IET Signal Processing

Research Article

Audio classification using braided convolutional neural networks

ISSN 1751-9675
Received on 19th August 2019
Revised 30th May 2020
Accepted on 15th June 2020
E-First on 23rd July 2020
doi: 10.1049/iet-spr.2019.0381
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 have 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 and deep neural networks for audio classification. This study aims to achieve audio classification by representing audio as spectrogram images and then use a CNN-based architecture for classification. This study presents an innovative strategy for a CNN-based neural architecture that learns a sparse representation imitating the receptive neurons in the primary auditory cortex in mammals. The feasibility of the proposed CNN-based neural architecture is assessed for audio classification tasks on standard benchmark datasets such as Google Speech Commands datasets (GSCv1 and GSCv2) and the UrbanSound8K dataset (US8K). The proposed CNN architecture, referred to as braided convolutional neural network, achieves 97.15, 95 and 91.9% average recognition accuracy on GSCv1, GSCv2 and US8 K datasets, respectively, outperforming other deep learning architectures.

1 Introduction

Content-based classification refers to the inspection of a data stream for useful information. In context to audio signals, the term is specifically used for recognising 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 (e.g. music and verbal speech) involves the classification of strongly structured and organised audio samples, whereas environmental sound classification involves semi-structured audio samples.

In the 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 a hidden Markov model (HMM) [6], matrix factorisation [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 the advent of GPUs, leading to the 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 leads to the 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 the 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 the generalisation 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 the 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 (DNNs) by iterating over limited data often leads to poor generalisation. Recent publications have addressed the problem of degradation using regularisation [17], weight normalisation [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 has a sparse architecture [20]. The architecture is localised, oriented, sparsely associated, and systematically organised. Imitating the natural sparse architecture of the 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 an optimal sparse architecture modelled using dense convolutional components.

Prior works have bifurcated the task of audio classification into two different domains of speech and environmental 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 CNN (BCNN) outperforms other deep neural architectures without any modification in the architecture with respect to the domain of audio signals.

The rest of the paper is organised 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 is providing a brief summary of the work.

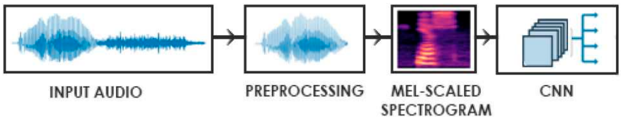

Fig. 1 *Schematic diagram representing CNN-based audio classification*

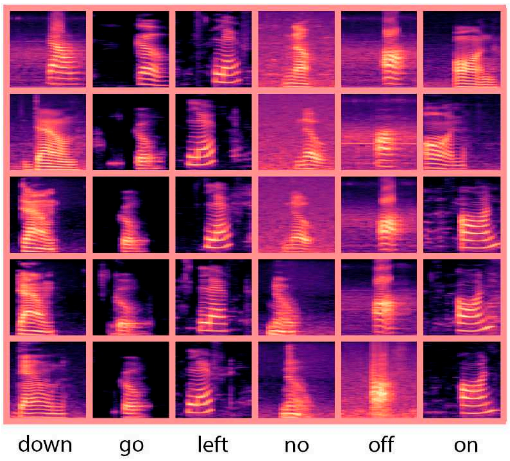

Fig. 2 *Illustration of mel-spectrogram from the audio files in GSCv1 dataset for various speech commands exhibiting intra-class variations and inter-subject similarity*

2 Related work

Identification of speech commands in short-audio segments has a wide range of applications especially with the 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 the development of IoT, researchers are using wireless sensors to analyse 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 the primary obstacle of environmental and demographic variations researchers have used techniques such as HMM [6], matrix factorisation [7], i-vector [28, 29], Hough transform [8], Radon transform [9], restricted Boltzmann machines (RBMs) [29], 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 HMMs [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 the exclusive focus on constraining GMMs, in order to enhance their speed and accuracy. With the advent of high-performance computing systems, DNNs have proven to perform exceptionally better than GMMs in robust modelling 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, the 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]. The audio consists of strong correlations in structure with respect to the frequency domain. The spatial correlations are not utilised by DNNs as they inherently do not model the topology of the input. Moreover, DNNs are not invariant to translational variances in audio signals. Although, an 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 the 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 the context of speech recognition demonstrate that increasing the depth unnecessarily may degrade the performance of the 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 the 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.

