much loss of spatial representation [44]. GSCv1 and GSCv2 datasets contains much more data for 'unknown' class. Utilizing class weights while training prevented severe class distribution skews.

### 4.2 *Computational Complexity*

This section discusses the computational complexity for the proposed approach, a key aspect of voice-based authentication schemes.

In terms of complexity class of decision problems, a basic neural network with two layers each with three nodes and threshold activation is NP-complete [53]. He and Sun [54] introduced general formula for complexity for various convolution layers in a typical CNN.

Typically, CNNs use a massive network of mutual weights to automatically extract relevant features for accurate classification. This raises CNN's computational complexity. However, training CNNs have been practically tractable in various fields [11, 44]. Some techniques allowing *improper learning* are non-linear activation functions such as ReLU, over-specification, and weight-regularization.

The running time assessed significantly depends on the specific equipment hardware and software used to test CNN design. The experiments were undertaken on a workstation with CPU as Intel Core i7 and 6 GB Nvidia Geforce GTX 1060 GPU. Python scripts are based on TensorFlow and Keras. A single audio signal input ( 4 sec) executes in 1s, which involves generating spectrogram from an input file, prediction and rendering visualization.

### 4.3 *Speech Recognition Experiments*

With advent of speech-driven user interfaces, it is important to recognize pre-defined commands with exactness. Apart from smart devices, the applications span to developing an OpenKWS system or content based search in conversations.

The classification accuracy of the proposed methodology is presented in Figure 4 and 5 along with accuracy achieved by CNN [1], ResNet [41] and attention based Convolutional Recurrent Neural Network (CRNN) [55]. These proposed BCNN outperforms all the models on benchmark datasets.

DNNs have outperformed GMM-HMMs in the domain of audio classification [30]. However, DNNs suffer from high computational complexity due to dense connections in-between layers. Typically, CNN architectures perform pooling to limit the overall computation of the network. Sainath et al. [32] claim that typical CNNs perform pooling in the frequency domain which is not applicable for audio classification. They present a *fstride-CNN* which strides over frequencies achieving 27% improvement over DNN and 6% over typical CNNs in terms of recognition accuracy. Further, Tang and Lin [41] mirror a neural architecture based on ResNet [14] which outperforms *fstride-CNN* proposed by Sainath et al. [32] achieving 95.2% accuracy on GSCv1 dataset. A neural-attention based recurrent neural architecture is trained by Douglas et al. [55], which performs convolutions only in the time domain achieving 94.1% accuracy.

In this work, several benchmark models such as GoogLeNet [13], VGG16 and VGG19 [12] are trained, to assess the performance of CNNs on auditory data. GoogLeNet being much more memory efficient and substantially deeper than VGG, attains a better accuracy. GoogLeNet has a significantly different architecture than VGG allowing efficient training of 22 convolutional layers. This is possible by batch normalization and RMSprop. However, BCNN significantly outperforms other models achieving 97.15% and 95% average recognition accuracy in GSCv1 and GSCv2 respectively. Superior performance of proposed BCNN without using any data augmentation, regularization or batch normalization indicates that BCNN is less prone to overfitting.

### 4.4 *Environmental Sound Classification*

Data centers are accruing vast amounts of data, especially in perspective to smart cities. Environmental sound plays an important role in providing a holistic view of the city. With development of IoT, researchers are using wireless sensors to analyze environmental

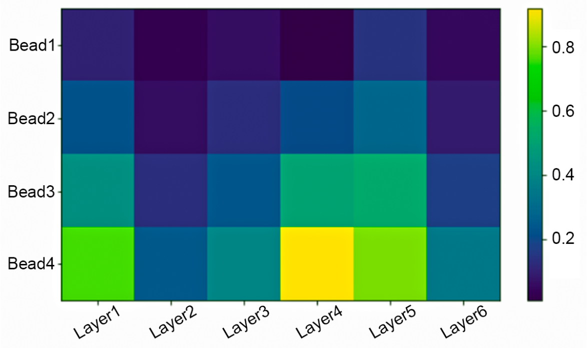

**Fig. 7**: Performance of the proposed BCNN on US8K, GSCv1 and GSCv2 datasets expressed in terms of CMC curve.

sounds particularly for bird species recognition [23], obtain obstacle information for visually-impaired people [24] and audio surveillance [25].

The experimental results of the proposed BCNN architecture on US8K Dataset is presented in Figure 6 along with the mean accuracy attained by SB-CNN [40], SKM [56] and PiczakCNN [57]. The figure also compares the different approaches with exemplar neural architectures such as VGG16, VGG19 [12] and GoogleNet [13].

