Neural Computing and Applications
https://doi.org/10.1007/s00521-021-06091-7

ORIGINAL ARTICLE

Rethinking environmental sound classification using convolutional neural networks: optimized parameter tuning of single feature extraction

Yousef Abd Al-Hattab[1] · Hasan Firdaus Zaki[1] · Amir Akramin Shafie[1]

Received: 25 July 2020 / Accepted: 28 April 2021

Abstract

The classification of environmental sounds is important for emerging applications such as automatic audio surveillance, audio forensics, and robot navigation. Existing techniques combined multiple features and stacked many CNN layers (very deep learning) to reach the desired accuracy. Instead of using many features and going deeper by stacking layers that are resource extensive, this paper proposes a novel technique that uses only a single feature, namely the Mel-Frequency Cepstral Coefficient (MFCC) and just three layers of CNN. We demonstrate that such a simple network can considerably outperform several conventional and deep learning-based algorithms. Through parameters fine-tuning of the data input, we reported a model that is significantly less complex in the architecture yet has recorded a similar accuracy of 95.59% compared to state-of-the-art deep models on UrbanSound8k dataset.

Keywords Convolutional neural networks (CNN) · Mel-frequency cepstral coefficients (MFCC) · Environmental sound classification · Feature extraction · Urbansound8Kdataset

1 Introduction

Intelligent Sound Recognition (ISR) is a technology that identifies sound events in the natural environment possible. This technology is mainly used in applying human perception of auditory signals to devices, machines, or robots. Essential development and part of ISR is environmental sounds classification (ESC), which recognizes and classifies detected sounds such as the barking of a dog, car horn, and child's crying. The challenge in classifying environmental sounds lies in the fact that it is non-stationary. This characteristic puts it in contrast with the classification of stationary sounds such as the human voice with automatic speech recognition (ASR) [1] and music through music information recognition (MIR) [2]. Due to this contrast, these methods are insufficient for the classification of environmental sounds. Therefore, there is a need for an ISR system that can satisfy and accurately recognize and classify environmental sounds.

The classification of environmental sounds comprises of two main components, which are acoustic features and classifiers. For extraction of acoustic features, detected sound events are divided into frames with a cosine window function, namely Hamming or Hanning. Afterward, each frame feature is extracted, used as one instance of training or testing [3]. The sum of all predicted probabilities of the separate segments becomes the signification result of a particular sound. Studies have shown that Mel-Frequency Cepstral Coefficients (MFCC) and Log Mel spectrogram, which are initially developed for ASR, perform considerably in ESC [4, 5]. Furthermore, researchers advise combing several features in sound recognition for an increase in performance.

Some of the most applied classifiers for sound event recognition are Gaussian Mixture Modeling (GMM), Hidden Markov Model (HMM), and Support Vector Machine (SVM) [6–8]. The greatest challenge of the 'traditional classifiers' lies in the fact that they can only model small variations. This leads to a lack of time and frequency invariance. The recent development in sound classification

✉ Yousef Abd Al-Hattab
alhattab.yousef@live.iium.edu.my

[1] Department of Mechatronics Engineering, International Islamic University Malaysia, 53100 Gombak, Selangor, Malaysia

Published online: 26 May 2021

proposes deep neural network-based models in classifying environmental sounds. Models such as the convolutional neural network (CNN) have proven to perform better than traditional classifiers. For instance, CNN solves the lack of time and frequency invariance problems with its learning filters [9]. CNN's strength is in its design that allows data processing in the form of multiple arrays, namely 1D for sound events like speaking or music and 2D for image or audio spectrograms [10]. CNN is applicable not only for ASR and MIR [4], but it's also highly competent in performing ESC tasks.

Even though CNN is a significant improvement, there is a lot to gain when juxtaposed with CNN based image classification algorithms. Researchers have proposed merging several CNN or combining CNN with other deep learning models [11, 12]. These methods first use CNN(s) to extract spatial information consisting of different audio features. The outputs are then merged by concatenation and fed into a Recurrent Neural Network (RNN) or other CNN layers for temporal information extraction.

Despite a higher performance by combining several acoustic features, implementing deeper CNN layers by including more CNN, or other deep learning models, results are still not satisfactory. Moreover, it leads to the higher complexity of the model and a significant amount of time for training and testing. Hence, the current study proposes a novel three-layer stacked CNN architecture SEnv-Net while using a single acoustic feature. It is argued that additionally to being a simple model, and this architecture is sufficient in accomplishing a high accuracy performance in ESC tasks. This will be discussed further; later on, for now, some of the notable works using CNN are mentioned subsequently.

1.1 Related works

Many studies showed that CNN-based models outperform conventional methods in various tasks in recent years, whether the fed input is 2D or 3D [13]. The study [14] was the first in applying CNN in ESC tasks. Their architecture was build up with two CNN layers with max-pooling. They used Log Mel spectrograms as an auditory feature in training the CNN. They reached a higher accuracy than traditional classification methods. The study of [15] investigated which neural network is most fitting as a classifier. They showed that by using a fully connected layer, a convolutional layer, and a layer without max-pooling, convolutional layers classifiers outperform other methods.

Through image recognition problems, the critical aspect of CNN in extracting features from raw inputs is verified. The researchers of [16] proposed the usage of CNN in obtaining raw waveforms while using SVM or extreme learning machines (ELM) as classifiers in ESC tasks. Even though their results suggest a better performance than CNN with an MFCC feature, their accuracy is merely 70–74%. Another study using raw waveforms is [17]. They studied the most optimal amount of CNN layers for an excellent performance. They found that going more in-depth in layers does not mean better performance and that using raw waveforms leads to similar performances of models using Log Mel spectrogram features.

