Environment Sound Classification using Multiple Feature Channels and Deep Convolutional Neural Networks

Jivitesh Sharma*, Ole-Christoffer Granmo† and Morten Goodwin‡
Centre for Artificial Intelligence Research
Department of Information and Communication Technology
University of Agder, Norway
*jivitesh.sharma@uia.no, †ole.granmo@uia.no, ‡morten.goodwin@uia.no

***Abstract*—In this paper, we propose a model for the Environment Sound Classification Task (ESC) that consists of multiple feature channels given as input to a Deep Convolutional Neural Network (CNN). The novelty of the paper lies in using multiple feature channels consisting of Mel-Frequency Cepstral Coefficients (MFCC), Gammatone Frequency Cepstral Coefficients (GFCC), the Constant Q-transform (CQT) and Chromagram. Such multiple features have never been used before for signal or audio processing. Also, we employ a deeper CNN (DCNN) compared to previous models, consisting of 2D separable convolutions working on time and feature domain separately. The model also consists of max pooling layers that downsample time and feature domain separately. We use some data augmentation techniques to further boost performance. Our model is able to achieve state-of-the-art performance on all three benchmark environment sound classification datasets, i.e. the UrbanSound8K (98.60%), ESC-10 (97.25%) and ESC-50 (95.50%). To the best of our knowledge, this is the first time that a single environment sound classification model is able to achieve state-of-the-art results on all three datasets and by a considerable margin over the previous models. For ESC-10 and ESC-50 datasets, the accuracy achieved by the proposed model is beyond human accuracy of 95.7% and 81.3% respectively.**

***Index Terms*—Deep Convolutional Neural Networks, Multiple Feature Channels, Environment Sound Classification, Mel-Frequency Cepstral Coefficients, Constant Q-transform, Gammatone Frequency Cepstral Coefficients, Chromagram, ESC-50, ESC-10, UrbanSound8K**

I. Introduction

THERE are many important applications related to speech and audio processing. One of the most important application is the Environment Sound Classification (ESC) that deals with distinguishing between sounds from the real environment. It is a complex task that involves classifying a sound event into an appropriate class such as siren, dog barking, airplane, people talking etc. This task is quite different compared to Automatic Speech Recognition (ASR) [1], since environment sound features differ drastically from speech sounds. In ASR, speech is converted to text. However, in ESC, there is no such thing as speech, just sounds. So, ESC models are quite different compared to ASR models.

ASR models typically consist of hybrid Deep Neural Network-Hidden Markov Model (DNN-HMM) [2] or more recently, end-to-end Recurrent Neural Networks (RNNs) encoder-decoder structure [3], sometimes with attention mechanism [4], [5] and coupled with Convolutional Neural Networks (CNNs) for feature extraction [6], frequently with language models [7]. On the other hand, there are enormous number of possibilities to build ESC models using different audio feature extraction techniques and AI or non-AI based classification models. The most successful ESC models consist of one or more standard audio feature extraction techniques and deep neural networks. The most widely used audio feature extraction technique is the Mel Frequency Cepstral Coefficient (MFCC) [8]. However, in this paper, we explore other feature extraction methods like the Constant Q-Transform (CQT) [9], Chromagram [10], Gammatone Frequency Cepstral Coefficients (GFCC) [11] and use a stack of multiple features as input to our classifier.

After feature extraction, the next stage is classification. Many machine learning algorithms have been used to classify sound, music or audio events such as the Decision Tree classifier [12], Random Forest [13], Support Vector Machine [14]–[16], Extreme Learning Machine [17], [18] etc. However, in the ESC task, Deep CNNs have been able to outperform all other classification techniques. They have been used by many researchers to achieve high classification performance [19]–[27]. In this paper, we also employ a Deep CNN for classification. However, we split between time and frequency domain feature processing by using separable convolutions [28] with different kernel sizes. Also, we use max pooling across only one of the domains at a time, until after the last set of convolutional layers to combine time and frequency domain features. This enables processing time and frequency domain features separately and then performing combining them at a later stage.

Using these techniques allows our model to achieve state-of-the-art performance on all three benchmark datasets for environment sound classification task, namely, ESC-10, ESC-50 [29] and UrbanSound8K [30]. The rest of the paper has been organized in the following manner: Section 2 briefly enlists previous research done on the ESC task using Deep CNNs and other previous state-of-the-art methods. Section 3 details our proposed ESC model consisting of multiple feature channels and a Deep CNN classifier. Section 4 explains our experimental setup and implementation details and also displays our final results on the datasets. Finally, section 5 concludes our work.

II. Related Work

There have been several innovative and high performance approaches proposed for the task of environmental sound classification (ESC). Here, we focus on deep learning based and state-of-the-art methods on the ESC-10, ESC-50 and UrbanSound8K benchmark datasets.

There have been many different deep learning and neural network based techniques used for the ESC task. One of the most popular methods that are at the core of the highest performing models for not only ESC but also for ASR are the Convolutional Neural Networks (CNNs) [31]. In [24], a deep CNN was shown to give competitive results for the ESC tasks by thorough and exhaustive experimentation on the three benchmark datasets.

In [32], phase encoded filterbank energies (PEFBEs) was proposed as a novel feature extraction technique. It was shown that it outperformed vanilla filterbank energies (FBEs). Finally, a score-level fusion of FBEs and PEFBEs with a CNN classifier achieved best performance. So, it was shown experimentally that PEFBEs posses complementary features to FBEs.

Another novel aggregation of feature extraction techniques was proposed in [33]. It was shown that aggregating multiple features with complementary behaviour along with a CNN outperformed models that consisted of Gaussian Mixture Model (GMM) classifier. The Teager Energy Operator (TEO) was used to modify the Gammatone cepstral coefficients to produce TEO-GTSC. TEO is able to track the energy changes in an audio signal very accurately. The best results were produced when Gammatone cepstral coefficients were combined TEO-GTSC with score-level fusion.

A multi-temporal resolution CNN was proposed in [34]. Here, multiple CNNs with different filter sizes and stride lengths work on a raw audio signal on different temporal resolutions, in parallel. This hierarchy of features is combined by direct connections between convolutional layers which has better information flow (somewhat similar to DenseNets [35]).

An end-to-end approach based on feature extraction and classification of raw audio signals by CNNs was proposed in [19]. The model, called EnvNet, was able to achieve competitive performance on all three benchmark datasets. In the second version of the EnvNet, called EnvNetv2 [23], the authors employed a mechanism called Between Class (BC) learning. In BC learning, two audio signals from different classes are mixed with each other with a random ratio. The CNN model is then fed the mixed sound as input and trained to output this mixing ratio. BC learning was also shown to boost the performance of other ESC models as well.

Another approach based on 1D-CNNs working on raw audio signals is proposed in [22]. A 1D-CNN is used to classify environmental sounds on variable length raw audio waveforms. They show that no feature extraction is needed since the first layer of the 1D-CNN can be initialized as a Gammatone filter bank. Initializing the convolution kernels of the first layer by 64 band pass gammatone filters, the raw input signal is decomposed into 64 frequency bands. This end-to-end approach achieves 89% accuracy.

