석사학위논문

딥러닝 기반 환경음 분류

Deep Learning Based Environmental Sound Classification

국민대학교 일반대학원

전자공학부

Shantanu Sen Gupta

2020

Deep Learning Based Environmental Sound Classification

by

Shantanu Sen Gupta

A thesis submitted to the Department of Electronics Engineering, Graduate School, Kookmin University in partial fulfillment of the requirements for the degree of Master of Science

Supervised by

Professor Dr. Ki-Doo Kim

2020 년 12 월

Graduate School, Kookmin University

Department of Electronics Engineering

2020

Deep Learning Based Environmental Sound Classification

A thesis submitted in partial fulfillment of the requirements for the

degree of Master of Science

by

Shantanu Sen Gupta

January, 2021

This is certified that it is fully adequate in scope and quality as a thesis for the degree of Master of Science

Approved by

Professor Dr. Young Il Park (Chair, Thesis committee)

Professor Dr. Jung Hun Oh (Member, Thesis committee)

Professor Dr. Ki-Doo Kim (Thesis supervisor & member, Thesis committee)

Graduate School, Kookmin University

Department of Electronics Engineering

2020

Shantanu Sen Gupta 의

석사학위 청구논문을 인준함

2021 년 02 월

심 사 위 원 장　박 영 일 ㊞

심 사 위 원　김 기 두 ㊞

심 사 위 원　오 정 헌 ㊞

국민대학교 일반대학원

Dedicated
to
My Family

Acknowledgement

Most importantly, I am appreciative and grateful to Almighty God, for allowing me the chance to finish my master's study. Simultaneously, I might want to render my enormous thankfulness to my supervisor, Professor Dr. Ki-Doo Kim, for his educational direction, continuous help, and motivation all through my whole master's study. I might want to offer my thanks to my all course instructors from whom I learn numerous new remarkable speculations and technological knowledge. My profound heartiest appreciation is for the thesis committee members who have contributed their time and significant information for improving this thesis in a favorable way. I would like to communicate my profound hearted regard and appreciation to Prof. Dr. Young Il Park and Prof. Dr. Jung Hun Oh for their profound management and evaluation of my dissertation. From that point forward, Kookmin University likewise merits of this affirmation. I am likewise appreciative to Korean Government for giving me the valuable occasion to study here in Republic of Korea. I might want to thank my friend Shifat Hossain and my senior brothers Rappy Saha and Partha Pratim Banik, for their motivation, conversation, examination, and experience that direct me to improve objective of my exploration works.

It is a significant privilege and consistently prestigious for me to be a member from the Multimedia Communication and Signal Processing Lab. On account of my dearest lab mates Tae-Ho Kwon and Chowdhury Azimul Haque, who are continually enlightening my motivation to guide me into my objective. It is actually an extraordinary opportunity for me to offer my plentiful thanks to Dr. Mostafa Zaman Chowdhury for his honorable help. I need to give my confession to the entirety of my Bangladeshi companions, youngsters, and senior siblings living in Korea, for their unlimited consideration and backing during my stay in Korea.

And last but not the least; I am indebted to my dearest parents, Mr. Nandan Sen Gupta and Mrs. Bandita Sen Gupta, and my younger brother, Atanu Sen Gupta for making me fixated on my goals, and also for their continuous support, enormous inspiration, and contribution in every aspect of my life.

Table of Contents

Table of Figures

List of Tables

Acronyms

Abbreviation	Full Form
ESC	Environmental Sound Classification
GTS	Gammatone Spectrogram
MFCC	Mel-frequency Cepstral Coefficient
CNN	Convolutional Neural Network
1D	One Dimensional
2D	Two Dimensional
IoT	Internet of Things
US	Urban Sound

Abstract

Deep Learning Based Environmental Sound Classification

by Shantanu Sen Gupta

Department of Electronics Engineering,

Kookmin University, Seoul, Korea.

Environmental sound (ES) consists of the surrounding sounds around us. The classification of this kind of sound has specific and significant contributions in many of the modern world applications specially in internet of things (IoT). But accurate prediction of the sound class of ES is really hard because of its uncertain pattern at maximum case. Even though so many studies had been carried out in past to develop a fully accurate environmental sound classification (ESC) system by making decision on different extracted features, but feature selection and classifier design is really a harsh job and also sometimes does not guarantee precise result. In recent years, ESC has attracted the attention of research community because of the development in the field of learning algorithms. Following this trend, a study has been done here to find and propose two CNN models (1D and 2D) for both raw signal input and for Gammatone spectrogram input. Comparing with 2D CNN model, 1D CNN for time series waveform is a recent idea. Assessing the models on two various datasets (ESC-10 and US-8K), the overall accuracy for 2D CNN is found 80.2% (ESC-10), 89% (US-8K). For 1D CNN obtained accuracy is 80.4% % (ESC-10), 86% (US-8K). The proposed 1D CNN performs almost similar to the proposed 2D CNN model. To achieve this high accuracy and introduce the models to more real world data, effective data augmentation procedure is done. Finally, the suggested models have very limited number of parameters to be trained and floating point operations per second (FLOPS) are also very limited, especially for 1D CNN.

Keywords: Environmental sound classification, CNN, Gammatone, Raw audio

Chapter 1

Introduction

1.1 Introduction

In recent years, there are so many studies have been done targeting music and speech signal processing. These researches include music tagging or genre classification, music information retrieval analysis of rhythm, harmonic analysis, and other low-level or high-level analysis. In case of speech signal, emphasis has also been given on speaker identification, speech-to-text conversion or vice-versa, automatic speech recognition etc. In contrast, less research has been done on environmental sound classification (ESC). The classification or tagging of different surrounding sound is referred to ESC. Actually, ESC is a kind of subset of environmental sound processing. The other two branches of environmental sound other than ESC are: acoustic scene classification and acoustic event detection [1]. Acoustic scene classification gives attentions to classify a sound recording into a single scene tag like “indoor”, “outdoor”, “home” etc. Acoustic event detection targets to predict the start and finish point of a single sound label from a whole audio. ESC or environmental sound tagging works to detect a single sound source label from the input audio. Besides speech and music sounds, environmental sounds also carry a lot of information, specially about our surroundings. This class of sound signals have a great contribution in evolving urban acoustic monitoring device, intelligent audio based surveillance system, environmental context aware processing, automatic crime scene investigation and many others. In addition to surveillance and security system development, ESC can be used to other daily life purposes also like large multimedia catalogues retrieval and indexing, context aware portable device, developing a robot to proficiently interact with the environment, audio-visual safety equipment development and many more. Overall, ESC can play a significant impactful role in this edge of internet of things (IOT).