Traditional CNN models have deficiencies for audio tasks. For instance, even though applied pooling layers reduce feature dimensions, it can also be a source of data loss and hindrance to the neural network performance. As it stands, it remains relatively difficult to explore CNNs and improve existing models. This is due to a scarcity of labeled data. A solution to this problem is data augmentation. This is the deformation of existing data, whereby new training data samples are created. Salamon and Bello [18] performed data augmentation for audio signals from an environmental data set. Methods of data augmentation for environmental sound data include dynamic range compression, pitch shifting, adding noise, and time stretching, all of which were used and analyzed on the UrbanSound8k dataset. It is also shown that class-conditional data augmentation further improves outcomes.

Other studies aimed for performance improvement by exploiting CNN models, which were developed for image recognition and applied them in audio tasks. For example, Boddapati et al. [19] used AlexNet and GoogleNet for ESC. Features like MFCC, cross-recurrence plot (CRP) of audio signals, and spectrogram were derived and used as image representations. These mixed features were after that used as an input for AlexNet and GoogleNet. The former achieved a classification accuracy of 92%, while the latter reached 93% on benchmark ESC datasets.

Another approach is the end to end that has its base in feature extraction and classification of audio signals. An approach of end to end was proposed by [20], which is primarily based on raw audio signals being worked by 1D-CNNs. Raw audio waveforms can be picked up with the help of 1D-CNN environmental sounds on variable length. To add to this, no feature extractions become necessary, as long as the first layer is initialized as a Gammatone filter bank in the 1D-CNN. The decomposition of raw input yields 64 frequency bands through which initialize the first layer of the convolution kernels by 64 bandpass Gammatone filters.

A rather complex structure was proposed by [21]. Their two-stream structure CNN model consists of two network streams that were joined at the end with decision-level fusion. The first is the MCNet, which takes Chroma, MFCC, Spectral Contrast, and Tonnetz as inputs. Simultaneously, the second stream works on the Chroma, Spectral Contrast, Tonnetz, and Log Mel spectrogram features of audio signals. The decisions were fused with the Dempster–Schafer evidence theory, through which the two-stream CNN based on decision-level fusion (TSDCNN-DS) model is obtained.

The final study to be discussed in [22] proposed a model consisting of multiple feature channels and a deeper CNN (DCNN) consisting of 2D separable convolutions. Furthermore, the included max-pooling layers for down-sampling of time and the feature domain separately. They found an accuracy of 97.35% for the UrbanSound8K dataset, which is the same dataset as in this study.

Note from the studies mentioned above that the bulk of ESC models use raw waveforms or aggregate auditory features for training the neural networks. Despite reaching higher accuracies and apparent improvements, it is observable that CNN based ESC models still have room for progress. Combining several features or CNNs often leads to complex models without the desired results while simultaneously taking a lot of time for training and testing. Moreover, after a comprehensive investigation of a considerable number of sound recognition works, we noticed that none of them paid full attention to the impact of using a significant feature instead of combing many features to reach the desired accuracy. Hence, as argued above, in the current paper, we show that using one audio feature and a simple CNN model as SEnv-Net, with novelty in fine-tuning the parameters of data input, suffices for achieving state-of-the-art performance.

This shall further be elaborated in the next sections. First, there is an explanation of the proposed framework. Afterward, the feature representation and methodology are subsequently elucidated. Then, the results and discussion are provided in the last section, closing with an overall conclusion.

2 Proposed framework

This study is based on sound event detection (SED) for monophonic urban sounds. SED detects and classifies each individual sound event. Source separation makes it possible to form a mixed-signal into a clean audio stream. With SED, there is a possibility for the classification of sounds, which is the determination of the type of sound. This monophonic sound event is fed into the SED system. An abstract overview of the SED system is given in Fig. 1.

The current study uses a time–frequency representation of the audio signals for feature extraction before normalizing and categorizing in ten folds. The audio signals, extracted from the UrbanSound8K dataset audio, were converted to a log scale time–frequency representation using the Mel-Frequency Cepstral Coefficient (MFCC). Afterward, the data was fed into the three layers of CNN. A three-layer CNN is chosen in this study after conducting training sessions with various layers, whereby three layers resulted in the best performance. Literature research shows that CNN is the most satisfactory existing neural network in SED. The proposed CNN model is evaluated using tenfold cross-validation. Subsequently, we tuned the Adam optimizer parameters in order to improve the model behavior. This network was implemented in a deep learning library called TensorFlow.

3 Feature representation

Audio events in the analysis process are regularly based on features such as acoustic that are extracted from audio signals. The audio can be represented then in a non-redundant and compact way. Algorithm recognition requires properties of acoustic features that have low variability among extracted features from examples that are usually assigned from the same class. Concurrently, high variability can allow discrimination between features that are extracted from examples that are assigned to each other [23] —representations of a feature that fulfill this property commonly more often than not make the learning problem easier. Additionally, the amount of memory and computational power becomes reduced, and less is necessary when a feature is compact.

To maximize the audio recognition performance, the audio signal's transformation is done into another representation that is also known as the function of feature extraction. Relevant to machine learning, the acoustic features represent the audio signal content in a numerical representation. Also, the audio signal is characterized by values that link or refer to the physical properties as well as dynamic range compression, pitch shifting, adding noise, and time stretching.

3.1 Feature extraction in SED

Audio signals for SED are acquired by recording the sound event in a studio or in a real-life environment. Time domain representation is the lowest level of representation, due to the reason of the signal not being process a lot before being used as a representation. While at the same time, the representation is redundant for a classifier in order to learn which sound event it belongs to. Thus, in SED situations, audio signals are usually, if not frequently, represented by certain acoustic features' extractions. The extraction of these features comes mainly from the frequency domain; since these signals share the same sound events normally share components and parts in the frequency domain.