SKM [56] presents a "'shallow" dictionary-learning based on spherical k-means. As the datasets for environmental sound classification is significantly smaller in size (87.64% smaller than GSCv1 dataset) limited variations [40], SB-CNN proposes an in-depth augmentation technique to train a CNN on the US8K dataset. PiczakCNN performs comparably to SB-CNN and SKM. However, the proposed BCNN significantly outperforms the techniques. In contrast to SB-CNN, proposed BCNN architecture uses a smaller (96 × 96) mel-spectrogram input image. SB-CNN uses a mel-spectrogram input image of size (128 × 128). This emphasizes that BCNN is much more efficient in progressively extracting higher level representations resulting in 16.5 % relative improvement in average accuracy. In addition, the proposed CNN doesn't use any explicit data augmentation (used in SB-CNN) or regularization (used in PiczakCNN). On that account, it can be stated that braided connectivity allows a CNN to learn optimal feature maps from mel-spectrograms realizing accurate classification.

Although GoogLeNet has much more efficient architecture than VGG16 and VGG19, it tends to overfit the US8K dataset. This can be attributed to fewer number of samples in comparison to GSCv1 and GSCv2 datasets. It also suggests that it is difficult to make efficient use of several neural architecture design principles used in GoogLeNet if the dataset is undersized. VGG16 and VGG19 are comparatively easier to train and it is effective for datasets in general.

## 5 Discussion

The following section explores several aspects of the proposed BCNN architecture. The proposed has architecture has significant modifications from the existing deep neural architectures used for audio classification which lead to superior recognition accuracy.

### 5.1 Feature Reuse

The proposed BCNN allows layer connectivity (as defined in Table 1), to extracted feature maps by the six preceding convolution layers in a single bead. An experiment was performed to analyze the use of this incentive by the BCNN network trained on GSCv2 dataset which estimates the average strength for each of the six convolutional layers. Figure 7 shows a heat map with six layers for all four beads and their respective weights. The average weight is a workaround to estimate a convolutional layer's reliance on its previous layers. There can be several observations from the Figure 7 :

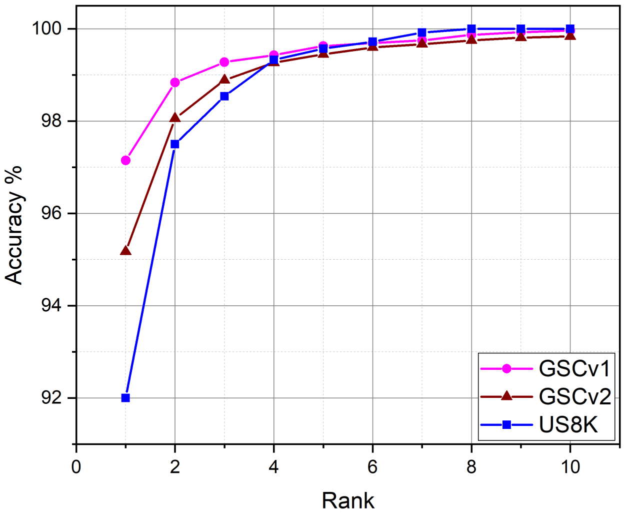

**Fig. 8**: Performance of the proposed BCNN on US8K, GSCv1 and GSCv2 datasets expressed in terms of CMC curve.

1. For each bead, the weights are distributed over multiple layers. This suggests that features extracted by the initial layers are used directly throughout the bead.
2. The lowest weights are designated for $Layer$ 2 and $Layer$ 6 meaning they obtain less relevant features as compared to other layers in the bead.
3. The final $Bead$ 4 depicted last row of the Figure 7 uses weights across the entire bead. It appears that the latter feature maps are clustered and indicate that more high-level features can indeed be generated late in the network.

### 5.2 CMC curves

Figure 8 presents the Cumulative Match Characteristic, used for assessing the closed-set identification performance of a model. Rank-$k$ denotes probability that the model predicts the correct label within top-$k$ predictions.

CMC curves are important especially with respect to audio classification as, in real scenarios there is a need to look at audio context for accurate predictions. For example, a smart assistant (trained for KWS task) can query the user to choose right course of action if user command is unclear. For a successful query to user, it is important the model has correct predictions within top-$k$ predictions. The proposed BCNN achieves very high rank-2 accuracy of 97.5%, 98.84% and 98.06% for US8K, GSCv1 and GSCv2 datasets respectively. This suggests that the proposed BCNN is promising in field of audio classification.