3 Proposed methodology

The proposed methodology is represented as three major components: preprocessing, Mel-scaled spectrogram generation and classification, as shown in Fig. 1.

3.1 Preprocessing

The input audio signal is re-sampled to 8 kHz at the pre-processing step. Re-sampling is applied to reduce the 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 the learning algorithm in a biased way.

3.2 Spectrogram generation

The pre-processed raw audio waveform is transformed into a two-dimensional image known as a spectrogram. A spectrogram can be understood as a two-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 a 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]. Fig. 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, $I \in \mathbb{R}^{s \times s \times c}$, where s, c are image size and number of channels, respectively. A kernel $K \in \mathbb{R}^{m \times m \times n}$ is convolved with the initial layer I to generate k feature maps $F \in \mathbb{R}^{(s-m+1) \times (s-m+1) \times k}$. A kernel (or filter) is shared across patches of the 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

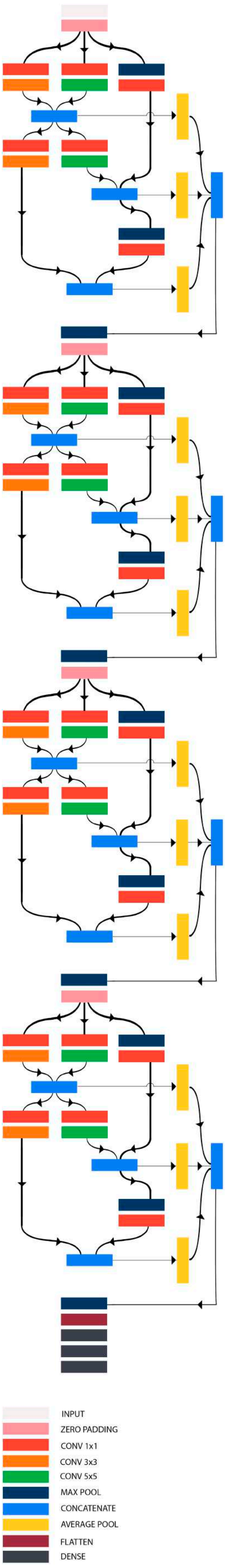

Fig. 3 *Proposed CNN (BCNN) architecture. The different blocks represent different convolutional and max-pooling layers as shown in the colour map (legend)*

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 generalised 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 utilises 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 the ReLU non-linear activation function leads to sparse feature maps naturally [47]. The proposed BCNN approximates optimal sparse structure by utilising available dense computations with uniformity of architecture, non-linear activation function (ReLU) and a large number of filters.

Thus, integrating sparsity in the learning algorithm can fundamentally solve the problem of learning an optimal representation [19]. The primary auditory cortex in mammals has a sparse architecture [20]. Imitating the natural sparse architecture of the 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 the context of speech recognition demonstrate that increasing the depth unnecessarily may further degrade the performance of the 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 BCNN architecture consists of four similar structures (referred to as a bead) connected sequentially, as shown in Fig. 3. The architecture of a single bead is summarised in Table 1. Each bead involves several convolutional and max-pooling layers connected in a braided fashion.

As explained in Section 3.4, the learning algorithm utilises multiple dense connections of standard kernel size 3×3 and 5×5. Each bead makes effective use of kernel size 1×1 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 that 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 maiximise performance [48].

A substantially DNN suffers from performance degradation [14]. Nonetheless, the depth is very crucial for a DNN. 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* 1×1

Table 1 Table summarises the convolutional neural architecture of a bead. The proposed BCNN architecture consists of four 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

convolution 3 × 3, (ii) *convolution* 1 × 1 *convolution* 5 × 5 and (iii) *max-pooling* 3 × 3 *convolution* 1 × 1. The three sets of extracted feature maps are concatenated in different combinations. Braiding feature maps (as shown in Fig. 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 next bead.