Previously, the task of ESC was accomplished by different hand crafted features. Those features include several spectro-temporal features like log scale spectrogram. Some other popular methods like non-negative matrix factorization (NNMF), Gaussian mixture model

(GMM), Gabor filter bank (GFB), singular value decomposition, mel-frequency cepstral coefficient (MFCC) were also used as feature extraction methods. As a classifier, some statistical machine learning (ML) classifier like support vector machine (SVM), K-nearest neighbor (KNN), etc. were also used. These algorithms were mainly used in the speech domain. But for non-speech signal (i.e. environmental signals), not all the time these mid-level features work well. Moreover, some additional challenges also make the classification task difficult for the machine. The main obstacles, found in the way to develop an intelligent environment aware machine, can be listed as: (i) As like speech, environmental sound properties cannot be caught properly when it is thought that speech is made of some basic building blocks because some sounds in our atmospheres are so random and pattern less [2]. (ii) In case of music there are some stationary features like melody and rhythm which are meaningful in nature but for environmental sounds these are unavailable. (iii) the position of microphone has also an impact in this context. When the placement of sound recording device is far from source, low signal to noise ratio (SNR) as well as echoes and reverberations can be happened [3]. (iv) Rather than other sound sources environmental sounds' various property like spectral content, volume and duration may be different in terms of background noise [3]. (v) The total number of annotated audio samples per class are so limited for ESC, compared to other audio signals' datasets. So, the classifier cannot get proper number of training data to learn [1]. Also, since most sounds have random temporal and spectral patterns, it is very impossible to make augmented dataset or create artificial data.

Previously, convolutional neural networks (CNNs) were gaining popularity in numerous problem areas. CNN architecture was first developed to tackle an image domain problem called handwritten architecture [4]. Later, in image classification, image segmentation, object detection, medical image analysis, CNN tackles most of the problems successfully. Now-a-days, in our day to day life from healthcare system like drug discovery to diverse recommender system everywhere CNN based artificial intelligence (AI) are used. Likewise, advanced computational power like the development of graphics processing unit (GPU) or tensor processing unit (TPU) have fast-tracked the AI by giving more processing speed and space.

However, in case of audio processing, use of CNN is comparatively a modern trend. One early work for speech recognition is found at [5]. Recently CNN based ESC system are more focused. Although CNN is some kind of similar to our brain architecture, proper selection and arrangement of layers inside the model is quite necessary to learn the deep features well.

1.2 Human ear architecture

In order to develop an effective environmental sound classifier, solid understanding of how human's auditory system works and the basic principle of the sound recognition process of human is obligatory. So, these two core topics are broadly discussed here.

The two ears of a person work kindly similar and creates symmetry to create a pleasant sound for us. Keeping that in mind, the function of only one ear is discussed here [6]. The auditory system of a human can be broadly divided into three sections as depicted in Fig. 1.1: outer ear, middle ear, and inner ear. Brief description of these three sections are given at this point.

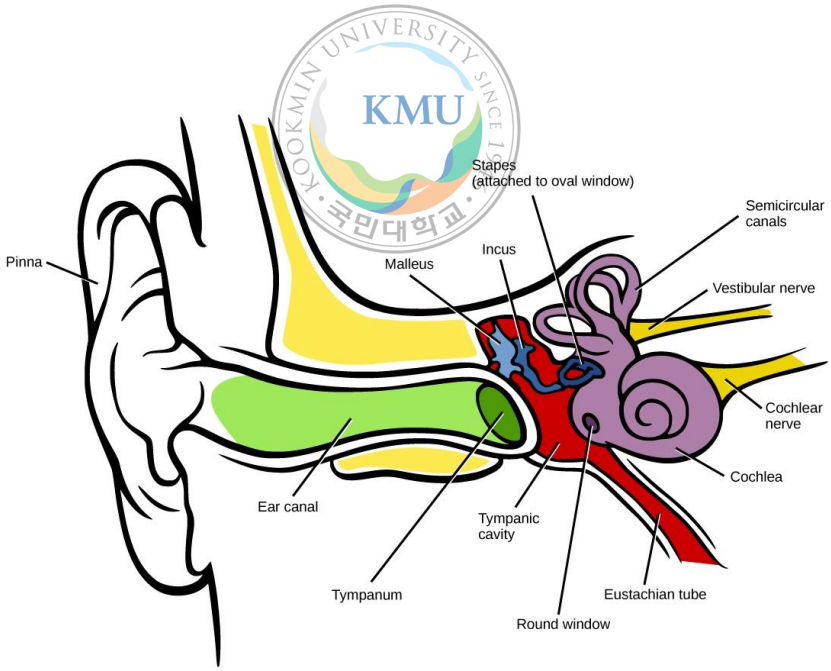

Fig. 1.1. The structure of the human peripheral auditory system [6].

Outer ear: This segment consists of pinna, concha, and ear canal. The outer most part of our ear is pinna. It mainly does filtering to the incoming sounds which mainly depends on the sound source direction. After that the concha leads the filtered sound to the ear canal. Ear canal is a one end opened and 2.5 cm long tube. The ear canal can be thought as a band pass filter as it is mostly sensitive to the frequency range 1000~6000 Hz.

Middle ear: Where the ear canal ends, the middle ear starts from there called ear drum. Ear drum is a sensitive membrane that responds to the pressure change at the end of the ear canal. The middle ear is full of air. To protect the ear drum from excessive pressure, equalizing the internal and external pressure is important. This responsibility is performed by Eustachian tube which connected to the back of the throat. If the sound from ear drum is directly transmitted to the water filled cochlea, most of the sound will be reflected back due to the impedance mismatching. This problem is solved by three tiny bones: malleus, incus, and stapes (these altogether also called ossicles). The ossicles convert the weak vibration at eardrum into a stronger one by concentrating the sound at the eardrum into a smaller area so that the pressure increased roughly 20 times at cochlea. So, the middle air can be treated as an impedance matching transformer as well as the ossicles perform as a high pass filter.

Inner ear: The two main parts of inner ear are cochlea and auditory nerve. Cochlea is filled with fluid (mostly water dissolved with different biologically important chemicals). Transduction which means converting the acoustic vibration into electrical neural activity occurs here. The cochlea is spiral shaped having about 3.5 cm length and has about 2 mm diameter (least at the apex and greatest at the base). For better view, a cross-section is displayed at Fig. 1.2. From this figure, it can be noticed that the tube is divided into two parts: reissner's membrane and basilar membrane. The vibration in stapes leads basilar membrane to vibrate when sound enters through the oval window. The basilar membrane is stiffer in base part than in the apex part where it is wider. So, this organ's different portions are sensitive to different range of frequencies like high frequency at the base part and low frequency at the apex part. In this sense, the basilar membrane can be treated as a bank of overlapped band-pass filters also known as auditory filters. Fig. 1.3 and 1.4 demonstrates the overlapped auditory filters and the decomposition of a complex signal in basilar membrane respectively. However, after this step, the vibration in membrane is detected by the inner hair cells and this cells cause electrical spikes in auditory nerves. The task of identifying and separating the signals is performed by brain.