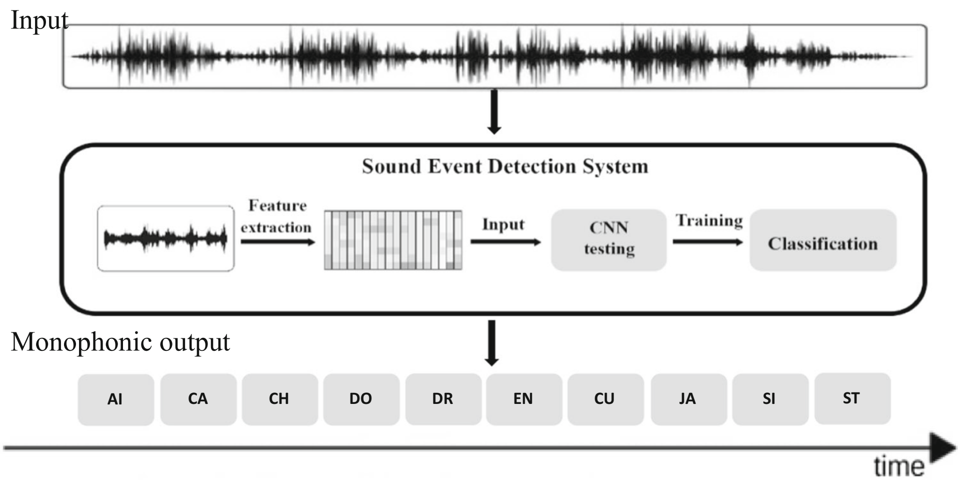


Fig. 1 Each event with sound class label + onset and offset timestamps

Furthermore, frequency domain representation is compact and more noise-robust than time domain representation. The number of processing steps determines the abstraction level of the sound representation in the time domain signal. There is a multitude of features that can be used in SED. In this paper, the most common are elaborated, which are the MFCC, Mel spectrogram, Chromagram.

(1) The Mel-frequency Cepstral Coefficients (MFCC) is a standard feature extraction technique that has performed well in benchmark ASR as well as ESC models [24]. MFCC development was motivated by the human auditory perception, and it produces a compact representation of an audio signal. The main difference between other cepstral features is that the frequency band is on the mel scale. We normalized the MFCC because of its susceptibility. In an MFCC, each short time window's FFT of a given audio signal input is calculated. Afterward, the FFT power of the spectrum is mapped into the Mel scale using triangular Mel-filter banks. These filter banks resemble essential human auditory system properties. After that, the logarithmic scale is applied to the Mel-spectrum to underline the signal's low varying frequency characteristics. In the final step, the discrete cosine transform (DCT) is computed for the de-correlation of the filter bank energies, while a number of cepstral features are held for classification. The frame size, or the set of parameters, the amount of Mel bands, and the number of DCT coefficients are necessary to be defined when computing the MFCCs. With these parameters, the dimensionality of the features and their performance are determined [21]. In Fig. 2, the five steps of an MFCC are provided.

(2) Another widely used feature is Mel spectrogram [14]. This feature is similar to MFCC. Nevertheless, the critical difference is that Mel spectrogram works with a linear spaced frequency scale. As a result, each frequency bin is spaced with equal numbers of Hertz apart from each other. The MFCC, on the other hand, uses a quasi-logarithmic spaced frequency scale, which is more like the human auditory system. In general, the MFCC can be seen as an extended Mel spectrogram, whereby the log and DCT are included.

(3) The third feature that has a good record of performance is the Chromagram. This feature is mainly applied in the music audio signal [26]. It is applicable for the pitch analysis of audio signals. It is useful for the discrimination between audio signals through pitch class profiles, which makes it exceptionally capable in audio structure analysis. Chromagram features are computed using a Short-Time Fourier Transform (STFT). Some studies have proposed the use of clusters of multiple feature sets that use divergent extraction methods for signal characteristics to achieve better performance [14]. Nevertheless, the MFCC method is the backbone feature, as it provides rich features and has the best performance in sound event detection. For this reason, this paper uses only the MFCC feature and shows that it is enough to achieve significant accuracy. Before we continue to justify this choice, we deem it necessary to explain the steps for audio feature extraction, as discussed in the following section.

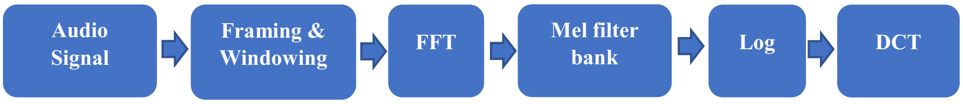


Fig. 2 Flow of MFCC computation

3.2 Steps of audio feature extraction

In the case of audio feature extraction, if we take, for example, SED, it becomes clear that it has three stages, which are blocking of frames, windowing, and calculations in frequency system. Short-Time Fourier Transform (STFT) is used to attain frequency spectrum, and the signal should have the capability to model a sum of stationary sinusoids. Thus, by dividing the signal Initial into short time frames, the frequency spectrums are calculated; this is the so-called frame blocking.

According to the frame length, there is a trade-off between the time and frequency resolution, in which the resolution of frequency increases as the frame length increases. This is a significant cause of diminishing time resolutions. This is the reason why selections of frame length become dependent upon the machine hearing task. When it comes to SED, the frame length is usually selected in the range of 20–60 ms. An overlap of 25–50% of the frame length is chosen to create smoother representations. Thereafter, through windowing, each short time frame signal is multiplied to avoid discontinuities at the borders of the frame. Otherwise, it can corrupt the frequency spectrum estimation. When it comes to SED, Hamming, Hann, and Blackman functions are usually used. After this, the frequency domain representation of every short time frame signal is acquired by calculating the Discrete Fourier Transform. The stages of audio feature extraction in the frequency domain are visualized in Fig. 3.

4 Methodology

This paper's central goal is the investigation of sound classification using deep learning networks that are designed for object recognition in images. The previous researches achieved high accuracy using multiple features. On the contrary, the novel ESC model and the CNN architecture in this paper, using just a single significant extracted feature from the audio signal. The CNN is used as the audio signal classifier for the ESC task. The audio signal is changed into a monophonic signal. An illustration of the sequential process of SEnv-Net is provided in Fig. 4.