Each bead, although it has a similar structure, consists 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 the 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 optimiser [49] is used to learn suitable kernels. Adadelta dynamically adjusts with time and uses first-order information with the least overhead computation loss beyond stochastic gradient descent (SGD). Adadelta is similar to Adagrad as it also aims to adapt the learning rate. Agagrad accumulates all the past squared gradients, which are 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

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

For every new value, the simple moving average is updated, using the last-in-first-out scheme (LIFO). The procedure is shown in the following equation:

$$\overline{g^2_{\mathrm{MA}}} = \overline{g^2}_{\mathrm{MA,prev}} + \frac{g^2_M}{n} - \frac{g^2_{M-n}}{n} \tag{2}$$

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

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

The term γ is analogous to momentum in stochastic gradient descent (SGD). Finally, the parameter θ_t is updated, as shown in the following equation:

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

where η 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 the 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 was 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 the 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, five, six, seven, eight, nine and unknown. The remaining 15 words are categorised into 'unknown' class.

The UrbanSound8K dataset [52] consists of 8732 sound clips up to 4 s in duration. In contrast to GSCv1 and GSCv2 datasets that 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, gunshot, 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 into 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 contain much more data for the 'unknown' class. Utilising class weights while training prevented severe class distribution skews.

4.2 Computational complexity

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

In terms of complexity classes 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 a 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]. Non-linear activation functions such as ReLU, over-specification, and weight-regularisation are used to achieve *improper learning*.

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 s) executes in 1 s, which involves generating spectrograms from an input file, prediction and rendering visualisation.

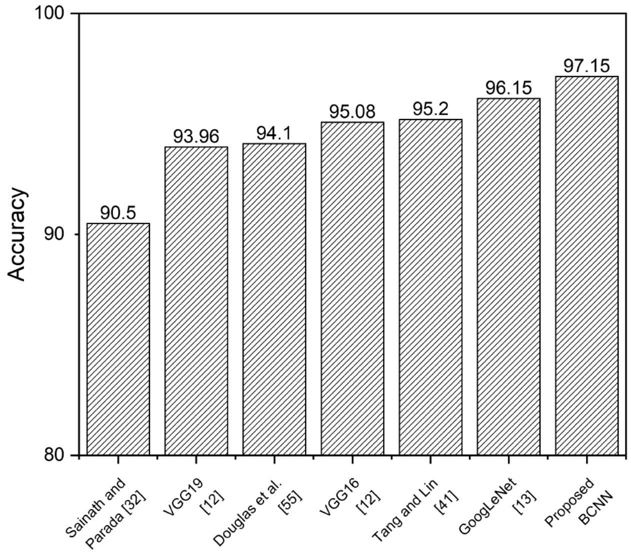

Fig. 4 *Comparative performance of the proposed BCNN on GSCv1 dataset, in terms of average recognition accuracy*

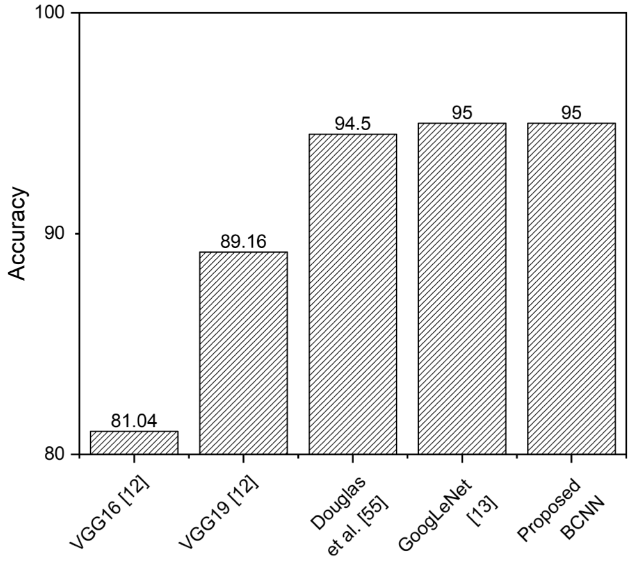

Fig. 5 *Comparative performance of the proposed BCNN on GSCv2 dataset, in terms of average recognition accuracy*