In [26], a pure convolutional approach to ESC was proposed. The model proposed in [26] consisted of a very deep fully convolutional neural network with a maximum of 34 layers. The network was carefully designed with batch normalization layers and residual learning. Their 18 layer model gave the best performance which matched the performance of models that used log-mel features.

A very innovative and effective unsupervised approach of learning a filterbank from raw audio signals was proposed in [36]. Convolutional Restricted Boltzmann Machine (ConvRBM), which is an unsupervised generative model, was trained to raw audio waveforms. The authors show that the sub-band filters in the mid-frequency range resemble fourier basis while in the low-frequency range resemble gammatone basis. A CNN is used as a classifier along with ConvRBM filterbank and score-level fusion with Mel filterbank energies. Their model achieves 86.5% on the ESC-50 dataset which was the state-of-the-art which we beat in this paper.

Another innovative approach of using visual knowledge transfer learning for sound recognition was proposed in [37]. The model, called SoundNet, leveraged the large collection of unlabelled videos (with audio) to transfer discriminative visual information to boost environment sound classification. The visual information of the unlabelled videos was given as input to visual recognition networks and the raw audio waveforms from those videos were fed to the SoundNet. The model was trained to minimize the Kullback-Leibler divergence between the outputs of the SoundNet and the visual recognition network to transfer the visual information to the SoundNet. Finally, ignoring the output layer of the SoundNet, the model is used as feature extraction to train an SVM classifier.

In [21], the scarcity of data for training a Deep CNN was addressed by data augmentation techniques for audio signals. Data augmentation methods such as time stretching, pitch shifting, dynamic range compression and adding noise were used and thoroughly analysed with experiments on the UrbanSound8K dataset. It was also shown that results can be further improved by class-conditional data augmentation. We use some of the augmentation techniques proposed in [21].

A novel data augmentation technique for audio was proposed in [20]. The method called Mixup is used to generate new training data for the CNN model. It consists of mixing two audio signals and their labels, in a linear interpolation manner, where the mixing is controlled by a factor λ. Time Stretch and Pitch Shift are also used for augmentation. Log-mel spectrograms and gammatone spectrograms are used as audio features as input to the Deep CNN model. In this way, their model achieves 83.7% accuracy on the UrbanSound8K dataset and competitive performance on the ESC-10 and ESC-50 datasets.

Some well known State-of-the-art Deep CNNs such as AlexNet [38] and GoogleNet [39] were used for ESC in [40]. Features such as MFCC, Spectrogram and CRP of audio signals were extracted and treated as image representations which were then fed to the Deep CNNs. Both AlexNet and GoogleNet were able to obtain decent classification accuracies on benchmark ESC datasets.

A complex two stream structure deep CNN model was pro-

posed in [27]. It consists two CNN streams which are combined with decision-level fusion at the end. One is the LMCNet which works on the log-mel spectrogram, chroma, spectral contrast and tonnetz features of audio signals and the other is the MCNet which takes MFCC, chroma, spectral contrast and tonnetz features as inputs. The decisions of the two CNNs are fused using the Dempster-Shafer evidence theory to get the final TSDCNN-DS model. It achieves 97.2% accuracy on the UrbanSound8K dataset, which was the state-of-the-art on that dataset, which we beat in this paper.

These exemplary research works mentioned above provide us with many insights by achieving high performance on difficult datasets. But, they also suffer from issues regarding feature extraction, computational complexity and CNN model architecture. In this paper, we try to address these issues and in doing so, achieve state-of-the-art performance. In the next section, we explain our model in detail.

III. Proposed Environment Sound Classification Model

We propose a novel ESC model that consists of multiple feature channels extracted from the audio signal and a new DCNN architecture consisting of separable convolutions, that works on time and frequency domain separately.

Usually, one or two feature extraction techniques along with some statistical information is used as the feature set. However, just a couple of feature extraction methods aren't able to obtain a majority of distinguishable features for all categories of environment sounds.

We address that issue by employing multiple feature extraction techniques and stacking their outputs like channels in an image to make them suitable for DCNN. The feature extraction stage consists of five channels of features, which are: Mel-Frequency Cepstral Coefficients, Gammatone Frequency Cepstral Coefficients, Constant Q-transform and Chromagram.

For the classification stage, we propose a CNN architecture that works better for audio data. We use separable convolutions to process time and frequency domain features separately and aggregate them at the end. Also, the downsampling value is different for time and frequency domains in the maxpooling layers. In the subsequent sub-sections, we explain the feature extraction and classification stages of our model.

A. Multiple Feature Channels

Some papers have advocated the use of aggregation of more than one set of features using different signal feature extraction methods to achieve higher performance [20], [24], [27], [41]–[43] in both ASR and ESC tasks. In this paper, instead of just using one or two feature extractors and feeding a one or two channel input to the CNN classifier, we employ four major audio feature extraction techniques to create a four channel input for the Deep CNN. Incorporating different features with different scales provides the CNN with more distinguishable characteristics and complementary feature representations to accurately classify audio signals. We process audio signals to extract features using the following methods:

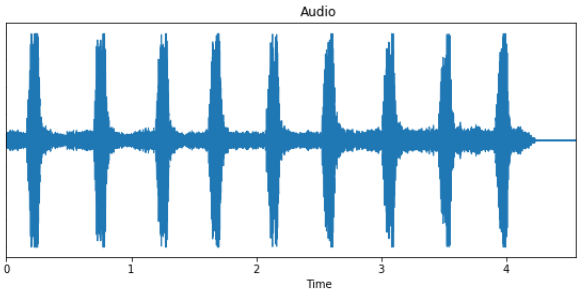
1) MFCC: The Mel-Frequency Cepstral Coefficients (MFCC) has been a standard feature extraction technique and has been successfully used to benchmark ASR as well as ESC models [8]. The development of MFCC was propelled by human auditory perception. MFCCs produce a compact representation of an audio signal. It differs from other cepstral features in the frequency bands which are on the mel-scale. The detailed five step procedure to extract MFCCs can be found in [44]. We use 128 bands in the Mel-filter bank to produce 128-dimensional features using standard hamming window size of 1024 and hop length of 512. Since MFCC is susceptible to noise, we normalize it between 0 and 1.

2) GFCC: The Gammatone Frequency Cepstral Coefficients (GFCC) has also been a popular choice of feature extraction in ESC and ASR tasks [11]. The gammatone filter is a linear filter that is outlined by an impulse response which is a product of a gamma distribution and sinusoidal tone. Hence, the name gammatone. It is especially advantageous to use GFCC with MFCC as they complement each other, due to the capability of GFCC being able to proficiently characterize transient sounds classes such as footsteps and gun-shots [41]. Detailed analysis of the benefits of combining MFCC and GFCC can be found in [42]. We use 128 bands Gammatone filters with standard hamming window size of 1024 and hop length of 512 to produce 128-dimensional features.