4.1 Proposed network architecture

The SEnv-Net architecture proposed in this paper consists of three CNN layers intertwined with two MaxPooling2D operations and two fully connected dense layers. This model architecture is chosen as it performed best in the conducted empirical evaluation of different configurations. This method of extraction is similar to previous works (e.g., [22]). However, we studied the parameters of the data input, which has been neglected before. Figure 5 shows the architecture of the proposed network. An overview of the different layers is as follows:

- l_1: Conv2D 32 with a receptive field of (5, 5). This is followed by (4, 2) MaxPooling2D and a rectified linear unit (ReLU) activation function.
- l_2: Conv2D 64 with a receptive field of (5, 5). Like l_1, this is followed by (4, 2) MaxPooling2D and a ReLU activation function.
- l_3: Conv2D 128 with a receptive field of (5, 5). This is followed by a ReLU activation function (no MaxPooling2D).
- l_4: Flatten layer.
- l_5: One hundred twenty-eight dense layers followed by a ReLU activation function.
- l_6: 10 output units followed by a Softmax activation function.

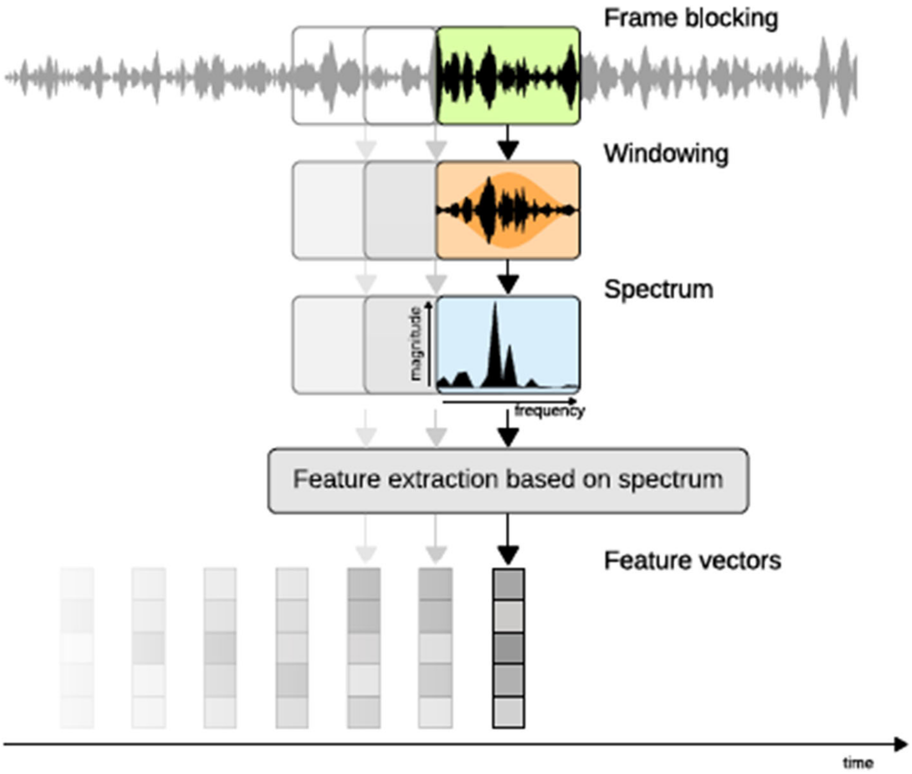
The small receptive field (5, 5) in l_1 Compared to the input dimensions (128, 128), it allows learning in small, localized patterns in the network. These patterns are fused at successive layers to gather evidence supporting larger time-frequencies, which indicate the presence or absence of various audio classes. This is regardless of spectro-temporal masking by interfering sources. With MaxPooling2D, the dimensions of the output feature maps are reduced, and therefore the training is fastened. Moreover, it builds a few scale invariances in the network. The commonly used activation function is ReLU [27], of which the equation is provided in Eq. (1). The final layer is the Softmax function, which is used to attain class probabilities. The equation can be founded before Eq. (2) where e^{x_i} is standard exponential function for input vector and e^{x_j} is standard exponential function for output vector.

$$f(x) = \max(0, x) \tag{1}$$

$$S(x_i) = \frac{e^{x_i}}{\sum_{j=0}^{C} e^{x_j}} \tag{2}$$

Softmax is used together with cross-entropy loss since they provide a gradient that makes computations much easier. The equation is shown in Eq. (3).

Fig. 3 Stages in Frequency domain sound feature extraction [23]



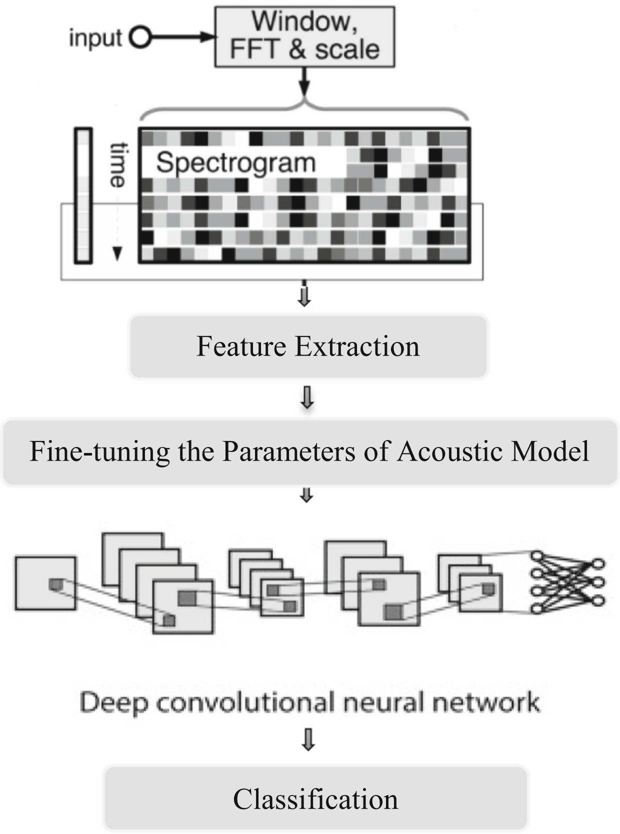

input
Window, FFT & scale
time
Spectrogram
Feature Extraction
Fine-tuning the Parameters of Acoustic Model
Deep convolutional neural network
Classification