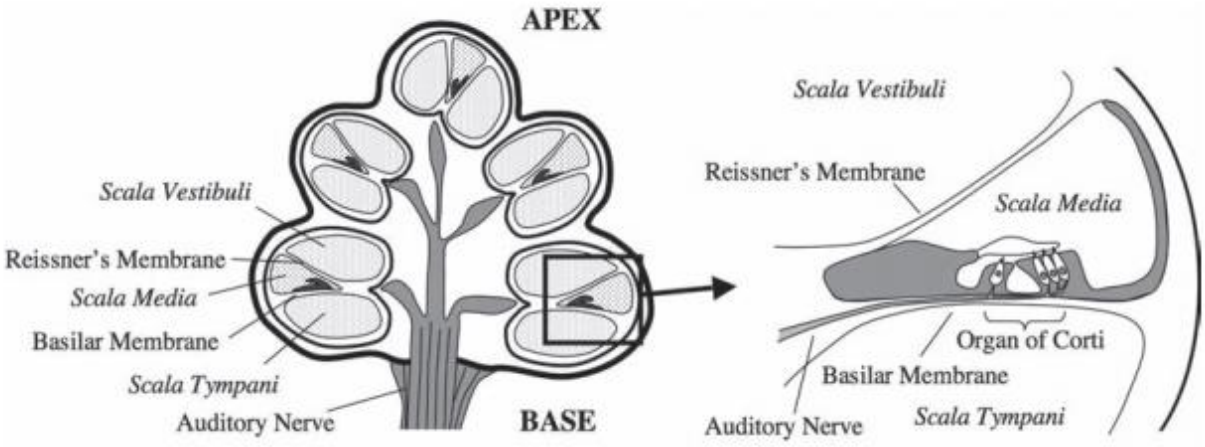

Fig. 1.2. Two enlarged side view of cochlea (cross-section) [6].

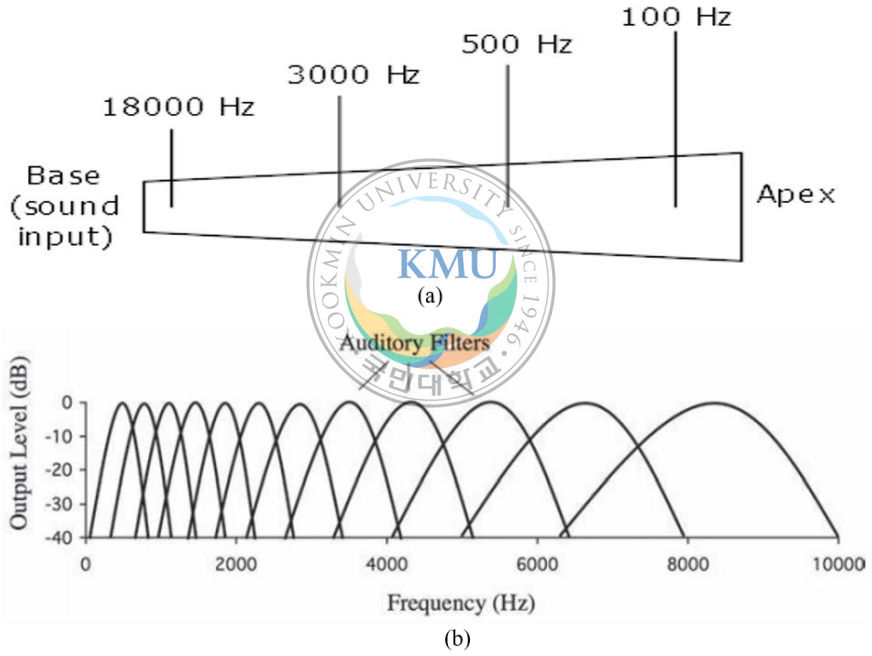

Fig. 1.3. In case of basilar membrane: (a) frequency response at different position, and (b) relative attenuation of each auditory filter as basilar membrane is a continuous array of filters. The right side of (b) indicates response of the left side of (a) and vice versa [6].

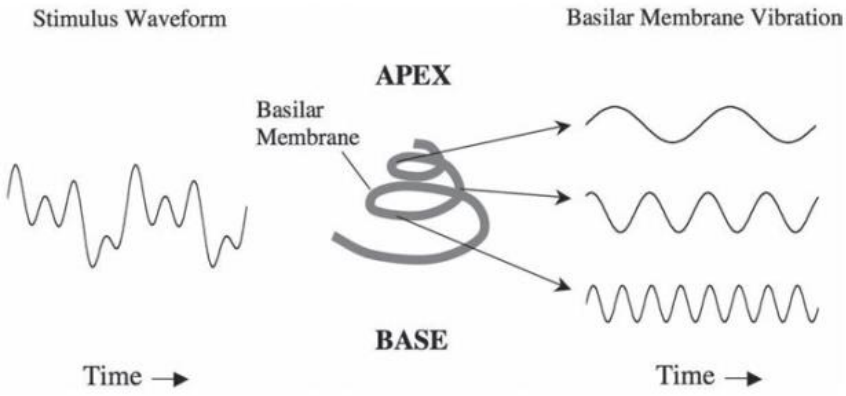

Fig. 1.4. The illustration how a complex sound is decomposed by the basilar membrane at different position [6].

auditory perception for speech processing is described and how CNN architecture can be considered as like human cortical organization is also analyzed.

In general, speech can be divided into two parts based on the relative context: (a) non-sensible, and (b) sensible. The probability of recognition of first one correctly is called articulation. And for the later one it is termed as intelligibility. At [7], a hypothetical model for recognition steps in human brain is explored. That hierarchical model can be pictured as Fig. 1.5. This figure pictorially describes about the process from an input signal to word intelligibility through articulation process. When a speech signal $s(t)$ arrives at cochlea, it is decomposed by a series of band pass filters (critical bands). If the input signal is gained by α then after first layer the phone features are represented by the articulation error e_k is given by (1.1) and (1.2); where k denotes the frequency channel. In layer II, the articulation s (phone space) is measured with the help of (1.3) and (1.4). After layer III, the calculated articulation from the previous layer is transformed into syllables by (1.5). At the last layer, using (1.6) the syllables are then captured as word intelligibility.

$$D_k(\alpha) = \frac{1}{K} SNR_k(\alpha)/30 \tag{1.1}$$

$$e_k = e^{D_k} \tag{1.2}$$

$$A(\alpha) = \sum_{k=1}^{K} D_k(\alpha) \tag{1.3}$$

$$s(A) = 1 - e_{min}^{A} \tag{1.4}$$

$$S(A) = s^3 \tag{1.5}$$

$$W(A) = 1 - (1 - S(A))^{j>1} \tag{1.6}$$

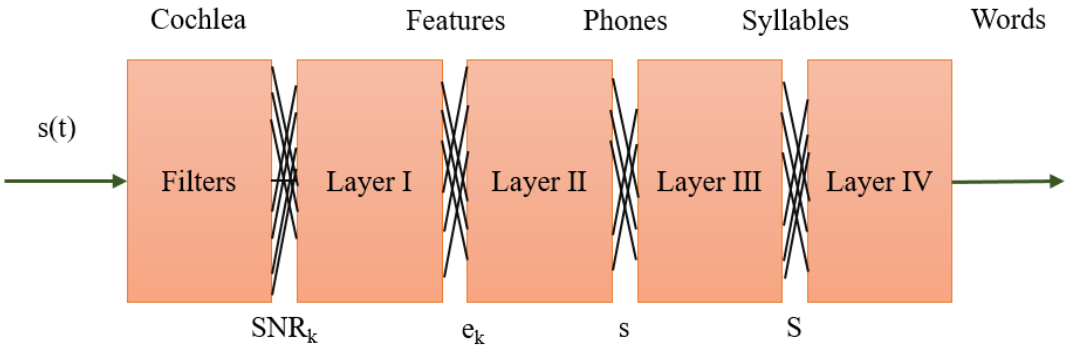

Fig. 1.5. Speech recognition layers as hypothetical cascade connection.