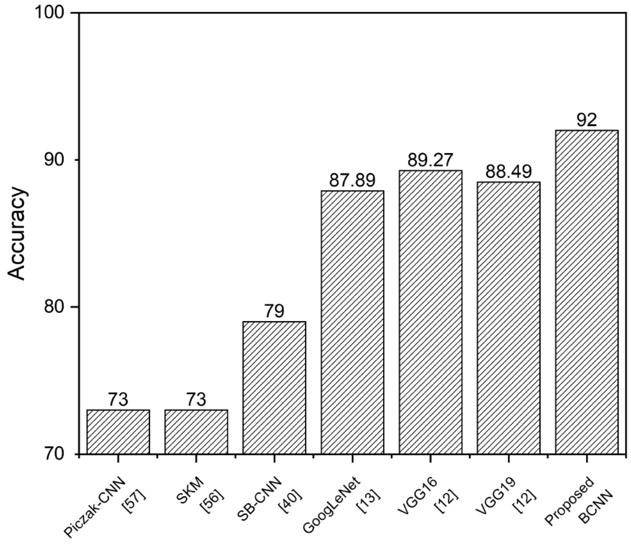

Fig. 6 *Comparative performance of the proposed BCNN on US8K dataset, in terms of average recognition accuracy*

4.3 Speech recognition experiments

With the advent of speech-driven user interfaces, it is important to recognise 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 Figs. 4 and 5, along with accuracy achieved by CNN [1], ResNet [41] and attention-based convolutional recurrent neural network (CRNN) [55]. This 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 normalisation and RMSprop. However, BCNN significantly outperforms other models achieving 97.15 and 95% average recognition accuracy in GSCv1 and GSCv2, respectively. The superior performance of proposed BCNN without using any data augmentation, regularisation or batch normalisation indicates that BCNN is less prone to overfitting.

4.4 Environmental sound classification

Data centres 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 the development of IoT, researchers are using wireless sensors to analyse environmental 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 the US8K dataset are presented in Fig. 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. Piczak-CNN 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 emphasises that BCNN is much more efficient in progressively extracting higher-level representations resulting in a 16.5% relative improvement in average accuracy. In addition, the proposed CNN does not use any explicit data augmentation (used in SB-CNN) or regularisation (used in PiczakCNN). On that account, it can be stated that braided connectivity allows a CNN to learn optimal feature maps from mel-spectrograms realising accurate classification.

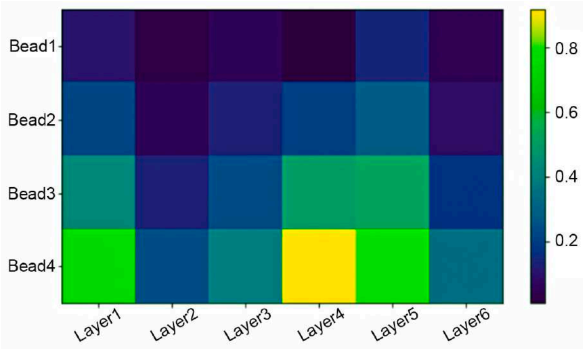

Fig. 7 *Heat map showing average weights of six layers in each bead of a BCNN trained on GSCv2 dataset*

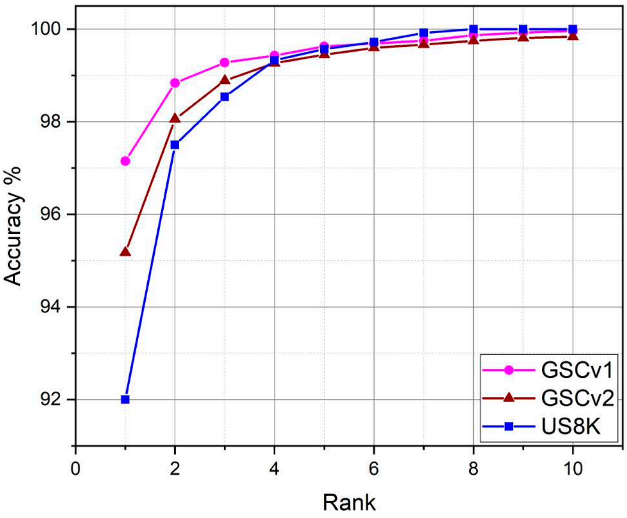

Fig. 8 *Performance of the proposed BCNN on US8K, GSCv1 and GSCv2 datasets expressed in terms of CMC curve*

Although GoogLeNet has much more efficient architecture than VGG16 and VGG19, it tends to overfit the US8 K dataset. This can be attributed to a 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 analyse the use of this incentive by the BCNN trained on the GSCv2 dataset, which estimates the average strength for each of the six convolutional layers. Fig. 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 Fig. 7.