Fig. 4 SEnv-Net: Sequential process of a sound event classification of the proposed system

$$L = -\sum_{i=0}^{C} t_i \log(S(x_i)) \tag{3}$$

4.2 Model ablation experiment

After building the model and selecting the number of layers, we tuned the hyperparameters and observed the model's behavior. First, we performed Dropout [28], which is a way of regularization. In ascending order, the probabilities of Dropout applied to the selected model were 0.2, 0.3, 0.35, and 0.5 for the last two layers' input. In this study, Dropouts were adjusted to the number of hidden layers in each CNN layer. This means that higher Dropout probabilities were applied to CNN layers with more hidden layers and vice versa.

Through experimental exploration in the form of trial and error, an empirical evaluation of the proposed model, the Dropout as mentioned above probabilities were found recorded and selected.

The other way of regularization that was applied in this paper is L2-regularization, which penalized the magnitude of the weights and reduced redundancy. This was used to the weights of the last two dense layers with regularization parameter $\gamma = 0.001$.

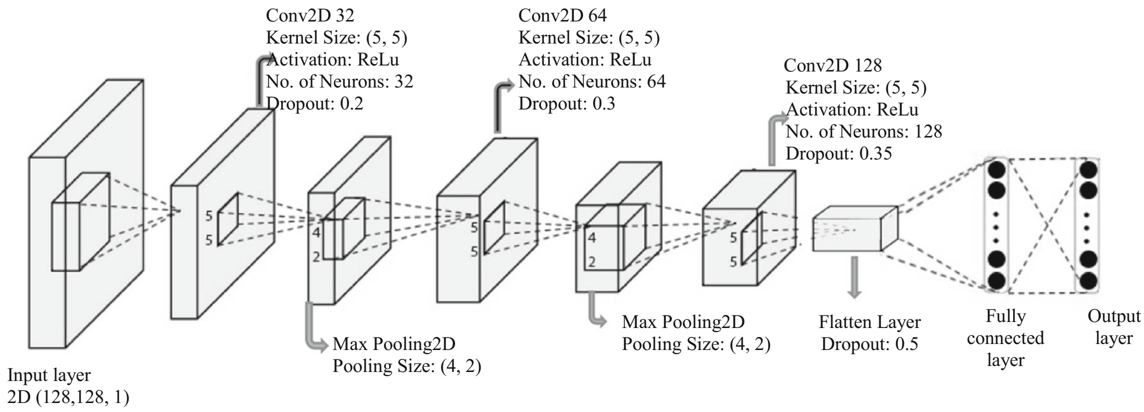


Fig. 5 The architecture of the proposed model

4.3 Adam optimizer

In an adaptive learning way, Adam optimizes algorithms that are specifically designed for the training of deep networks, like CNN and Convolutional Recurrent Neural Network (CRNN) [29]. It computes learning rates individually for different parameters. It tunes three parameters in total.

The most critical parameter is the learning rate. This is the amount the weights are updated during training. It is also called a step size. In order to find the value of the learning rate, the following Eq. (4) is used:

$$\alpha = 0.95^{x}.\alpha_0 \tag{4}$$

Here x stands for the number of epochs and α_0 for the expected potential with a value range of 0.1–0.9. Before tuning the hyperparameters, the model was trained with 200 epochs. At epoch 123 onwards, the model barely learned, whereby 123 was chosen as x. With $x = 123$ and $\alpha_0 = 0.1$ the result was $\alpha = 0.00018197$. The model was trained until $\alpha_0 = 0.6$. After that, the accuracy difference was minimal. This is further elaborated in the results section. After exploratory nudging upwards, we found an optimal learning rate of $\alpha = 0.00018964$. α was included in the training model, and the model started learning. After epoch 99, the models' performance stagnated.

The second Adam optimizer parameter is the beta, which stands for the exponential decay rate. The values of beta1 for the first moment estimates were 0.9. Moreover, the value of beta2 for the second moment was 0.999. These values were used throughout the whole study.

The third and final parameter of Adam is the Decay rate. The decay rate is the way in which the learning rate changes over time (training epochs) In order to find the decay rate, the following Eq. (5) was applied:

$$\text{Decay rate} = \text{learning rate } (n)/\text{epochs} \tag{5}$$

In our paper, this meant 0.00018964/99 = 0.0000019156.

Furthermore, it is worth mentioning is that the input is normalized to a unit vector so that the interval covariance shift is reduced. Herewith the input distribution of a layer changes because of changes in the parameters of previous layers. This requires the lowering of the learning rate that slows down training. With the normalization of both dimensions, the spectro-temporal energy distribution pattern is preserved. Also, it eliminates the difference between the audio clips throughout the dataset in terms of linear distortion.

4.4 Main experiment

The main experiment had two steps. Firstly, we studied how selecting the feature that needs to be extracted affects classification accuracy. Meanwhile, we ran the code using three features, namely MFCC, Mel spectrogram, and Chromagram. All three had the same Mel-filter bank of 128 bands. We chose 128 bands since most previous studies showed that this band has the most robust performance [22, 30]. In this way, 128-dimensional features were produced, whereby a frequency range from 0 to 22,050 Hz was covered.

Using the Windowing method reduces the amplitude of the discontinuities at each of the boundaries of each finite sequence acquired by the digitizer. This method consists of multiplying the time record by a finite-length window, making sure that the amplitude varies smoothly and gradually toward zero at the edges. Thus making the endpoints of the waveform eventually meet. This ultimately results in a continuous waveform without sharp transitions. This methodology is also known as applying a window. Thus, we included in the main experiment two types of a window when we extracted the MFCC. These windows are called

Hamming and Hanning. Hanning function is written like this:

$$\omega(n) = \frac{1}{2} - \frac{1}{2}\cos\left(\frac{2\pi n}{M-1}\right) \tag{6}$$

And, the Hamming function is this:

$$\omega(n) = 0.53836 - 0.46154\cos\left(\frac{2\pi n}{M-1}\right) \tag{7}$$

where M is the amount of data in the dataset input of FFT, and n is a number from 0 to $M-1$.