## 6 Conclusions

This paper evaluated a deep convolutional neural architecture for two-dimensional image classification of sound events using a mel-spectrogram representation. In particular, it assessed different standard architectures such as VGG16, VGG19 and GoogleNet on benchmark datasets such as GSCv1, GSCv2 and US8K. The experimental results confirm that by introducing sparsity and braided connectivity in consecutive layers, a CNN can efficiently learn spectral correlations eliminating environmental variations. BCNN achieves best average recognition accuracies in all three datasets irrespective of domain (environmental or speech commands) of audio samples. The key idea of the proposed novel architecture is the combination of sparsity with an efficient reuse of convolutional feature maps. It also suggests that a CNN can be used to imitate auditory neurons in mammals achieving improved results on competitive datasets.

## 7 References

1 Tara N Sainath and Carolina Parada. Convolutional neural networks for small-footprint keyword spotting. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

2 Andrej Škraba, Radovan Stojanović, Anton Zupan, Andrej Koložvari, and Davorin Kofjač. Speech-controlled cloud-based wheelchair platform for disabled persons. *Microprocessors and Microsystems*, 39(8):819–828, 2015.

3 Iulia Lefter, Léon JM Rothkrantz, and Gertjan J Burghouts. A comparative study on automatic audio–visual fusion for aggression detection using meta-information. *Pattern Recognition Letters*, 34(15):1953–1963, 2013.

4 Pasquale Foggia, Nicolai Petkov, Alessia Saggese, Nicola Strisciuglio, and Mario Vento. Reliable detection of audio events in highly noisy environments. *Pattern Recognition Letters*, 65:22–28, 2015.

5 Michel Vacher, Anthony Fleury, François Portet, Jean-François Serignat, and Norbert Noury. Complete sound and speech recognition system for health smart homes: application to the recognition of activities of daily living. *In-Tech*, pages 645–673, 2010.

6 J Robin Rohlicek, William Russell, Salim Roukos, and Herbert Gish. Continuous hidden markov modeling for speaker-independent word spotting. In *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, pages 627–630. IEEE, 1989.

7 Yong-Choon Cho and Seungjin Choi. Nonnegative features of spectro-temporal sounds for classification. *Pattern Recognition Letters*, 26(9):1327–1336, 2005.

8 Jonathan Dennis, Huy Dat Tran, and Eng Siong Chng. Overlapping sound event recognition using local spectrogram features and the generalised hough transform. *Pattern Recognition Letters*, 34(9):1085–1093, 2013.

9 Pawan K Ajmera, Dattatray V Jadhav, and Raghunath S Holambe. Text-independent speaker identification using radon and discrete cosine transforms based features from speech spectrogram. *Pattern Recognition*, 44(10-11):2749–2759, 2011.

10 J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

11 Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

12 Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

13 Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. Going deeper with convolutions. Cvpr, 2015.

14 Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

15 Gaurav Pandey and Ambedkar Dukkipati. To go deep or wide in learning? *arXiv preprint arXiv:1402.5634*, 2014.

16 Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

17 Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

18 Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015. URL `http://dl.acm.org/citation.cfm?id=3045118.3045167`.

19 Roger Grosse, Rajat Raina, Helen Kwong, and Andrew Y Ng. Shift-invariance sparse coding for audio classification. *arXiv preprint arXiv:1206.5241*, 2012.

20 Jos J. Eggermont. Chapter 3 - multisensory processing. In Jos J. Eggermont, editor, *Hearing Loss*, pages 71 – 90. Academic Press, 2017. ISBN 978-0-12-805398-0. doi: https://doi.org/10.1016/B978-0-12-805398-0.00003-7. URL `https://www.sciencedirect.com/science/article/pii/B9780128053980000037`.

21 Peerapol Khunarsa. Single-signal entity approach for sung word recognition with artificial neural network and time–frequency audio features. *The Journal of Engineering*, 2017(12):634–645, 2017.

22 Serkan Kiranyaz and Moncef Gabbouj. Generic content-based audio indexing and retrieval framework. *IEE Proceedings-Vision, Image and Signal Processing*, 153 (3):285–297, 2006.

23 Loris Nanni, Yandre MG Costa, Diego Rafael Lucio, Carlos Nascimento Silla Jr, and Sheryl Brahnam. Combining visual and acoustic features for audio classification tasks. *Pattern Recognition Letters*, 88:49–56, 2017.