(i) 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.

(ii) 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.

(iii) The final *Bead 4* depicted the last row of Fig. 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 Cumulative match characteristic (CMC) curves

Fig. 8 presents the CMC, used for assessing the closed-set identification performance of a model. Rank-*k* denotes the 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 the audio context for accurate predictions. For example, a smart assistant (trained for KWS task) can query the user to choose the right course of action if the user command is unclear. For a successful query to the 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 US8 K, GSCv1 and GSCv2 datasets, respectively. This suggests that the proposed BCNN is promising in the field of audio classification.

6 Conclusions

This paper evaluated a deep convolutional neural architecture for 2D 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 the 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] Sainath, T.N., Parada, C.: ‘Convolutional neural networks for small-footprint keyword spotting’. Sixteenth Annual Conf. of the Int. Speech Communication Association, Dresden, Germany, 2015

[2] Škraba, A., Stojanović, R., Zupan, A.*, et al.*: ‘Speech-controlled cloud-based wheelchair platform for disabled persons’, *Microprocess. Microsyst.*, 2015, **39**, (8), pp. 819–828

[3] Lefter, I., Rothkrantz, L.J.M., Burghouts, G.J.: ‘A comparative study on automatic audio–visual fusion for aggression detection using meta-information’, *Pattern Recognit. Lett.*, 2013, **34**, (15), pp. 1953–1963

[4] Foggia, P., Petkov, N., Saggese, A.*, et al.*: ‘Reliable detection of audio events in highly noisy environments’, *Pattern Recognit. Lett.*, 2015, **65**, pp. 22–28

[5] Vacher, M., Fleury, A., Portet, F.*, et al.*: ‘*Complete sound and speech recognition system for health smart homes: application to the recognition of activities of daily living*’ (In-Tech, United Kingdom, 2010), pp. 645–673

[6] Robin Rohlicek, J., Russell, W., Roukos, S.*, et al.*: ‘Continuous hidden Markov modeling for speaker-independent word spotting’. 1989 Int. Conf. on Acoustics, Speech, and Signal Processing, 1989. ICASSP-8., 1989, pp. 627–630

[7] Cho, Y.-C., Choi, S.: ‘Nonnegative features of spectro-temporal sounds for classification’, *Pattern Recognit. Lett.*, 2005, **26**, (9), pp. 1327–1336

[8] Dennis, J., Tran, H.D., Chng, E.S.: ‘Overlapping sound event recognition using local spectrogram features and the generalised Hough transform’, *Pattern Recognit. Lett.*, 2013, **34**, (9), pp. 1085–1093

[9] Ajmera, P.K, Jadhav, D.V, Holambe, R.S.: ‘Text-independent speaker identification using radon and discrete cosine transforms based features from speech spectrogram’, *Pattern Recognit.*, 2011, **44**, (10–11), pp. 2749–2759

[10] Deng, J., Dong, W., Socher, R.*, et al.*: ‘Imagenet: a large-scale hierarchical image database’. Computer Vision and Pattern Recognition09, Miami Beach, FL, USA., 2009

[11] Krizhevsky, A., Sutskever, I., Hinton, G.E.: ‘Imagenet classification with deep convolutional neural networks’. Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA., 2012, pp. 1097–1105

[12] Simonyan, K., Zisserman, A.: ‘Very deep convolutional networks for large-scale image recognition’, arXiv preprint arXiv:1409.1556, 2014

[13] Szegedy, C., Liu, W., Jia, Y.*, et al.*: ‘Going deeper with convolutions’. Computer Vision and Pattern Recognition, Boston, MA, USA., 2015

[14] He, K., Zhang, X., Ren, S., *et al.*: ‘Deep residual learning for image recognition’. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, 2016, pp. 770–778

[15] Pandey, G., Dukkipati, A.: ‘To go deep or wide in learning?’, arXiv preprint arXiv:1402.5634, 2014

[16] Bengio, Y.: ‘Learning deep architectures for AI. Foundations and trends’, *Mach. Learn.*, 2009, **2**, (1), pp. 1–127

[17] Srivastava, N., Hinton, G., Krizhevsky, A., *et al.*: ‘Dropout: a simple way to prevent neural networks from overfitting’, *J. Mach. Learn. Res.*, 2014, **15**, (1), pp. 1929–1958