The main difference between these windows is that the Hanning window begins and ends at zero while removing any discontinuity. The Hamming window ends just slightly apart from zero. This has as a result that there is a shy discontinuity. Figure 6 illustrates these characteristics of the Hamming and Hanning windows.

While selecting the most fitting features, we studied what effect changing the input parameters would have on the classification accuracy, as shown in Table 1. We studied the lengths of the FFT window and the Hop length of the data, which is the number of samples between successive frames. The Fourier Transform of a block of time data points is recognized as a Fast Fourier Transform (FFT). A time domain representation of a signal is deconstructed by the Fourier transform into the frequency domain representation. The Fourier transform receives and breaks down a signal into sine waves of different amplitudes and frequencies. A series of sines after that represent all signals in the time domain.

After representing the single in the frequency domain, we need to determine the FFT window's length. To do achieve this, we tried two different values of it (2048 and 1024) to figure out its impact when applied to feature extractions. We analyzed our time domain signal using the same size FFT and hop length. To ensure the peaks of the system and hop length are in the right positions, we increase FFT size. Our experimenters reveal that the increase in the length of the FFT window improves the performance of SED. This denotes that the spectrum looks more 'smooth,' and in some cases, it allows for better visuality as to where peaks might exist. Essentially, we are, in fact, interpolating the frequency.

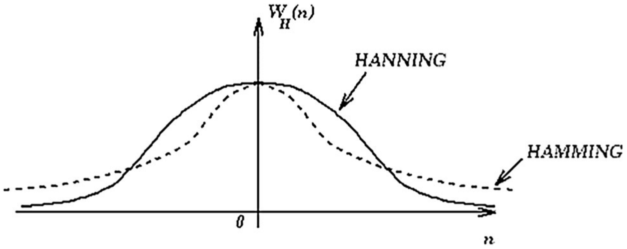


Fig. 6 Continuity of Hamming and Hanning windows

The second one is the hop length. We tried to increase the number of samples between successive frames by increasing the hop length. The experiment showed that decreasing the hop length of input might reduce accuracy since it might nudge some important existing between the frames. Overall, the window size influences the temporal or frequency resolution of the analysis. The more samples, the more slices of frequency range we get, and the more precise these slices are. We used in our experimenters three values of hop length. The first one was equal to the size of the FFT divided by the overlap factor, which was 512. Then, we tried the same values for FFT and hop length because when the number of samples in a window is equal to the window size, the frequency band of a sample is the same as the window's frequency band. Finally, we tried to increase the values of both of them to its maximum, which resulted in the best among the other values.

4.5 Model training

The proposed model is evaluated using tenfold cross-validation. SEnv-Net is trained for a duration of 75 epochs, with checkpoints after each epoch. Both models were implemented with Keras. The resampling of the audio clips and the extraction of the MFCC is done by the Librosa Python library. Each image of the input was fed into the network using the model fit generator. Existing generators were used for reshaping the data. The simplified validation algorithm is founded in [30].

The dataset used in this study is UrbanSound8k [31]. It contains 8732 audio clips with a maximum duration of 4 s, divided into ten environmental sound classes. These classes are Air conditioner, Engine idling, Dog bark, Engine idling, Gunshot, Car horn, Children playing, Jackhammer, Drilling, Siren, and Street music. The proposed approach is evaluated in terms of classification. The dataset was randomly divided into ten same size stratified folds. The dataset used in this study is the UrbanSound8k, because of the fact that it is an imbalanced dataset. Furthermore, it has a varying sampling rate and audio lengths. This makes the dataset a better reconstruction of daily environmental sounds, thus more appropriate. Other urban datasets are more predictable, thus less 'realistic,' and less suitable.

5 Results and discussion

This paper shows that using a single feature and just three layers, not very deep CNN is sufficient to reach high accuracy. The core difference between our technique and other existing algorithms lies in the model's architecture and the fine-tuning of parameters of the feature. With only a three-layer CNN and one audio feature, we found the

Table 1 Experiments for Features Extraction

Feature	Mel-filter bank	N_FFT	Hop length	Window	K-fold val.
Chromagram	128	2048	2048	Hamming	K= 10
Mel spectrogram	128	2048	2048	Hamming	K= 10
MFCC	128	2048	2048	Hamming and Hanning	K= 10
	128	1024	512	Hamming	K= 10
					K= 5
	128	1024	1024	Hamming	K= 5

Table 2 Single Feature Classification accuracy

Model	Feature	Accuracy (%)
1	MFCC	95.59
2	Mel spectrogram	90.61
3	Chromagram	78.54

third-highest accuracy up to date evaluated on the Urban8kSound dataset.

We report the proposed method's performance using the MFCC feature as an input, which is extracted with tenfold cross-validation and evaluated at a sample rate of 44,100 Hz. The results show an average accuracy of 95.59%, as depicted in Table 2. This is the highest achieved accuracy to date on the UrbanSound8K dataset while using single feature-based classification.

In Table 2, the results for the three features are depicted. As observed, the MFCC has the highest accuracy. The worst-case and average-case accuracy of MFCC based classification performance are significantly higher than other single feature-based classification, as shown in Fig. 7. MFCC provides some positive values in every region, with ranges of capabilities that make it stands out from the other two features. It functions as the cardinal stone of the input and supplies robust features with the highest accuracy while maintaining the parameter's same value.

For comparison, we extracted the MFCC feature input at the same sample rate with two different parameters in fivefold cross-validation, as shown below in Table 3.

In Table 4, the results for Hamming and Hanning are illustrated. It can be observed that Hamming has a higher accuracy of over 2.25%. This suggests that Hamming is more fitting for Sound Event Detection tasks since Hamming could reduce the height of the maximum side-lobe.