Although human can successfully interpret the speech signal very fast, one of the main limitations mentioned in that research [7] is that the ambiguity of existence any feedback connection between the upper, mid and deeper layer. Keeping that in mind, CNN models are used here at two stages. One model assumes that the input signal is in the raw form i.e. as before cochlea without any preprocessing. The other model undertakes that the signal is preprocessed i.e. after cochlea filter bank. In the next paragraph, the contingency of CNN architecture and the usage in case of human auditory system will be made clear.

Whether the CNN architecture can replicate the human auditory cortex was analyzed at [8]. A dual task (word recognition and music genre classification) achiever CNN network was designed in that experiment. After doing functional magnetic resonance (fMRI) of human brain and analyzing the last feature extraction layer of CNN for same type of sound it was seen that the fMRI pattern is almost same. Other performance indicator parameters like correlation coefficient between human observation and network output also support the previous statement. The developed CNN architecture also performs better than the other handcrafted feature based method and other spectro-temporal filter bank input. The cochleagram input is also a time-

frequency representation but rather than other cochlear model its' frequency resolution is more close to the real cochlear architecture. However, the hierarchical CNN architecture here somehow backs the theoretical speech recognition architecture at [7].

In the following description, we will find that quite a lot of input signal representations are possible for CNN model. Based on this kind of representation, one dimensional (1D) or two dimensional (2D) CNN architecture are possible where 1D CNN works on the raw data and 2D CNN works on preprocessed data. For representation of the input signal, Gammatone filter bank (GTFB) based Gammatone spectrogram (GTS) is used here because it imitates the human auditory perception more closely. In this study, two separate 1D and 2D CNN models are proposed for classifying the input sound from raw wave and from mid-level representation correspondingly. Furthermore, the target is to obtain same classification accuracy with respect to computational cost and number of parameters. As the number of supporting data per class is too few, effective data augmentation methods are applied keeping real world scenario in mind. The proposed two models perform almost similar and obtain accuracy close to the other state-of-the art methods.

Chapter 2

Related Works

According to input signal preprocessing, modelling, network architecture and classification methods, the ESC system can be divided into so many categories. However, the related works are analyzed here by dividing those into two big categories. This section leads to better understanding the sequence of developing ESC system from traditional methods to currently most popular deep learning methods. Therefore, along with classification methods, ESC system can be divided into two groups: 1) Machine learning based ESC, and 2) Deep learning based ESC.

2.1 Machine learning based ESC

Environmental sound classification is a new research topic compared to the other fields in sound signal processing. So, ESC using classical statistical theory was not so common. Rather some related topics can be found in this field which used classical machine learning or shallow learning network for this purpose. Targeting acoustic scene classification (ASC), Bisot et. al. [9] did a study on the effectiveness of various matrix factorization methods which are used as feature set. As time-frequency representation, constant Q-transform (CQT) was used. After that they applied matrix factorization approach. After this step, created data matrix was given input to the logistic regression classifier. For comparison, non-negative matrix factorization, sparse matrix factorization, convolutive non-negative matrix factorization, and kernel-based matrix factorization were used. As well as, they also developed deep neural network (DNN) for two different kind of inputs. Their final finding was that the matrix factorization based feature learning performs as good as DNN and sometimes better. To recognize environmental sounds, mel-frequency cepstral coefficients (MFCC) are of great use. In [10] Chu et. al. tried to develop time-domain related features which can either be a substitute of MFCC or can be jointly improve recognition accuracy with MFCC. For this purpose, they suggested matching pursuit (MP) based feature learning. A vast empirical analysis was done to find the most suitable feature set for audio sound categorization. With MP features, others commonly used features

like MFCC, linear prediction cepstral coefficients (LPCC), and their respective delta were also used. Among these huge set of feature dictionary, different feature combination (subset) were used. The result shows MP features as an excellent candidate itself while other frequency-domain features may fail sometimes. Dhanalakshmi et. al. [11] used a shallow autoassociative artificial neural network (AANN) and Gaussian mixture model (GMM) for classifying an audio into several categories. For feature input they used linear prediction coefficients (LPC), linear prediction cepstral coefficients (LPCC) and MFCC. Upon the test dataset, they established that MFCC input works superior in both cases and overall AANN performs best. In [12], Geiger et. al. evaluated the performance of gabor filterbank (GF) features alongside with MFCC features. They developed an effective audio surveillance system employing audio classifier. It was concluded in their research that for noisy conditions, specially for static sound, GF features perform better than MFCC features. On the other hand, in case of clean conditions, MFCC performs superior. Wang et. al. [13] proposed a home automation system based on ESC. In their methodology they added a noisy signal enhancement module. After this enhancement, the signal is classified by a frame based SVM classifier as well as file based classifier. For each case of classifier, two types of input sets were tested. MFCC performed better than independent component analysis (ICA) transformed MFCC feature set for clean and noisy case. But when noisy signals were enhanced then ICA transformed MFCC worked better. Wang et al. [14] proposed another noise suppression model for environmental sound recognition (ESR) based on wavelet transform. In that research, they showed that wavelet based noise suppressed features work better than MFCC feature based classification system. In the noise suppression module, by decomposing the signal into wavelet-packet further preprocessing was done. The output from this module was a noise suppressed enhanced signal. From this clear output, wavelet subspace based features were extracted. In the classifier module, probability product-kernel based SVM was used. Another matching pursuit (MP) based ESC method was studied by Ghoraani et. al. [15]. They targeted seven time-frequency matrix based features which were extracted by some prior steps. First of all, using MP time-frequency distribution, a time-frequency matrix (TFM) was proposed. After that, the TFM was decomposed with the help of non-negative matrix decomposition algorithm. At the last step, seven significant features were

selected and given input to the linear discriminant analysis (LDA) classifier. In their result they showed a significant improvement over the system where MFCC features are used. Akbal et. al. [16] stated a textural and statistical feature extraction method for environmental sound classification which is something different from others. In the several stage of that research, at first stage one dimensional local binary pattern (1D-LBP), one dimensional ternary pattern (1D-TP) were extracted as textural feature. As statistical feature a nineteen different moments were selected. By this method, a huge number of features have been collected for overlapping windows across the sound. So, to reduce the computational load neighborhood component analysis (NCA) was used to select to find the most prominent features. At the last stage, support vector machine (SVM) classifier was used to find the appropriate class of the input sound. Chandrakala et. al. [17] analyzed another way for representing the sound in order to sound event recognition task and Environmental audio scene recognition (EASR) task. They proposed an instance-specific adapted Gaussian mixture-models (ISAGMMs) for EASR task and found another model for SER task called instance-specific hidden markov-models (ISHMMs). Although, the number of dimensions, parameters, and required training data are low, the accuracy for the proposed system was not so high. To simulate the human auditory system, Valero et. al. [18] adopted Gammatone cepstral coefficients (GTCC) as input features for classifying sounds. They empirically compared the GTCC features against MFCC features for four different classifiers: decision tree (DT), SVM, k-nearst neighbor (KNN), and ANN. The outcome of that research is that in maximum cases classification accuracy for GTCC improved.