[18] Ioffe, S., Szegedy, C.: ‘Batch normalization: accelerating deep network training by reducing internal covariate shift’. Proc. of the 32nd Int. Conf. on Machine Learning, Lille, France, 2015, Vol. 37, ICML'15, pp. 448–456. JMLR.org, 2015, http://dl.acm.org/citation.cfm?id=3045118.3045167

[19] Grosse, R., Raina, R., Kwong, H., *et al.*: ‘Shift-invariance sparse coding for audio classification’, arXiv preprint arXiv:1206.5241, 2012

[20] Eggermont, J.J.: ‘Chapter 3 - multisensory processing’, In, Eggermont, J.J. (Ed).: ‘*Hearing loss*’ (Academic Press, Netherlands, 2017), pp. 71–90, ISBN 978-0-12-805398-0. doi: https://doi.org/10.1016/B978-0-12-805398-0.00003-7, https://www.sciencedirect.com/science/article/pii/B9780128053980000037

[21] Khunarsa, P.: ‘Single-signal entity approach for sung word recognition with artificial neural network and time–frequency audio features’, *J. Eng.*, 2017, **2017**, (12), pp. 634–645

[22] Kiranyaz, S., Gabbouj, M.: ‘Generic content-based audio indexing and retrieval framework’, *IEE Proc., Vis. Image Signal Process.*, 2006, **153**, (3), pp. 285–297

[23] Nanni, L., Costa, Y.M.G, Lucio, D.R., *et al.*: ‘Combining visual and acoustic features for audio classification tasks’, *Pattern Recognit. Lett.*, 2017, **88**, pp. 49–56

[24] Chen, C.-L., Liao, Y.-F., Tai, C.-L.: ‘Image-to-midi mapping based on dynamic fuzzy color segmentation for visually impaired people’, *Pattern Recognit. Lett.*, 2011, **32**, (4), pp. 549–560

[25] Cristani, M., Bicego, M., Murino, V.: ‘On-line adaptive background modelling for audio surveillance’. Proc. of the 17th Int. Conf. on Pattern Recognition, Cambridge, UK., 2004, Vol. 2, pp. 399–402

[26] Ntalampiras, S.: ‘Hybrid framework for categorising sounds of mysticete whales’, *IET Signal Process.*, 2016, **11**, (4), pp. 349–355

[27] Qian, K., Xu, Z., Xu, H., *et al.*: ‘Automatic detection, segmentation and classification of snore related signals from overnight audio recording’, *IET Signal Process.*, 2015, **9**, (1), pp. 21–29

[28] Nautsch, A., Rathgeb, C., Saeidi, R., *et al.*: ‘Entropy analysis of i-vector feature spaces in duration-sensitive speaker recognition’. 2015 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, April 2015, pp. 4674–4678, doi: 10.1109/ICASSP.2015.7178857

[29] Nautsch, A., Hao, H., Stafylakis, T., *et al.*: ‘Towards pldarbm based speaker recognition in mobile environment: designing stacked/deep plda-rbm systems’. 2016 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, March 2016, pp. 5055–5059, doi: 10.1109/ICASSP.2016.7472640

[30] Hinton, G., Deng, L., Yu, D., *et al.*: ‘Deep neural networks for acoustic modeling in speech recognition’, *IEEE Signal Process. Mag.*, 2012, **29**, pp. 82–97

[31] Chen, G., Parada, C., Heigold, G.: ‘Small-footprint keyword spotting using deep neural networks’. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (icassp), Florence, Italy, 2014, pp. 4087–4091

[32] Sainath, T.N., Kingsbury, B., Saon, G., *et al.*: ‘Deep convolutional neural networks for large-scale speech tasks’, *Neural Netw.*, 2015, **64**, pp. 39–48

[33] Sinha, H., Ajmera, P.K.: ‘Interweaving convolutions: an application to audio classification’. 2018 ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD) Deep Learning Day, London, UK., 2018

[34] Costa, Y.M.G., Oliveira, L.S., Koerich, A.L., *et al.*: ‘Music genre classification using lbp textural features’, *Signal Process.*, 2012, **92**, (11), pp. 2723–2737