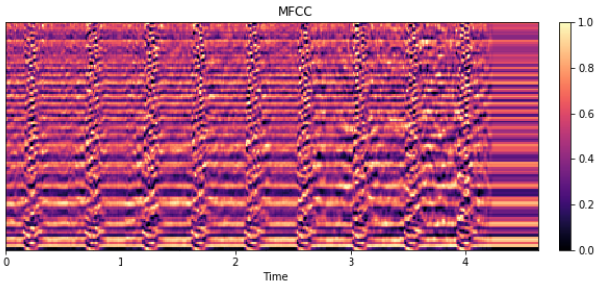
3) CQT: The Constant Q-transform is a time-frequency analysis technique that is particularly suitable for music audio signals [9], [45], [46]. It is essentially a Gabor wavelet transform, so unlike STFT, it has higher frequency resolution for lower frequencies and higher time resolution for higher frequencies. Due to this, it was shown in [47] that CQT outperformed standard MFCC feature extraction for ESC using CNNs. The results shown in [48], illustrated CQT's ability to capture low-to-mid level frequencies better than MFCC for audio scene classification, which is essentially the same task as ESC. We set the number of bands per octave to 128 and window size of 1024 with a hop length of 512, so we get feature vectors of equal dimensionality as MFCC and GFCC.

4) Chromagram: Another feature extraction technique that is popular with music audio signals is the Chromagram [10]. Chroma based features are especially useful for pitch analysis of audio signals. They can be used to distinguish among audio signals by assigning them pitch class profiles. This makes chromagrams particularly proficient in audio structure analysis [49]. We use STFT (Short-time Fourier Transform) to compute chroma features. The number of chroma features extracted from the audio waveform is set to 128 with window size of 1024 and hop length of 512.

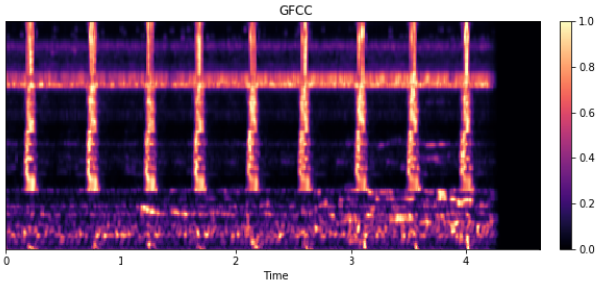
The MFCC, GFCC, CQT and Chroma features, 128 dimensional each, are stacked together to create a four channel input for the Deep CNN. Each feature plays it's part in the classification task. MFCC acts as the backbone by providing rich features, GFCC adds transient sound features, CQT contributes



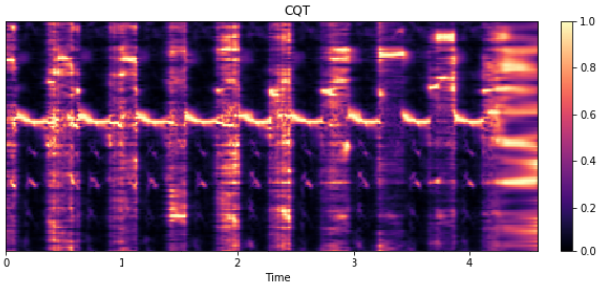
(a) Audio Signal



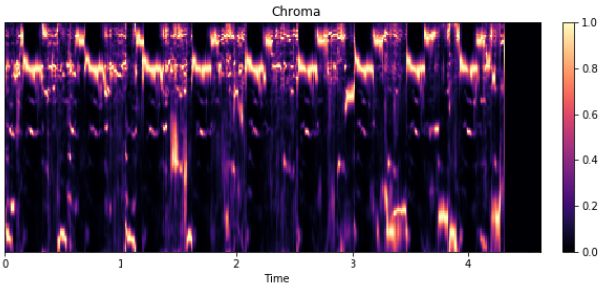
(b) Mel Frequency Cepstral Coefficients



(c) Gammatone Frequency Cepstral Coefficients



(d) Constant Q-Transform



(e) Chromagram

Fig. 1: Multiple Feature Channels

with better low-to-mid frequency range features and finally Chromagram provides pitch category analysis and signal structure information. Figure 1 shows a graphical representation of the features extracted from an audio signal (Figure 1(a)). All features are normalized between 0 and 1 using min-max normalization. From the figure, we can see the contrast in the values of each feature.

For example, the spike on the 4th second is represented differently in each feature. Some features, like the MFCC and GFCC, represent that spike with high values (as evident from the colour bar next to the graphs), whereas CQT and Chromagram represent it with low values (dark regions). The representation of MFCC is completely different as it provides some positive value in every region, with enough discrimination capabilities. So, it acts as the backbone of the multi-channel input. On the other hand, the other features act as complementary features that eke out some additional distinguishable features.

There might be more feature channels that can be added to further increase the discrimination strength of the input, but that can also increase the pre-processing overhead, number of kernels required to process the multi-channel input and computational complexity of the whole model. Hence, we restrict the number of channels to 4.

B. Deep Convolutional Neural Network

We use the Convolutional Neural Network [31] as the audio signal classifier for the ESC task. The architecture of our Deep CNN for environmental sound classification proposed in this paper is shown in Table 1. It consists of eight repetitions of *Conv2D-Conv2D-MaxPool-BatchNorm* with different number of kernels and kernel sizes. From Table 1, we can see that almost all convolutional layers are made up of depth-wise separable convolutions [28]. However, unlike depth-wise separable convolutions where an $1 \times m$ kernel is followed by an $m \times 1$ kernel, we use two consecutive $1 \times m$ kernels followed by two $n \times 1$ kernels, where $n \neq m$.

This is because we separate using the convolution operation on the time and frequency domain. The output $\mathcal{O}_i$ of a convolution operation is given by:

$$\mathcal{O}_i = \phi(W \otimes X_i + b) \tag{1}$$

where, W is the kernel, b is the bias, ϕ is the activation function and X_i is the input. In the case of the ESC task, the input are the features extracted from the audio signals. Each feature set is of the shape (t, f, c), where t is the compressed time domain (compressed due to window size and hop length) and c is the number of channels. Each window of time yields f number of features ($f = 128$ in our model). So, we treat the time domain and the feature domain separately. The kernels with the form $1 \times m$ work on the feature domain and the ones with $n \times 1$ work on the time domain.

Using the $1 \times m$ type of convolution operation enables the network to process each set of features from a time window separately. And, the $n \times 1$ type of convolution allows the aggregation of a feature along the time domain. Now, c corresponds to the number of feature extraction methods

TABLE I: The Deep CNN Architecture

Layer Type	Kernel Size	Pooling Size	No. of Kernels/ Neurons
Conv2D	1x3	-	32
Conv2D	1x3	-	32
BatchNorm	-	-	-
MaxPool2D	-	1x2	-
Conv2D	7x1	-	32
Conv2D	7x1	-	32
BatchNorm	-	-	-
MaxPool2D	-	4x1	-
Conv2D	1x3	-	64
Conv2D	1x3	-	64
BatchNorm	-	-	-
MaxPool2D	-	1x2	-
Conv2D	7x1	-	64
Conv2D	7x1	-	64
BatchNorm	-	-	-
MaxPool2D	-	4x1	-
Conv2D	1x3	-	128
Conv2D	1x3	-	128
BatchNorm	-	-	-
MaxPool2D	-	1x2	-
Conv2D	7x1	-	128
Conv2D	7x1	-	128
BatchNorm	-	-	-
MaxPool2D	-	4x1	-
Conv2D	7x3	-	256
Conv2D	7x3	-	256
BatchNorm	-	-	-
MaxPool2D	-	4x2	-
Conv2D	7x3	-	512
Conv2D	7x3	-	512
BatchNorm	-	-	-
MaxPool2D	-	4x2	-
Flatten	-	-	-
Dense	-	-	1024
Dense	-	-	1024
Dense	-	-	No. of Classes

we adopt (in our model, $c = 4$). So, each kernel works on each channel, which means that all different types of features extracted from the signal feature extraction techniques is aggregated by every kernel. Each kernel can extract different information from an aggregated combination of different feature sets.