So, after reviewing the researches targeting ESC using the basic machine learning models and superficial networks, it can be resolved that these systems do not always perform superior or as expected. Moreover, sometimes it is difficult to catch the proper representation of input sound i.e. feature set. So, it is necessary to find a model that can mimic the human brain of sound perception while the performance is tremendous for with or without pre-processed sound.

2.2 Deep learning based ESC

Deep learning based ESC system is comparatively newly focused area than others. As deep learning network, convolutional neural network (CNN) architecture is mainly used. Besides

this, recurrent neural network is also a great choice. In this field, inputs are given in two formats, either time-frequency representation (which is also termed as intermediate feature) or only time domain features like amplitude value. Therefore, based on input feature selection, the deep learning based networks are called either 1D or 2D network.

Piczak et. al. [19] proposed a shallow CNN network, which takes input into two types (2 channels): log-mel spectrogram and the respective delta. They also applied data augmentation on the training dataset. Their result was improved than mel frequency cepstral coefficient (MFCC) which was input at their baseline model. Pons et. al. [20] studied the different CNN networks' front end (which is responsible for extracting features) to compare the output based on support vector machine (SVM) and extreme learning machine (ELM). They tried several 1D CNN feature extraction layer: sample-level, frame-level-many-shapes, and frame-level. For 2D CNN, their choices were: 7×96, 7×86, VGG, timbral, temporal, timbral+temporal feature extraction front-end on spectrogram. Beside these features, MFCC was used as input features for baseline model. This technique was both applied for music classification and sound classification. In this study, (timbral + temporal) and VGG front-ends achieved remarkable result. As there is insufficiency of ESC dataset, data augmentation can be a good option to increase the number of supporting examples per class. Some data augmentation techniques were applied by Salamon et. al. [21]. Before giving input to their proposed deep CNN model as log-mel spectrogram, each of the signal was augmented by four ways. The first augmentation was time-stretching where four factors were used: {1.23, 1.07, 0.93, 0.81}. The second type of augmentation was pitch-shifting by also four factors: {2, 1, -1, -2}. The third way is to compress the dynamic range of sound by four standards: music, film speech, and radio standard. The fourth and the last way to augment the dataset was adding four kinds of background noises: {park, street-traffic, street-workers, street-people}. As in their experiment they found that the pitch shifting is quite effective, they later included other four factors pitch shifting: {3.5, 2.5, -2.5, -3.5}. Their finding was that deep CNN model with data augmentation performs better than both deep CNN model without augmentation and shallow CNN model with augmentation. In this research, the idea of pitch shifting and time stretching is incorporated. In [22], Hoshen

et. al. tried raw speech waveform along with CNN. Their purpose was to check if it is possible for CNN filters to extract auditory filter-bank like features from the input signal. The network took input a two channel raw waveform and showed that it can learn a bank of band-pass beamformers like feature representation. The network's main task was to automatic speech recognition but however it performs slightly worse than their baseline model. Their baseline model input was log-mel spectrogram. One of the prior and successful work in the field of ESC using 1D CNN is [23]. Their 1D CNN took input 32,000 data sample (full 4s audio at 8KHz) rate. They examined five deep CNN networks named M3, M5, M11, M18, M34-res (mimics ResNet [24] architecture of image recognition). Here, 3/5/18/34 means the number of layers in network. The authors at [23] found that, M18 performs better than other architectures. They also stated that large receptive fields in the first layer of CNN can be considered as band- pass filters. The proposed 1D and 2D CNN architecture in this research is greatly influenced by M34-res architecture. But the suggested networks consist of much less layer than M34-res. So, number of parameters have also been decreased. Moreover, the residual layer is also opted-out in the proposed networks. Another end-to-end ESC system was proposed by Tokozume et. al. [25]. They proposed a convolutional neural network for raw waveform input. They compared their model with log-mel feature based model and found that their proposed system performed better than later. Further improvement was done on their model by both combining the raw waveform and log-mel features. Another two important findings in their model were that they experimentally found that if the input audio length is between 1s to 2.5s the model performs almost similar but when before or beyond this length model accuracy goes down. Similarly, they also analyzed the filter size and showed that for 1D convolution layer large filter size improves accuracy and however for 2D and 3D convolution layer small filter size performs better. Following this research [25], Tokozume et. al. published another improved CNN architecture at [26]. This time their proposed system was named as EnvNet-v2. For training their network, they incorporated between class (BC) learning. The proposed model was trained using mixed sounds from different class besides the original training data and gave output the mixing ratio. In one way, the BC learning works as a data augmentation procedure. In another way, BC learning helps the model to learn the discriminative features between different classes.

This is really helpful when there is so many classes. They experimental result showed that the proposed procedure can successfully minimize the error rate on different datasets. They also analyzed how in the feature space BC learning helps to enlarge of Fisher's criterion. Another recent research using 1D CNN for ESC was done by Abdoli et. al. [27]. In their paper, they suggested a few-layer CNN architecture (around 5 convolution layers). This kind of architecture also requires less parameters to train. They also tested their method against different input size and found that 1s input at 16 kHz sampling rate performed best. Among the two input methods, their model performs better for Gammatone input type. At [28], Park et. al. proposed another 1D CNN model for ESC task. Their proposed model named as 1D only because 1D raw input. But after the input layer 2D layers were used. Learnable Gammatone filter bank (LGTFB) as well as equal-loudness normalization were incorporated in their study. However, the accuracy improvement was not so much. Bavu et. al. [29] proposed TimeScaleNet which consists of two parts: BiquadNet and FrameNet. The first one works on sample level data and the later one works on frame level data. In BiquadNet module, RNN layers were used by considering them as infinite impulse response (IIR) filter and 1D CNN layers were used at FrameNet module as finite impulse response filter (FIR). In case of speech recognition, they achieve very high accuracy by this method. But for ESC dataset, their method performs average. In [30], Aytar et. al. described a very different method for natural sound recognition. A huge unlabeled video dataset was used for their CNN training purpose. The visual recognition task was done by previous state-of-the art knowledge or transfer learning procedure. For sound recognition task they designed very deep CNN model. This whole model can be considered as teacher -student learning scheme where the visual recognition module performs as a teacher network. In this manner, they achieved high accuracy on acoustic scene classification (ASC) and also for environmental sound classification dataset. Another approach based on 2D input representation is [31] by Khamparia et. al. They proposed two CNN models where the first model used only simple sequential CNN and the second one is tensor deep stacking network (TDSN) where each layer was divided into two sub layer. The input to these network was normal spectrogram. Nevertheless, their accuracy was not up to the mark.