[35] Nanni, L., Costa, Y., Brahnam, S.: ‘Set of texture descriptors for music genre classification’. 22nd Int. Conf. in Central European Computer Graphics, Visualization and Computer Vision in Co-operation with EUROGRAPHICS Association, Pilsen, Czech Republic, 2014, pp. 145–152

[36] Nanni, L., Costa, Y.M.G., Lumini, A., *et al.*: ‘Combining visual and acoustic features for music genre classification’, *Expert Syst. Appl.*, 2016, **45**, pp. 108–117

[37] Muda, L., Begam, M., Elamvazuthi, I.: ‘Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques’, arXiv preprint arXiv:1003.4083, 2010

[38] Povey, D., Ghoshal, A., Boulianne, G., *et al.*: ‘The Kaldi speech recognition toolkit’, Technical report, IEEE Signal Process. Soc., 2011

[39] LeCun, Y., Bengio, Y.: ‘Convolutional networks for images, speech, and time series’, *Handbook of Brain Theory Neural Netw.*, 1995, **3361**, (10), p. 1995

[40] Salamon, J., Bello, J.P.: ‘Deep convolutional neural networks and data augmentation for environmental sound classification’, *IEEE Signal Process. Lett.*, 2017, **24**, (3), pp. 279–283

[41] Tang, R., Lin, J.: ‘Deep residual learning for small-footprint keyword spotting’. 2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018, pp. 5484–5488

[42] LeCun, Y., Huang, F.J., Bottou, L.: ‘Learning methods for generic object recognition with invariance to pose and lighting’. Proc. of the 2004 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2004, Vol. 2, pp. I–104

[43] Huang, G., Liu, Z., Der Maaten, L.V., *et al.*: ‘Densely connected convolutional networks’. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708

[44] Szegedy, C., Vanhoucke, V., Ioffe, S., *et al.*: ‘Rethinking the inception architecture for computer vision’. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, NV, USA., 2016, pp. 2818–2826

[45] Lin, M., Chen, Q., Yan, S.: ‘Network in network’, arXiv preprint arXiv:1312.4400, 2013

[46] Arora, S., Bhaskara, A., Ge, R., *et al.*: ‘Provable bounds for learning some deep representations’. Int. Conf. on Machine Learning, 2014, pp. 584–592

[47] Zeiler, M.D., Ranzato, M., Monga, R., *et al.*: ‘On rectified linear units for speech processing’. 2013 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Vancouver, BC Canada, 2013, pp. 3517–3521

[48] Wang, J., Clark, S. C., Liu, E., *et al.*: ‘Parallel Bayesian global optimization of expensive functions’, arXiv preprint arXiv:1602.05149, 2016

[49] Zeiler, M.D.: ‘ADADELTA: an adaptive learning rate method’, CoRR, abs/1212.5701, 2012, http://arxiv.org/abs/1212.5701

[50] Warden, P.: ‘*Launching the speech commands dataset*’ (Google Research Blog, 2017), Accessed Online, Google AI Blog

[51] Warden, P.: ‘Speech commands: a dataset for limited-vocabulary speech recognition’, arXiv preprint arXiv:1804.03209, 2018

[52] Salamon, J., Jacoby, C., Bello, J.P.: ‘A dataset and taxonomy for urban sound research’. Proc. of the 22nd ACM Int. Conf. on Multimedia, Orlando, FL, USA., 2014, pp. 1041–1044

[53] Blum, A.L., Rivest, R.L.: ‘Training a 3-node neural network is np-complete’, *Neural Netw.*, 1992, **5**, (1), pp. 17–127

[54] He, K., Sun, J.: ‘Convolutional neural networks at constrained time cost’. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Boston, MA, USA., 2015, pp. 5353–5360

[55] De Andrade, D.C., Leo, S., Viana, M.L.D.S., *et al.*: ‘A neural attention model for speech command recognition’, arXiv preprint arXiv:1808.08929, 2018

[56] Salamon, J., Bello, J. P.: ‘Unsupervised feature learning for urban sound classification’. 2015 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 2015, pp. 171–175

[57] Piczak, K.J.: ‘Environmental sound classification with convolutional neural networks’. 2015 IEEE 25th Int. Workshop on Machine Learning for Signal Processing (MLSP), Boston, MA, USA, 2015, pp. 1–6