Another major advantage of using these type of convolutions is the reduction in number of parameters. This was the primary advantage of separable convolutions when they were proposed in [28]. For a kernel of size $1 \times m$, one dimension of the kernel is 1, it has the same number of parameters as a 1D convolution of kernel size m. But, it has the operational advantage of 2D convolution, as it works on two spatial dimensions.

Also, this type of convolution operates in accordance with the way data is represented. In case of standard square kernels like $n \times n$, which are used for computer vision tasks, the dimensions of the kernel are in accordance to the image's spatial structure. The 2D structure of an image represents pixels, i.e. both dimensions of an image represent the same homogeneous information. Whereas, in case of audio features, one dimension gives a compact representation of frequency features of a time window and the other dimension represents the flow of time (or sliding time window). So, in order to process information accordingly and respect the information from different dimensions of the input, we use $1 \times m$ and $n \times 1$ separable convolutions.

Figure 2 shows a small example of the difference between

Fig. 2: Separable Convolutions working in the time and feature domains vs Standard Convolutions

separable convolutions and standard square convolutions. The x-axis is the time domain (time windows) and the y-axis contains the features extracted in a time window. The red rectangle represents a $1 \times m$ kernel working in the feature domain, the blue rectangle represents a $n \times 1$ kernel working in the time domain and the yellow rectangle represents a $n \times n$ square kernel. The green arrows give the direction of the kernel's movement. The separable convolutions work on either the time or the feature domain. For example, the red kernel processes all features in time window t^j and the blue kernel processes the flow of feature f^i for all consecutive time windows. Since both the dimensions convey different information, the data usage efficiency if greater. However, in case of the yellow kernel, if it's a 2×2 kernel, it processes two features f^{i-1} and f^i in two time windows t^{j-1} and t^j. In this case, there is no consistency in gain of information. It reduces data usage efficiency and increases redundancy since both dimensions have different types of information. The square convolutions work best for images because they have the same information (pixel information) in both dimensions. The pooling operations are also separate across the time and feature domain, in the same manner as the convolution operations. Note that, the time domain and feature domain kernel sizes are different. This is because $t > f$. These factors make the idea of combining multiple feature extraction methods to create a multi channel input more appealing for a convolutional neural network.

The final batches of convolution and pooling operations combine the time and feature domain by using $n \times m$ kernels. These layers learn additional information across both the time and feature domains and assemble the information for the fully connected layers at the end to get the final solution.

We also add batch normalization layers [50] after every couple of convolutional layers. These layers normalize the

input to the next layers in order to reduce internal covariate shift which refers to the phenomenon that occurs during training when the input distribution of each layer changes due to the changes in the parameters of the previous layers. This requires lowering learning rates which slows down training. Hence, batch normalization is now an indispensable part of a CNN architecture.

The most common choice of activation function is the Rectified Linear Unit (ReLU) [51], in equation 2. It has several advantages over other activation functions such as faster computation, it does not saturate unlike sigmoid and hyperbolic tangent functions, it has sparse activation and is biologically inspired. However, we use the Leaky ReLU [52] activation function, in equation 3, after every convolution and fully connected layer (except the last/output layer). This is because due to random initialization of weights, some may fall below zero and result in a constant gradient feedback of zero. Leaky ReLU alleviates this problem by allowing a small positive gradient to pass even if the weight is below zero.

$$f(x) = \max(0, x) \tag{2}$$

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.01x, & \text{otherwise} \end{cases} \tag{3}$$

The softmax function was used at the final layer to obtain class probabilities, as in equation 4. We use the categorical cross-entropy loss, as shown in equation 5. The softmax function and cross-entropy loss are used together because they provide a smooth and simple gradient which makes computations much easier. The gradient is calculated as shown in equation 6, where, y_i is the output probability of sample x_i.

$$\mathcal{S}(x_i) = \frac{e^{x_i}}{\sum_{j=0}^{C} e^{x_j}} \tag{4}$$

$$\mathcal{L} = -\sum_{i=0}^{C} t_i log(\mathcal{S}(x_i)) \tag{5}$$

$$\frac{\partial \mathcal{L}}{\partial x_i} = y_i - t_i \tag{6}$$

The DCNN was trained using the Adam optimizer [53] with Nestorov momentum [54] with a very small step size $\eta = 0.0001$ and default parameters to minimize the loss function. Momentum accelerates the speed of convergence. It acts as a memory for the network to keep the updates in the direction with the maximum improvement by taking account of previous gradients, as shown in equation 7. However, if there are oscillations in the gradient updates, the momentum can lead the loss in the wrong direction. The Nestorov momentum acts as a correction to vanilla momentum. It uses the moving average of gradients to calculate an interim set of parameters to "look ahead" and calculate the gradient from these interim parameters, as shown in equation 8. If this gradient leads to an increase in loss, then the gradients will direct the update towards the previous set of parameter [54].

$$\begin{aligned} v_{t+1} &= \alpha v_t - \eta \nabla \mathcal{L}(w_t) \\ w_{t+1} &= w_t + v_{t+1} \end{aligned} \tag{7}$$

$$\begin{aligned} w_{interim} &= w_t + \alpha v_t \\ v_{t+1} &= \alpha v_t - \eta \nabla \mathcal{L}(w_{interim}) \\ w_{t+1} &= w_t + v_{t+1} \end{aligned} \tag{8}$$

where, α is the momentum parameter and v_t is the momentum of gradients at time t. In order to avoid overfitting, we use Dropout [55] with ratio 0.5 after each dense layer. To further improve the generalization performance of our model, L_2 regularization was used on the weights of the dense layers with regularization parameter $\lambda = 0.1$. L_2 regularization penalizes the magnitude of the weights and reduces redundancy. It's derivative is easily computable which makes it suitable for gradient based learning algorithms.

The results in the next section show that, all these aspects of our model and each feature in the multi channel input contributes towards improving the performance of the system. We exhaustively test our model on the three benchmark ESC datasets: ESC-10, ESC-50 and UrbanSound8K. We achieve state-of-the-art results on all three datasets.

IV. Experiments and Results

We report state-of-the-art results on ESC benchmark datasets, i.e. UrbanSound8K, ESC-10 and ESC-50, using the proposed model. We train and test our model on each of the three benchmark datasets according to the specified folds of training by using k-fold cross validation [56] and averaging the classification accuracy across the folds. For ESC-10 and ESC-50, we use $k = 5$ and for UrbanSound8K, we use $k = 10$.