In order to classify environmental sound more accurately some researchers also include ensemble method. In [32], Su et. al. proposed an aggregated 2D CNN model for obtaining high accuracy in this task. They used two separate 4 -layer CNN models to classify the given sound. At the decision level fusion, Dempster–Shafer (DS) evidence theory was used. This two stream CNN includes overlapping feature sets: LM-CST (spectral contrast, chroma, log-mel spectrogram, tonnetz), and MFCC-CST (tonnetz, MFCC, spectral contrast, chroma). Another similar method is [33] by Li et. al. Their proposal was to use two separate networks to recognize environmental sound successfully. One network was designed for classifying sound based on raw waveform input and the other network receives mel-spectrogram. Their final decision was also fused by DS theory. Another interesting research that considered the sound representation at frequency domain as image is [34]. In that article, Boddapati et al. [34] used famous image recognition networks like AlexNet [35] and GoogleNet [36]. They gave input to these networks three basic representation of sound as three image channels (like R, G, B). For three channels, spectrogram, MFCC, and cross recurrent plot (CRP) were considered. Their observation was that these famous CNN image recognition networks can also classify sound successfully. Beside these techniques, recurrent neural network (RNN) are also used for learning the signal's temporal feature. Zhang et. al. [37] developed a convolutional recurrent neural network (CRNN) model where convolution layers were used alongside with recurrent layer (gated recurrent unit (GRU)). As input features log-scaled Gammatone spectrogram and its delta were used. Alternative interesting approach was taken by Salamon et. al. [38]. In that study, unsupervised feature learning method along with spherical k-means algorithm was used for urban sound classification. The three main steps involved in their research was: preprocessing, feature learning, and classification. The random forest (RF) classifier was use to classify the sounds.

In the recent years, 1D CNN architecture is gaining popularity because it does not require any preprocessing step. Although yet now it is less accurate than 2D architecture in the sense of performance, continuous work on this field will improve the quality hopefully. After carefully studying the previous related researches, we can enlist our focuses and contributions in this thesis work as follows:

- Develop a one dimensional (1D) CNN network which can perform traditional 2D CNN.
- Get free from input pre-processing.
- Implement effective data augmentation procedure to enlarge the dataset size and to deal with real world problems.
- Modest number of trainable parameters and low FLOPS.

Chapter 3

Proposed Classification Method

The suggested environmental sound classification system in this piece of work is divided into several steps. The overall system architecture is pictured in Fig. 3.1. Fig. 3.1 demonstrates the classification strategy for raw audio input as well as for Gammatone spectrogram input. In this work, two separate networks are proposed. Among this two networks, one is 1D CNN model which works on raw audio input and other is 2D CNN model which takes Gammatone spectrogram as input. The classification steps can be nicely described by the following points:

3.1 Dataset collection

If the model performs better on more than one dataset, the model can be considered as a generalized model. Therefore, to improve the generalization performance, the proposed CNN models are supposed to be trained on different diverse dataset. In order to achieve that, data from [39] and [40] has been collected. These datasets are known as ESC-10 and US-8K respectively. The details description of these datasets can be found in chapter 4.

3.2 Data preprocessing

Data preprocessing step consists of several other sub-steps. They are: (a) Data cleaning, (b) Input preprocessing.

3.2.1 Data cleaning:

In this step, the collected datasets are checked to make it free from any kind of not-a-number (NaN) value and from other outliers also. Moreover, the target labels are also encoded with value between 0 and (number of classes) – 1.

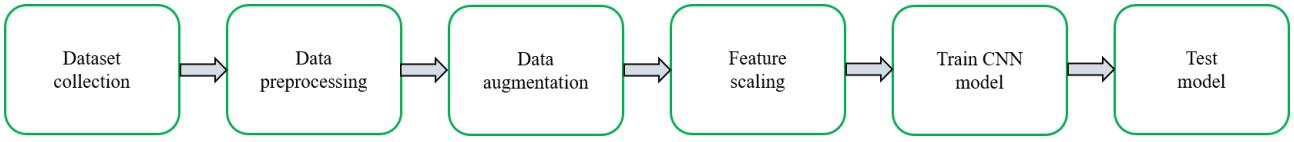

Fig. 3.1 Process diagram of proposed ESC system.

3.2.2 Input preprocessing

3.2.2.1 Signal framing

Each input sound signal is sampled at 22.5 kHz. After that, the signal is framed at 1 sec period i.e. 22,500 samples per second. The following frames are overlapped by around 35% samples (hop length 8000). In this way, not only the number of data per class increases but also it can be considered as data augmentation. The framing process can be viewed as Fig. 3.2. This 1s frame is then given input to further preprocessing and to the CNN networks.

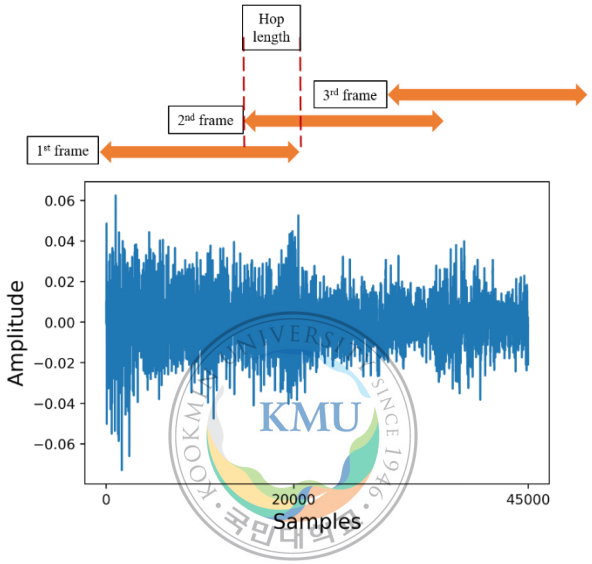

Fig. 3.2. Input signal re-sampling and framing.

3.2.2.2 Gammatone spectrogram

Spectrogram basically helps us to visualize the variation of frequency over time. Moreover, it also expresses the distinction of different energy levels for different frequencies over the time. So, in a spectrogram time is plotted along the X-axis and frequency is plotted along the other axis. The energy is indicated by color bar which can be considered as a third axis. Fig. 3.3(a) shows a spectrogram of a particular audio in linear scale, and Fig. 3.3(b) shows the same in log scale. However, human being's sound perception is more similar to log-scale spectrogram.