24 Chieh-Li Chen, Yan-Fa Liao, and Chung-Li Tai. Image-to-midi mapping based on dynamic fuzzy color segmentation for visually impaired people. *Pattern Recognition Letters*, 32(4):549–560, 2011.

25 Marco Cristani, Manuele Bicego, and Vittorio Murino. On-line adaptive background modelling for audio surveillance. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, pages 399–402. IEEE, 2004.

26 Stavros Ntalampiras. Hybrid framework for categorising sounds of mysticete whales. *IET Signal Processing*, 11(4):349–355, 2016.

27 Kun Qian, Zhiyong Xu, Huijie Xu, Yaqi Wu, and Zhao Zhao. Automatic detection, segmentation and classification of snore related signals from overnight audio recording. *IET Signal Processing*, 9(1):21–29, 2015.

28 A. Nautsch, C. Rathgeb, R. Saeidi, and C. Busch. Entropy analysis of i-vector feature spaces in duration-sensitive speaker recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4674–4678, April 2015. doi: 10.1109/ICASSP.2015.7178857.

29 A. Nautsch, H. Hao, T. Stafylakis, C. Rathgeb, and C. Busch. Towards plda-rbm based speaker recognition in mobile environment: Designing stacked/deep plda-rbm systems. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5055–5059, March 2016. doi: 10.1109/ICASSP.2016.7472640.

30 Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

31 Guoguo Chen, Carolina Parada, and Georg Heigold. Small-footprint keyword spotting using deep neural networks. In *Acoustics, speech and signal processing (icassp), 2014 ieee international conference on*, pages 4087–4091. IEEE, 2014.

32 Tara N Sainath, Brian Kingsbury, George Saon, Hagen Soltau, Abdel-rahman Mohamed, George Dahl, and Bhuvana Ramabhadran. Deep convolutional neural networks for large-scale speech tasks. *Neural Networks*, 64:39–48, 2015.

33 Harsh Sinha and Pawan K Ajmera. Interweaving convolutions: An application to audio classification. *2018 ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) Deep Learning Day*, 2018.

34 Yandre MG Costa, LS Oliveira, Alessandro L Koerich, Fabien Gouyon, and JG Martins. Music genre classification using lbp textural features. *Signal Processing*, 92(11):2723–2737, 2012.

35 Loris Nanni, Yandre Costa, and Sheryl Brahnam. Set of texture descriptors for music genre classification. *22nd International Conference in Central Europeon Computer Graphics, Visualization and Computer Visionin co-operation with EUROGRAPHICS Association*, pages 145–152, 2014.

36 Loris Nanni, Yandre MG Costa, Alessandra Lumini, Moo Young Kim, and Seung Ryul Baek. Combining visual and acoustic features for music genre classification. *Expert Systems with Applications*, 45:108–117, 2016.

37 Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*, 2010.

38 Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. Technical report, IEEE Signal Processing Society, 2011.

39 Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10): 1995, 1995.

40 Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, 2017.

41 Raphael Tang and Jimmy Lin. Deep residual learning for small-footprint keyword spotting. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5484–5488. IEEE, 2018.

42 Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–104. IEEE, 2004.

43 Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

44 Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

45 Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

46 Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In *International Conference on Machine Learning*, pages 584–592, 2014.

47 Matthew D Zeiler, M Ranzato, Rajat Monga, Min Mao, Kun Yang, Quoc Viet Le, Patrick Nguyen, Alan Senior, Vincent Vanhoucke, Jeffrey Dean, et al. On rectified linear units for speech processing. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3517–3521. IEEE, 2013.

48 Jialei Wang, Scott C Clark, Eric Liu, and Peter I Frazier. Parallel bayesian global optimization of expensive functions. *arXiv preprint arXiv:1602.05149*, 2016.

49 Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012. URL `http://arxiv.org/abs/1212.5701`.

50 Pete Warden. "launching the speech commands dataset. *Google Research Blog*, 2017.

51 Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

52 Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044. ACM, 2014.

53 Avrim L Blum and Ronald L Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127, 1992.

54 Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5353–5360, 2015.

55 Douglas Coimbra de Andrade, Sabato Leo, Martin Loesener Da Silva Viana, and Christoph Bernkopf. A neural attention model for speech command recognition. *arXiv preprint arXiv:1808.08929*, 2018.

56 Justin Salamon and Juan Pablo Bello. Unsupervised feature learning for urban sound classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 171–175. IEEE, 2015.

57 Karol J Piczak. Environmental sound classification with convolutional neural networks. In *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, pages 1–6. IEEE, 2015.