We use Tensorflow [57] and Keras [58] to implement our CNN classifier and Librosa [59], Essentia [60] and the Matlab Signal Processing Toolbox [61] for audio processing and feature extraction. In terms of hardware, we use the NVIDIA DGX-2 consisting of 16 NVIDIA Tesla V100 GPUs with 32 Gigabytes of VRAM each and a system memory of 1.5 Terabytes.

As shown in [20], [21], data augmentation plays a very important role in improving performance, especially when the model is large and data is scarce. Here, we use time stretch, pitch shift and add random gaussian noise as in [21]. These data augmentation techniques improve the accuracy of the model even further.

Table 2 displays the results of previous state-of-the-art ESC models that tested their methods on one or more of the three benchmark datasets. All of these models have been briefly described in Section 2. The last row of the table shows the results of our proposed model on the three datasets. For the UrbanSound8K dataset, the previous state-of-the-art accuracy was 97.2%, which we beat by 1.4%. However, the previous state-of-the-art accuracies on the ESC-10 and ESC-50 datasets were 92.2% and 86.5% respectively. Whereas, our proposed model gains considerably higher accuracies of 98.25% and 95.48% on the ESC-10 and ESC-50 datasets respectively. This could be partly because some state-of-the-art models on the UrbanSound8K dataset, such as [22], [27], weren't tested on the ESC-10 and ESC-50 datasets. But, it's mostly because ESC-10/ESC-50 datasets have different dynamics compared to the UrbanSound8K dataset. This is evidenced from the fact that almost all models shown in Table 2 get varying accuracies for ESC-10/ESC-50 and UrbanSound8K datasets.

Since ESC-10 and ESC-50 come from exactly the same distribution, the difference in the reported accuracies on them are quite predictable. However, the UrbanSound8K comes from a different distribution and more importantly it is an imbalanced dataset, unlike ESC-10/ESC-50. Also, the UrbanSound8K dataset has varying sampling rate and audio length. We are pointing out these differences to show that despite these dissimilarities, our model is able to achieve state-of-the-art performance on all three datasets, which, to the best of our knowledge, hasn't been done before.

From Table 2, we can also see that our proposed model is able to surpass human performance on the ESC-10 and ESC-50 datasets. Now, we show experimental analysis and results of our model on the ESC-10/ESC-50 and UrbanSound8K datasets separately. We show the performance of our model on the datasets in terms of moving average accuracy across the folds. Also, we show the increase in performance after adding more feature channels and the benefit of using data augmentation. We also test different architectures of our model. We compare our proposed separable convolution architecture with standard square kernel convolution architecture. We name our architecture in Table 1 as DCNN-8, where 8 is the number of sequences of *Conv2D-Conv2D-MaxPool-BatchNorm* layers. We keep the two fully connected layers and one output softmax layer for all architectures.

TABLE II: Previous state-of-the-art ESC models vs Proposed model

Model	ESC-10	ESC-50	UrbanSound8K
Human [29]	95.70	81.30	-
EnvNet [19]	86.80	66.40	66.30
EnvNet+logmel-CNN [19]	88.10	74.10	71.10
EnvNetv2 [23]	88.80	81.60	76.60
EnvNetv2+strong augment [23]	91.30	84.70	78.30
M18 [26]	-	-	71.68
SoundNet [37]	92.20	74.20	-
PiczakCNN [24]	90.20	64.50	73.70
Multilevel Features+Multi-temporal resolution CNN [34]	-	75.10	-
AlexNet [40]	86.00	65.00	92.00
GoogleNet [40]	86.00	73.00	93.00
SB-CNN [21]	-	-	79.00
CNN+Augment+Mixup [20]	91.70	83.90	83.70
GTSC⊕TEO-GTSC [33]	-	81.95	88.02
PEFBEs [32]	-	73.25	-
FBEs⊕PEFBEs [32]	-	84.15	-
ConvRBM-BANK [36]	-	78.45	-
FBEs⊕ConvRBM-BANK [36]	-	86.50	-
1D-CNN Random [22]	-	-	87.00
1D-CNN Gamma [22]	-	-	89.00
LMCNet [27]	-	-	95.20
MCNet [27]	-	-	95.30
TSCNN-DS [27]	-	-	97.20
Multiple Feature Channel + Deep CNN (Proposed)	**97.25**	**95.50**	**98.60**

A. ESC-10/ESC-50

The ESC-50 dataset is one of the most widely used environmental sound classification benchmark datasets [29]. It consists of 2000 audio files of 5 seconds length each, sampled at 16kHz and 44.1kHz. We use the set of audio files sampled at 44.1kHz. The recordings in the ESC-50 dataset are categorized into 50 balanced and disjoint classes. The sounds can be divided into 5 major groups: animals, natural soundscapes and water sounds, human non-speech sounds, interior/domestic sounds, and exterior/urban sounds. The dataset has been pre-arranged into 5 folds for unbiased comparable results. We use these pre-determined folds with 5-fold cross validation and report the average accuracy of our model across the 5 predefined folds, as mentioned in [29].

The ESC-10 is a subset of the ESC-50 dataset that consists of 10 balanced and disjoint classes (dog bark, rain, sea waves, baby cry, clock tick, person sneeze, helicopter, chainsaw, rooster, fire crackling) of 400 audio files. It uses the same implementation of pre-arranged 5 folds, which we follow for testing our model. We train our model for 50 epochs per fold and test the model on the remaining fold. We calculate the average of this test accuracy for all folds.

In Table 3, we show the effect of features on the model's performance on the ESC-50 dataset. The accuracy of the model increases with every additional feature. Adding more features might have increased performance even further, but it would have increased the computational cost as well. Table 4 shows the effect of data augmentation on the performance of the model. As mentioned above, we use time stretch, pitch shift and add random gaussian noise to the audio signals as in [21]. Augmentation plays an important role in achieving state-of-the-art performance as shown in table 4.

Figure 3 shows the convergence of our model in terms of moving average accuracy per fold for the ESC-50 dataset. As the model learns, the testing accuracy increases per fold. This, in turn, increases the average accuracy per fold, which is how the performance of a model is calculated on the ESC-50 dataset.