Let's assume a signal X with samples N. If the signal is framed at m samples per frame, then X becomes $X \in R^{m\times(N-m+1)}$. Now, the discrete Fourier transform (DFT) representation of X is related to the original signal X by (3.1)

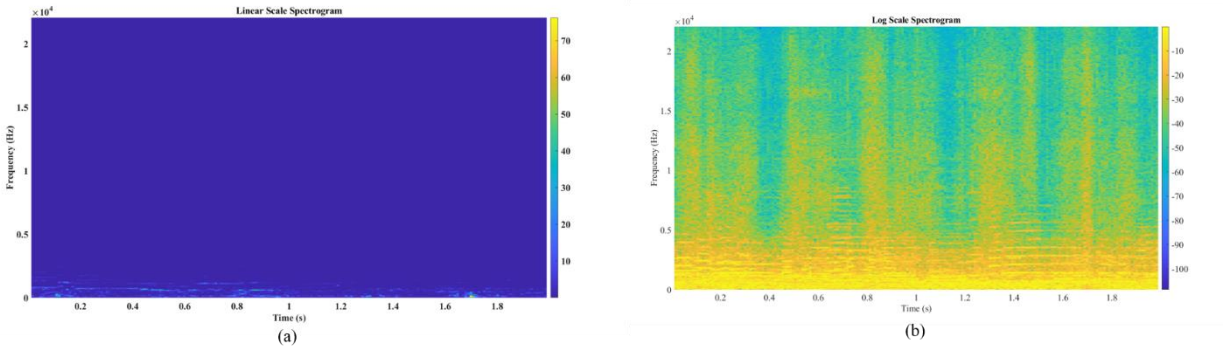

Fig. 3.3 Spectrogram of sound class "*street music*": (a) linear scale, and (b) log-scale.

$$\hat{X} = \bar{F}X \tag{3.1}$$

where, $\bar{F}$ is the complex conjugate of Fourier matrix. $\hat{X}$ is the spectrogram which the original signal's time frequency representation.

Besides calculating from time-domain signal using Fourier transform, spectrogram can also be obtained by passing a signal through a series of filter bank. The later technique mimics the human auditory system. So, it is expected that when CNN networks are given input in this format (spectrogram), they perform better. Among several human auditory system representations like mel-frequency spectrogram, Gammatone spectrogram (GTS), etc. the second one is chosen. The main motivation behind this choice is that Gammatone filter approximates the human auditory perception as asymmetrical filters whereas mel filter bank assumes to be triangular. It is argued that the Gammatone filter bank better mimics the basilar membrane of human ear [41].

Gammatone filterbank was first proposed by Patterson et. al. [42]. The name "Gammatone" came from the sinusoidal tone and gamma distribution's product. The Gammatone function can be written as (3.2)

$$g(t) = at^{n-1}e^{-2\pi bt}\cos(2\pi f_0 t + \phi) \tag{3.2}$$

where, a is the signal amplitude, f_0 represents the center frequency in Hz, ϕ is the phase of the carrier in radians, b is the filter's bandwidth (in Hz), and t is time in second. The human auditory filter shape and its' bandwidth is approximated by a rule called equivalent rectangular bandwidth (ERB) [43]. The conversion from f_0(in KHz) to ERB (in Hz) is given by (3.3)

$$ERB[f_0] = 24.7 \times (4.37 \times f_0 + 1) \tag{3.3}$$

The filterbank obtained from (3.3) is then multiplied by the spectrogram obtained from (1). Thus the FFT based spectrogram is reassigned as Gammatone based approximation. Fig. 3.4(a) shows the filter bank construction approximated by ERB scale. It is an overlapped band-pass filter bank which mimics the basilar membrane frequency analyzer at the inner ear of human. In this figure, the filterbank weights are normalized. Fig. 3.4(b) and 3.4(c) represents the Gammatone spectrogram of two sample sound class. To obtain Fig. 3.4, all the parameters are taken from [44] where summation window is 0.025 sec, hop length between successive windows is 0.010 sec, lowest frequency is 50Hz and highest frequency is the half of the sampling rate.

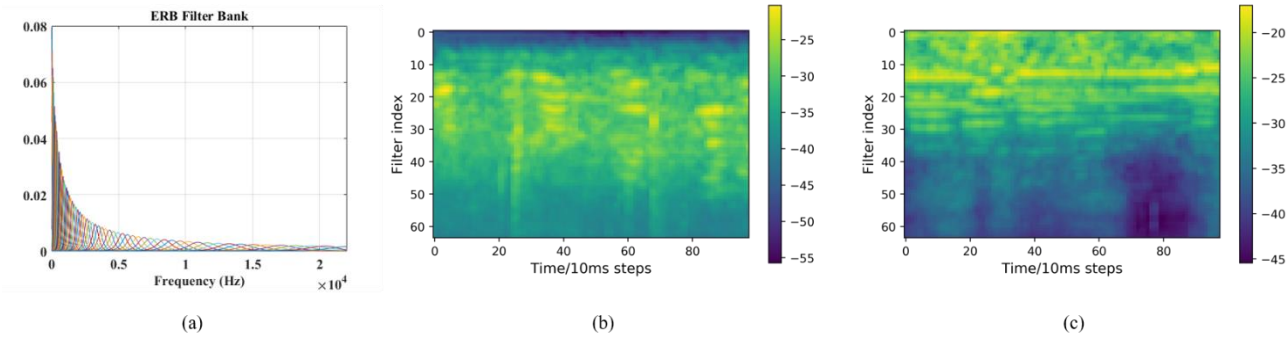

Fig. 3.4 Gammatone filterbank approximation: (a) filter bank using ERB scale, (b) Gammatone spectrogram of sound class "*children playing*", and (c) Gammatone spectrogram of sound class "*street music*".

3.3 Data augmentation

The main purposes of data augmentation are two. The number of dataset regarding ESC is not so high and therefore the number of supporting data per class is also insufficient. To fulfill

these requirements data augmentation is a necessary step. Besides this, augmentation also serves another purpose for efficient training, where the model is trained with more complex data and real world examples. In case of image recognition/object detection learning, various kinds of data augmentation are possible like image rotation, adding noise to image, color space changing and more like this. Similar to this, audio data augmentation has also been reported to improve accuracy by Salamon et. al. [21]. However, in this work two types of augmentation have been applied: time stretching, and pitch shifting.

These two types of augmentation are accomplished by an old technique known as Phase Vocoder [45]. In this technique, signals are represented by short time amplitude and phase. The main purpose of this vocoder is to reduce transmission bandwidth. But it also serves the purpose of time stretching and pitch shifting. Phase vocoder is divided into three parts like Fig. 3.5. The analysis part extracts the information about magnitude and phase by doing short time Fourier transform (STFT) and synthesis part does the reverse i.e. inverse STFT (ISTFT). The audio effects are mainly done in the processing portion.