To provide a comparison with previous studies, we want to display results of other researches without feature extraction, namely Raw waveforms, which are representations that are directly taken from the audio sound event from the Urban8kSound, meaning without any handcrafted feature as shown in Table 5. As a result of this, in [32], CNN performed worst with an accuracy of 66.30%. Nevertheless, the best result of an end-end approach was found in [20], where the researchers used a one-dimensional CNN model. They found the highest accuracy of

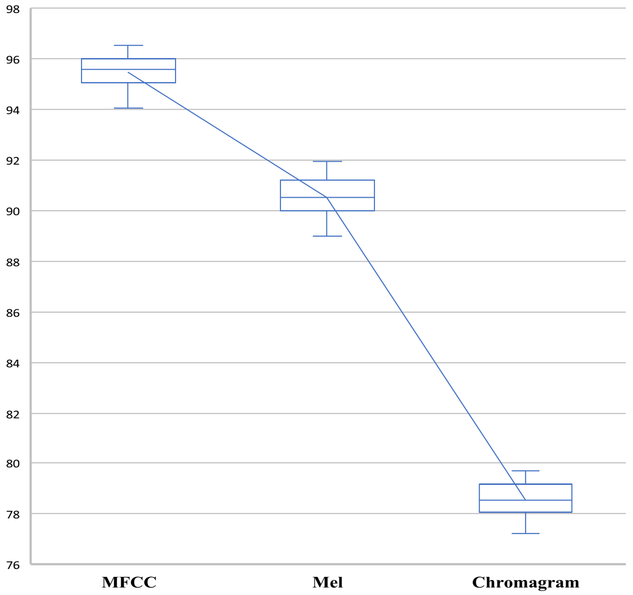


Fig. 7 Classification Accuracy, Average accuracy is 95.59, 90.61and 78.54% for MFCC, Mel spectrogram (Mel), and Chromagram, respectively

Table 3 Classification results for different K size

Window size	Hop length	K-fold cross-val	Accuracy (%)
2048	2048	K = 10	95.59
1024	512	K = 10	92.93
		K = 5	91.95
1024	1024	K = 5	93.87

Table 4 Classification results for different windows

Window size	Hop length	window	Accuracy (%)
2048	2048	Hamming	95.59
2048	2048	Hanning	93.24

Table 5 Raw waveforms results

References	Feature used	Classifier	Accuracy (%)
[32]	Raw waveforms	CNN	**66.30**
[20]	Raw waveforms	1D-CNN	**89.00**
This work	MFCC	CNN	95.59

Table 7 Result studies with various number of layers

Reference	Net depth	Accuracy (%)
[19]	Shallowed	92
[22]	Six layers	92.25
[19]	Twenty-two layers	93
This work	Three layers	95.59

89% with raw waveforms; this work shows a significant achievement in the accuracy through MFCC.

We used the MFCC to visualize the spectrum of frequencies in a particular audio event and the variation in a period. Our model outperformed baseline implementations that rely on Mel spectrogram. As mentioned in the method, MFCC uses a quasi-logarithmic spaced frequency scale, which is the best feature for allowing the visualization of the sound frequencies to be in concordance with the original audio fragment.

All previous studies that used Mel spectrogram as their main feature performed less. In [14], where the model structure was CNN and [30], only Mel spectrogram was used with LSTM as their model structure. These studies did snot reach a higher accuracy than 83%. The highest efficiency of [18] was 79%, whereby data augmentation was applied. The highest accuracy was found in [19], whereby AlexNet was used to combine three features. None of these studies implemented an architecture that was similar to our research. Table 6 gives an overview of studies that used Mel spectrogram as their feature.

In [22], they used a similar model as ours. They used MFCC and a very deep six-layer CNN, but only reached an accuracy of 86%. Afterward, they used four features, namely MFCC, Gammatone Frequency Cepstral Coefficients (GFCC), Constant Q-transform (CQT), and Chromagram, without data augmentation. This made them reach 92.25%. Their window size was 1024, and the Hop length 512. An overview is provided in Table 7. They also fed an eight-layer CNN, but thereby their accuracy decreased. The main difference with the current study is the difference in the architecture of the model and the method of validation. We showed that this is an essential aspect of reaching high accuracy.

The study with the highest accuracy up to date is [21], whereby they combined three features and a Two-Stream CNN based on Decision-Level Fusion. They achieved 97.15%. Whereas, we used only three CNN layers, single feature extraction, and reached the start of art accuracy. This proves that increasing layers does not necessarily imply an increase in accuracy. In fact, in [22], a model consisting of eight layers of CNN showed a decrease in accuracy, in comparison with a six-layered CNN. Our proposed technique shows that going very deep in the number of CNN layers makes no avail.

In comparison with [22], AlexNet, and GoogleNet [19], we achieved the highest accuracy while using the least amount of layers. AlexNet used a shallow method, which is not a deep learning method, and reached 92%. GoogleNet achieved a higher accuracy with a deep, twenty-two layers of CNN, model. Our proposed method used only three layers without pre-training. The above is illustrated in Table 8. Therefore, we propose that the focus should not be on the amount of CNN layers per-se, but more on the method of modeling the input representation.

The uniqueness of this study is the difference in parameters and architecture of the model. We used a relatively easy feature extraction method, just three layers, and only a single feature. This makes our architecture simple and more robust. Moreover, we included window Hamming and Hanning. The results show that even with a higher amount of validation data (fivefold cross-validation), the average accuracies are relatively high.