In figure 4, we show the results of testing different model sizes on the ESC-50 dataset. We test for 4,6,8 and 10 sequences of *Conv2D-Conv2D-MaxPool-BatchNorm* layers. The best performing model is DCNN-8. The model's accuracy increases with the increase in the number of layers but starts to decrease

TABLE III: Performance on the ESC-50 dataset

Model	MFCC	GFCC	CQT	Chroma	Accuracy
DCNN-8	✓				83.025
DCNN-8	✓	✓			89.750
DCNN-8	✓	✓	✓		93.125
DCNN-8	✓	✓	✓	✓	95.500

TABLE IV: Effect of Data Augmentation for ESC-50 dataset

Model	Accuracy
Multi-channel input + DCNN-8	89.250
Multi-channel input + DCNN-8 with strong augmentation	95.500

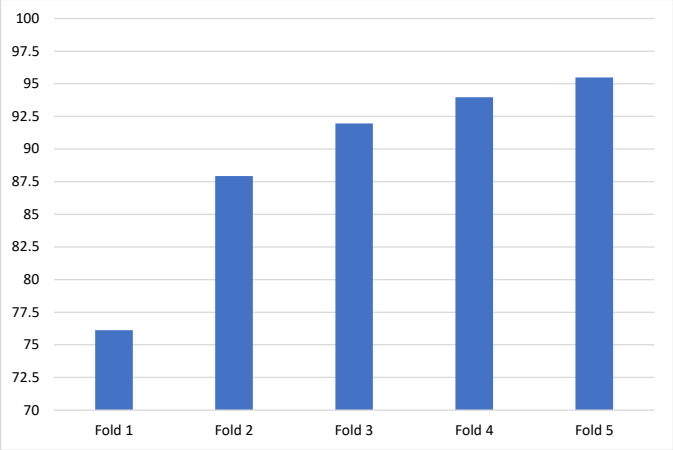


Fig. 3: Moving Average Accuracy per Fold on ESC-50

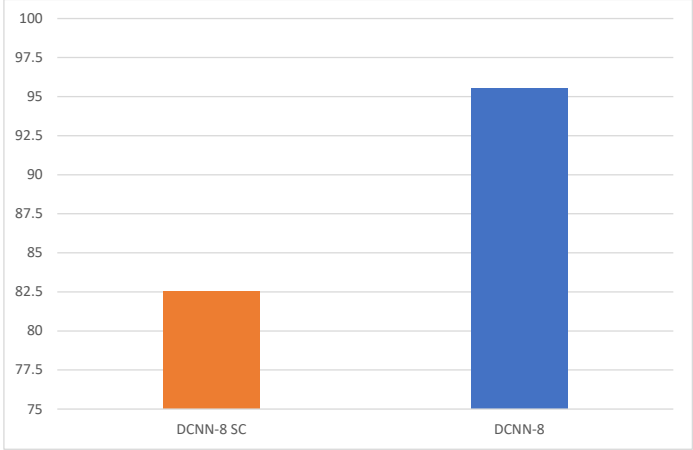


Fig. 5: Comparison of our model with Standard Convolutions model on ESC-50

after DCNN-8. We conjecture that it could be due to overfitting on one or more folds.

To show that separable convolutions work better than standard

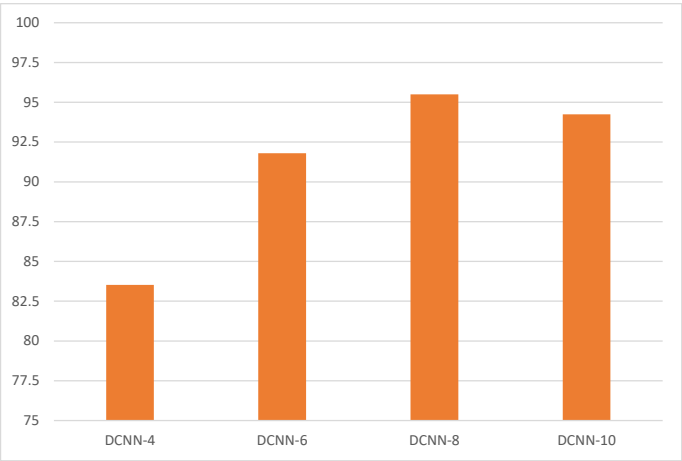


Fig. 4: Comparison of different sizes of the DCNN model on the ESC-50

convolutions for audio signal processing, we test our model with the same architecture, but with one change. We use standard square convolutions (abbreviated as SC) instead of separable convolutions. So, $1 \times m$ becomes $m \times m$ and $n \times 1$ becomes $n \times n$. The same is done for pooling layers. Figure 5 shows the comparison between the DCNN-8 and DCNN-8 with standard square convolutions (DCNN-8 SC) on the ESC-50 dataset. From the figure we can see the huge gap in performance clearly.

The ESC-50 is a difficult dataset due to it's small size and large number of classes. We exhaustively test our model on this dataset and experimentally show the importance of having multi-channel feature input, data augmentation, model size and separable convolutions. Our final model, the DCNN-8 with 4 channel feature input (MFCC, GFCC, CQT and Chromagram) and data augmentation achieves state-of-the-art performance with 95.5% accuracy on the ESC-50 dataset.

The same tests were conducted on the ESC-10 dataset. Due to lack of space we do not include the results of our model on it. But, we also report state-of-the-art performance on the ESC-10 dataset with 97.25% accuracy.

B. UrbanSound8K

The UrbanSound8K is a bigger dataset compared to ESC-10/ESC-50, with a collection of 8732 short (less than 4 seconds) audio clips of various environment sound sources [30]. It has also been widely used by researchers as a benchmarking dataset for their ESC models. Unlike ESC-50, the UrbanSound8K has varying sample rates for audio files. We sample the audio files at 22kHz. The dataset consists of audio signals categorised into 10 disjoint imbalanced classes: air conditioner, car horn, playing children, dog bark, drilling, engine idling, gun shot, jackhammer, siren, street music. So, even though UrbanSound8K has less categories than ESC-50, it has the class imbalance problem which makes generalization difficult.

The dataset has been pre-arranged into 10 folds for unbiased comparable results. We use these pre-determined folds with 10-fold cross validation and report the average accuracy of our model across the 10 predefined folds, as mentioned in [30]. We train our model for 50 epochs per fold and test the model on the remaining fold. We calculate the average of this test accuracy for all folds.

We perform the same tests on our ESC model for the UrbanSound8K dataset, as done for the ESC-50 dataset in the previous sub-section. Table 5 shows the importance of adding multiple feature channels to improve system performance. In table 6, the importance of data augmentation is shown for the UrbanSound8K dataset. Even though the UrbanSound8K dataset has more training data than ESC-50, augmentation still plays an important role in boosting model accuracy.

Figure 6 displays the convergence of the DCNN-8 model on the UrbanSound8K dataset. There is a consistent increment in the average test accuracy after every fold. Here, the model starts off at a good position in the first fold itself and slowly

converges towards a local optimum yielding high performance.