Table 6 Overview studies using Mel spectrogram

References	Classifier	Feature used	Accuracy (%)
[14]	CNN	Mel spectrogram	**73.00**
[18]	CNN	Mel spectrogram	**73.00**
[18]		Mel spectrogram + data augmentation	**79.00**
[30]	LSTM	Mel spectrogram	**83.00**
[19]	AlexNet	Mel spectrogram, MFCC, CRP	**92.00**
This work	CNN	MFCC	95.59

Table 8 Results of study [22] with variations in number of layers and feature

Net Depth	Feature used	Accuracy (%)
Six layers	MFCC	86.00
Six layers	MFCC, GFCC, CQT and Chromagram	92.25
Eight layers	MFCC, GFCC, CQT and Chromagram	90.10

6 Conclusion

This paper proposes a simple yet effective algorithm for environmental sound classification, named SEnv-Net. We argue that careful modeling of the input representation is more crucial for environmental sound classification than progressively increase the depth of a learning model. Our experiments show that a simple model that uses only one audio feature and simple three layers CNN is sufficient to achieve considerable performance gain compared to other deep learning-based methods when evaluated on the UrbanSound8k dataset. Therefore, we propose that future studies should focus more on the setting and fine-tuning learning parameters and input data instead of using more features, complex network architecture, or stacking in more network layers, which are computationally expensive and resource hungry.

Declaration

Conflict of interest We have no conflicts of interest to disclose.

References

1. Ali H, Tran SN, Benetos E, Garcez ASDA (2018) Speaker recognition with hybrid features from a deep belief network. Neural Comput Appl 29(6):13–19
2. Ghosal, D, Kolekar MH (2018) Music genre recognition using deep neural networks and transfer learning. In: Interspeech, pp 2087–2091
3. Chachada S, Kuo CCJ (2014) Environmental sound recognition: a survey. APSIPA Trans Signal Inf Process 3
4. Zhang Z, Xu S, Cao S, Zhang S (2018) Deep convolutional neural network with mixup for environmental sound classification. In: Chinese conference on pattern recognition and computer vision (prcv), Springer, Cham, pp 356–367
5. Shkurti F, Chang WD et al (2017) Underwater multi-robot convoying using visual tracking by detection. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 4189–4196. IEEE
6. Chu S, Narayanan S, Kuo CCJ (2009) Environmental sound recognition with time–frequency audio features. IEEE Trans Audio Speech Lang Process 17(6):1142–1158
7. Giannoulis D, Benetos E, Stowell D, Rossignol M, Lagrange M, Plumbley MD (2013) Detection and classification of acoustic scenes and events: an IEEE AASP challenge. In: 2013 IEEE workshop on applications of signal processing to audio and acoustics, pp 1–4. IEEE
8. Zhang H, McLoughlin I, Song Y (2015) Robust sound event recognition using convolutional neural networks. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 559–563. IEEE
9. LeCun Y, Bengio Y, Hinton G (2015). Deep Learn Nat 521(7553):436–444
10. Palaz D, Collobert R (2015) Analysis of cnn-based speech recognition system using raw speech as input (No. REP_WORK). Idiap
11. Adavanne, S., & Virtanen, T. (2017). Sound event detection using weakly labeled dataset with stacked convolutional and recurrent neural network. arXiv preprint axXiv:1701.02998
12. Adavanne S, Pertilä P, Virtanen T (2017) Sound event detection using spatial features and convolutional recurrent neural network. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 771–775. IEEE
13. Zaki HF, Shafait F, Mian A (2016) Modeling 2D appearance evolution for 3D object categorization. In: 2016 international conference on digital image computing: techniques and applications (DICTA), pp 1–8. IEEE
14. Piczak KJ (2015) Environmental sound classification with convolutional neural networks. In: 2015 IEEE 25th international workshop on machine learning for signal processing (MLSP), pp 1–6. IEEE
15. Meyer, M., Cavigelli, L., & Thiele, L. (2017). Efficient convolutional neural network for audio event detection. arXiv preprint axXiv:1709.09888
16. Pons J, Serra X (2019) Randomly weighted cnns for (music) audio classification. In: ICASSP 2019–2019 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 336–340. IEEE
17. Dai W, Dai C, Qu S, Li J, Das S (2017) Very deep convolutional neural networks for raw waveforms. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 421–425. IEEE
18. Salamon J, Bello JP (2017) Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Process Lett 24(3):279–283
19. Boddapati V, Petef A, Rasmusson J, Lundberg L (2017) Classifying environmental sounds using image recognition networks. Proc Comput Sci 112:2048–2056
20. Abdoli S, Cardinal P, Koerich AL (2019) End-to-end environmental sound classification using a 1D convolutional neural network. Expert Syst Appl 136:252–263
21. Su Y, Zhang K, Wang J, Madani K (2019) Environment sound classification using a two-stream CNN based on decision-level fusion. Sensors 19(7):1733
22. Sharma, J., Granmo, O. C., & Goodwin, M. (2019). Environment sound classification using multiple feature channels and attention based deep convolutional neural network. arXiv preprint axXiv:1908.11219
23. Virtanen T, Plumbley MD, Ellis D (eds) (2018) Computational analysis of sound scenes and events. Springer, Heidelberg, pp 3–12
24. Sahidullah M, Saha G (2012) Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition. Speech Commun 54(4):543–565

25. Shepard RN (1964) Circularity in judgments of relative pitch. J Acoust Soc Am 36(12):2346–2353
26. Paulus J, Müller M, Klapuri A (2010) State of the art report: audio-based music structure analysis. In: Ismir, pp 625–636
27. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR workshop and conference proceedings, pp 315–323
28. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint axXiv:1207.0580
29. Jung SH, Chung YJ (2020) Performance analysis of the convolutional recurrent neural network on acoustic event detection. Bull Electr Eng and Info 9(4):1387–1393
30. Lezhenin I, Bogach N, Pyshkin E (2019) Urban sound classification using long short-term memory neural network. In: 2019 federated conference on computer science and information systems (FedCSIS), pp 57–60. IEEE
31. Salamon J, Jacoby C, Bello JP (2014) A dataset and taxonomy for urban sound research. In: Proceedings of the 22nd ACM international conference on Multimedia, pp 1041–1044
32. Tokozume, Y, Harada T (2017) Learning environmental sounds with end-to-end convolutional neural network. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 2721–2725). IEEE