TABLE V: Performance on the UrbanSound8K dataset

Model	MFCC	GFCC	CQT	Chroma	Accuracy
DCNN-8	✓				86.028
DCNN-8	✓	✓			91.748
DCNN-8	✓	✓	✓		96.830
DCNN-8	✓	✓	✓	✓	98.602

TABLE VI: Effect of Data Augmentation for UrbanSound8K dataset

Model	Accuracy
Multi-channel input + DCNN-8	93.251
Multi-channel input + DCNN-8 with strong augmentation	98.602

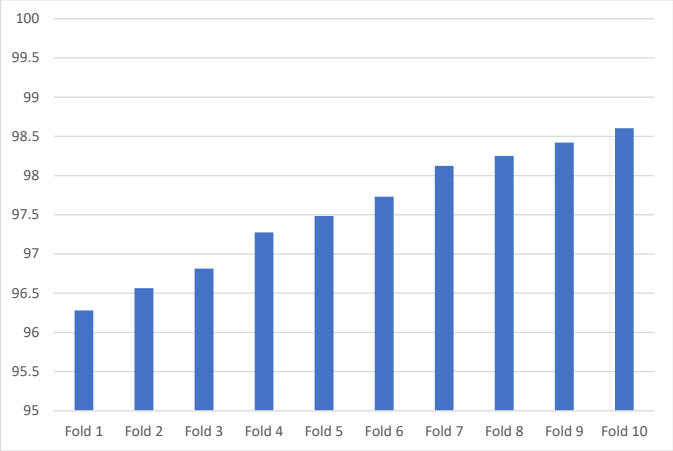


Fig. 6: Moving Average Accuracy per Fold on UrbanSound8K

We try out different number of repetitions of the *Conv2D-Conv2D-MaxPool-BatchNorm* layers. As we can see from Figure 7, DCNN with eight repetitions performs the best. The accuracy of the system increases from a sequence of four to six to eight, but decreases at ten. This might be due to overfitting or imbalanced classification.

To experimentally prove that separable convolutions outperform regular square kernel convolutions, we do the same test as on the ESC-50 dataset. We use the same model for the UrbanSound8K and replace the $1 \times m$ and $n \times 1$ convolutions with $n \times n$ and $m \times m$ convolutions. In figure 8, we show the difference in performance between our DCNN-8 with separable convolutions and DCNN-8 with standard square convolutions (DCNN-8 SC). The gap in the accuracy is quite profound.

The UrbanSound8K dataset, like the ESC-50 dataset, presents many challenges like imbalance class distribution and varying sampling rates. But, our model is still able to perform and set the new state-of-the-art performance on the UrbanSound8K dataset by achieving 98.6% accuracy.

Overall, from our exhaustive experimentation on the benchmark datasets for ESC task, we show that our model achieves

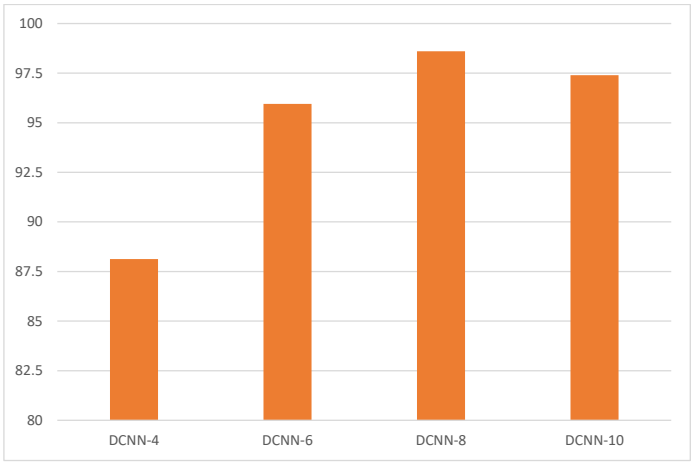


Fig. 7: Comparison of different sizes of the DCNN model on the UrbanSound8K

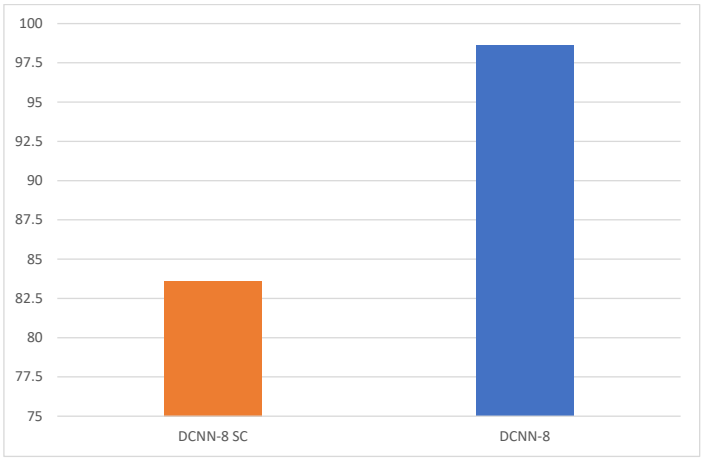


Fig. 8: Comparison of our model with Standard Convolutions model on UrbanSound8K

state-of-the-art performance on all the benchmark datasets. We show the importance of data augmentation in further increasing accuracy and using separable convolutions that separate the processing of time and feature domain information. We also show the importance of the multi-channel feature input and the increase in performance that one can achieve from carefully selecting multiple complementary feature extraction techniques. We explore the size of our DCNN model and choose the DCNN-8 based on the best performance. Our system of multiple feature channel input and separable convolutions Deep CNN with data augmentation achieves state-of-the-art performance on all three environmental sound classification benchmark datasets. To the best of our knowledge, this is the first time that an ESC model has been able to achieve state-of-the-art performance on all three datasets.

V. Conclusion

We propose a novel approach for environmental sound classification that consists of multiple feature channels and

deep convolutional neural network with domain wise convolutions. We combine feature extraction methods like the MFCC, GFCC, CQT and Chromagram to create a multi channel input for the CNN classifier. Each of these feature sets provide some specific and discriminatory information that increases classification accuracy. As the results suggest, each feature set contributes in boosting performance of the model. We employ a Deep CNN consisting of separable convolutions, pooling and batch normalization layers along with Leaky ReLU activation and dropout and L_2 regularization. The convolution and pooling layers work on the time and feature domains separately to extract relevant information from each time window and each feature along the time separately. We test our model on the three benchmark datasets: ESC-10, ESC-50 and UrbanSound8K. We use simple data augmentation techniques like time stretch and pitch shift and add some random gaussian noise to further improve performance. Our model achieves 97.25%, 95.50% and 98.60% accuracy on ESC-10, ESC-50 and UrbanSound8K respectively, which is state-of-the-art performance on all three datasets. To the best of our knowledge, this is the first time when a model has achieved state-of-the-art performance on all three benchmark datasets.

REFERENCES

[1] Dong Yu and Li Deng. *Automatic Speech Recognition: A Deep Learning Approach*. Springer Publishing Company, Incorporated, 2014.

[2] Dong Yu and Li Deng. *Deep Neural Network-Hidden Markov Model Hybrid Systems*, pages 99–116. Springer London, London, 2015.

[3] Liang Lu, Xingxing Zhang, Kyunghyun Cho, and Steve Renals. A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[4] W. Chan, N. Jaitly, Q. Le, and O. Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964, March 2016.

[5] Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Katya Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. State-of-the-art speech recognition with sequence-to-sequence models. *CoRR*, abs/1712.01769, 2017.

[6] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584, April 2015.

[7] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Y. Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Y. Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, and Zhenyao Zhu. Deep speech 2: End-to-end speech recognition in english and mandarin. *CoRR*, abs/1512.02595, 2015.

[8] Md. Sahidullah and Goutam Saha. Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition. *Speech Communication*, 54(4):543 – 565, 2012.

[9] Christian Schörkhuber. Constant-q transform toolbox for music processing. 2010.

[10] Roger N Shepard. Circularity in judgments of relative pitch. *The Journal of the Acoustical Society of America*, 36(12):2346–2353, 1964.

[11] Y. Shao, Z. Jin, D. Wang, and S. Srinivasan. An auditory-based feature for robust speech recognition. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4625–4628, April 2009.

[12] Yizhar Lavner and Dima Ruinskiy. A decision-tree-based algorithm for speech/music classification and segmentation. *EURASIP Journal on Audio, Speech, and Music Processing*, 2009(1):239892, Jun 2009.

[13] Chun-Yan Yu, Huang Liu, and Zi-Ming Qi. Sound event detection using deep random forest. Technical report, Technical report, DCASE2017 Challenge, 2017.

[14] Wang Shuiping, Tang Zhenming, and Li Shiqiang. Design and implementation of an audio classification system based on svm. *Procedia Engineering*, 15:4031 – 4035, 2011. CEIS 2011.

[15] Shruti Aggarwal and Naveen Aggarwal. Classification of audio data using support vector machine. 2011.

[16] Jia-Ching Wang, Jhing-Fa Wang, Cai-Bei Lin, Kun-Ting Jian, and W. Kuok. Content-based audio classification using support vector machines and independent component analysis. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 4, pages 157–160, Aug 2006.

[17] Musatafa Abbas Abbood Albadr, Sabrina Tiun, Fahad Taha AL-Dhief, and Mahmoud A. M. Sammour. Spoken language identification based on the enhanced self-adjusting extreme learning machine approach. *PLOS ONE*, 13(4):1–27, 04 2018.

[18] S. Scardapane, D. Comminiello, M. Scarpiniti, and A. Uncini. Music classification using extreme learning machines. In *2013 8th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 377–381, Sep. 2013.

[19] Y. Tokozume and T. Harada. Learning environmental sounds with end-to-end convolutional neural network. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2721–2725, March 2017.

[20] Zhichao Zhang, Shugong Xu, Shan Cao, and Shunqing Zhang. Deep convolutional neural network with mixup for environmental sound classification. In Jian-Huang Lai, Cheng-Lin Liu, Xilin Chen, Jie Zhou, Tieniu Tan, Nanning Zheng, and Hongbin Zha, editors, *Pattern Recognition and Computer Vision*, pages 356–367, Cham, 2018. Springer International Publishing.

[21] Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *CoRR*, abs/1608.04363, 2016.

[22] Sajjad Abdoli, Patrick Cardinal, and Alessandro Lameiras Koerich. End-to-end environmental sound classification using a 1d convolutional neural network. *CoRR*, abs/1904.08990, 2019.

[23] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Learning from between-class examples for deep sound recognition. *CoRR*, abs/1711.10282, 2017.

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

[25] Jordi Pons and Xavier Serra. Randomly weighted cnns for (music) audio classification. *CoRR*, abs/1805.00237, 2018.

[26] Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das. Very deep convolutional neural networks for raw waveforms. *CoRR*, abs/1610.00087, 2016.

[27] Yu Su, Ke Zhang, Jingyu Wang, and Kurosh Madani. Environment sound classification using a two-stream cnn based on decision-level fusion. *Sensors*, 19(7), 2019.

[28] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[29] Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press.

[30] 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*, MM '14, pages 1041–1044, New York, NY, USA, 2014. ACM.

[31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[32] Rishabh N. Tak, Dharmesh M. Agrawal, and Hemant A. Patil. Novel phase encoded mel filterbank energies for environmental sound classification. In B. Uma Shankar, Kuntal Ghosh, Deba Prasad Mandal, Shubhra Sankar Ray, David Zhang, and Sankar K. Pal, editors, *Pattern Recognition and Machine Intelligence*, pages 317–325, Cham, 2017. Springer International Publishing.

[33] D. M. Agrawal, H. B. Sailor, M. H. Soni, and H. A. Patil. Novel teo-based gammatone features for environmental sound classification. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1809–1813, Aug 2017.

[34] Boqing Zhu, Kele Xu, Dezhi Wang, Lilun Zhang, Bo Li, and Yuxing Peng. Environmental sound classification based on multi-temporal

resolution CNN network combining with multi-level features. *CoRR*, abs/1805.09752, 2018.
[35] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.
[36] J. Salamon and J. P. Bello. Unsupervised feature learning for urban sound classification. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 171–175, April 2015.
[37] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 892–900, USA, 2016. Curran Associates Inc.
[38] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.
[39] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.
[40] Venkatesh Boddapati, Andrej Petef, Jim Rasmusson, and Lars Lundberg. Classifying environmental sounds using image recognition networks. *Procedia Computer Science*, 112:2048 – 2056, 2017. Knowledge-Based and Intelligent Information and Engineering Systems: Proceedings of the 21st International Conference, KES-20176-8 September 2017, Marseille, France.
[41] S. Chachada and C. . J. Kuo. Environmental sound recognition: A survey. In *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pages 1–9, Oct 2013.
[42] Wilson Burgos. *Gammatone and MFCC Features in Speaker Recognition (Doctoral Dissertation)*. PhD thesis, 2014.
[43] Deepanway Ghosal and Maheshkumar H Kolekar. Music genre recognition using deep neural networks and transfer learning. In *InterSpeech*, pages 2087–2019, 2018.
[44] Beth Logan. Mel frequency cepstral coefficients for music modeling. In *ISMIR*, volume 270, pages 1–11, October 2000.
[45] Judith C Brown. Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434, 1991.
[46] Sicong Huang, Qiyang Li, Cem Anil, Xuchan Bao, Sageev Oore, and Roger B. Grosse. Timbretron: A wavenet(cyclegan(cqt(audio))) pipeline for musical timbre transfer. *CoRR*, abs/1811.09620, 2018.
[47] Muhammad Huzaifah. Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. *CoRR*, abs/1706.07156, 2017.
[48] Thomas Lidy and Alexander Schindler. Cqt-based convolutional neural networks for audio scene classification. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, volume 90, pages 1032–1048. DCASE2016 Challenge, 2016.
[49] Jouni Paulus, Meinard Müller, and Anssi Klapuri. State of the art report: Audio-based music structure analysis. In *International Society for Music Information Retrieval*, pages 625–636, Aug 2010.
[50] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.
[51] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudik, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.
[52] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *CoRR*, abs/1505.00853, 2015.
[53] D.P. Kingma and L.J. Ba. Adam: A method for stochastic optimization. In *ICLR*, International Conference on Learning Representations (ICLR), page 13, San Diego, CA, USA, 7–9 May 2015. Ithaca, NY: arXiv.org.
[54] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.
[55] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.
[56] Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, 1974.
[57] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
[58] François Chollet et al. Keras. https://keras.io, 2015.
[59] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. 2015.
[60] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, O. Mayor, Gerard Roma, Justin Salamon, J. R. Zapata, and Xavier Serra. Essentia: an audio analysis library for music information retrieval. In *International Society for Music Information Retrieval Conference (ISMIR'13)*, pages 493–498, Curitiba, Brazil, 04/11/2013 2013.
[61] *MATLAB Signal Processing Toolbox 2019*. The MathWorks Inc., Natick, Massachusetts, United States, 2019.