If any signal $f(t)$ is passed through a bank of band-pass filters $BP_1 \dots BP_n$ the regenerated signal can be expressed as (3.4)

$$f(t) \cong \sum_{n=1}^{N} f_n(t) \tag{3.4}$$

where, $f_n(t)$ is the output of filter nth filter among total of N band-pass filters. $f_n(t)$ also can be viewed as the convolution of $f(t)$ and $g_n(t)$, where $g_n(t)$ is the impulse response of nth filter. If $F(\omega_n, t)$ be the complex spectrum, then it can be expressed as (3.5)

$$F(\omega_n, t) = a(\omega_n, t) - jb(\omega_n, t) \tag{3.5}$$

where

$$a(\omega_n, t) = \int_{-\infty}^{t} f(\lambda)h(t-\lambda)cos\omega_n \lambda \, d\lambda \tag{3.6}$$

$$b(\omega_n, t) = \int_{-\infty}^{t} f(\lambda)h(t-\lambda)sin\omega_n \lambda \, d\lambda \tag{3.7}$$

Short time magnitude spectra and short time phase spectra can be computed by (3.8) and (3.9) respectively

$$|F(\omega_n, t)| = (a^2 + b^2)^{\frac{1}{2}} \tag{3.8}$$

$$\dot{\phi}(\omega_n, t) = (\frac{\dot{a}b + \dot{b}a}{a^2 + b^2}) \tag{3.9}$$

For sampled data at T sec interval, (3.6) and (3.7) can be rewritten as (3.10) and (3.11)

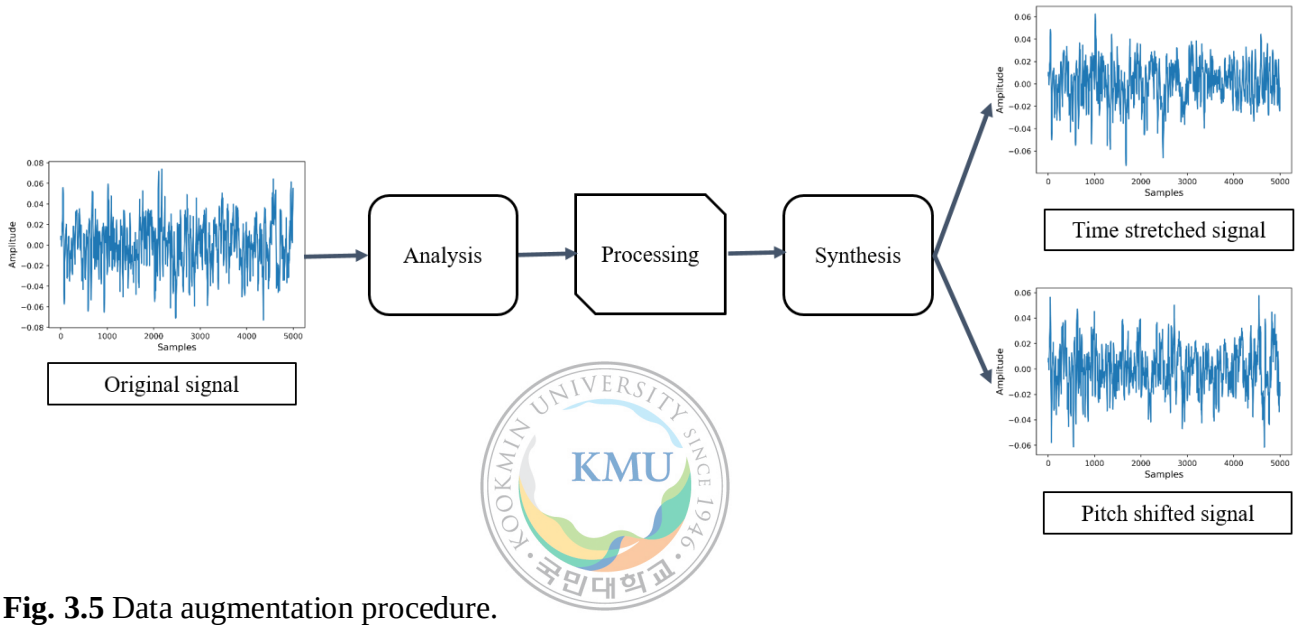

Fig. 3.5 Data augmentation procedure.

$$a(\omega_n, mT) = T \sum_{l=0}^{m} f(lT)[cos\omega_n lT]h(mT - lT) \tag{3.10}$$

$$b(\omega_n, mT) = T \sum_{l=0}^{m} f(lT)[sin\omega_n lT]h(mT - lT) \tag{3.11}$$

The difference values from (3.10) and (3.11) are obtained by (3.12) and (3.13)

$$\Delta a = a[\omega_n, (m+1)T] - a[\omega_n, mT] \tag{3.12}$$

$$\Delta b = b[\omega_n, (m+1)T] - b[\omega_n, mT] \tag{3.13}$$

So, in discrete form, (3.14) and (3.15) is the rewritten form of (3.8) and (3.9)

$$|F(\omega_n, mT)| = (a^2 + b^2)^{\frac{1}{2}} \tag{3.14}$$

$$\frac{\Delta\phi}{T}(\omega_n, mT) = \frac{1}{T}(\frac{\Delta ab + \Delta ba}{a^2 + b^2}) \tag{3.15}$$

The main advantage of phase vocoder is that it can separate temporal and frequency components. So, in the processing step, data augmentation is more convenient by this algorithm. In this step, time stretching and pitch shifting are done to perform the required augmentation.

3.3.1.1 Time stretching

The target of this kind of augmentation is to make the signal slower or faster than the original signal without affecting the signal's original pitch information. We can interpret the concept of phase vocoder from two view-points: either filter bank analysis or Fourier transform analysis. In the filter-bank interpretation there is time varying amplitude and frequency signals for each oscillator which work as control signals. These signals carry temporal information only. So, simply expand or compress the time duration of this signal is enough to achieve required time stretch. On the other hand, from Fourier transform consideration, if we want to slow down the original signal then in the synthesis part of phase vocoder the inverse FFT should be set apart further or vice versa. But another important consideration is missing here. When the IFFT spacing is increased then the phase interval between FFT points is affected. Let, the increment between successive phase values within a FFT bin is n degrees. After spacing, the same phase difference will be occurred over longer time period which means that the signal's frequency profile is altered. So, the solution to this problem is to rescale the phase value increment by which time scale is expanded ($2 \times n$ in this example). In the current study, data augmentation with time scale 2.0 is used.

3.3.1.2 Pitch shifting

The purpose of pitch shifting is to modify the pitch profile of the signal, whereas the time duration is kept as same as original signal. To easily accomplish this step, it is necessary to perform a time scale by the desired pitch transposition factor and after that the signal has to be played back at another sampling rate. For example, to raise the pitch of a signal by an octave, the signal first has to be time stretched by factor two. After this stage, the signal is then sped up by doubling the sampling rate thus the signal duration is kept same but a higher pitch is achieved. In mathematical terminology, increasing the pitch by one step means 12 semitones. So, increase or decrease the pitch scaling by n semitones means frequency multiply the factor

$2^{\frac{n}{12}}$. However, some extra processing (regarding spectral envelope) is needed to keep the intelligibility of the sound. This processing step is specially needed for a speech signal.

At the synthesis part of the phase vocoder, the signal is again reconstructed ($\widetilde{f_n}(mT)$) after the transmission or processing (like augmentation). The reconstruction after different processing is done by (3.16). The outputs for n channels are then added according to (3.4)

$$\widetilde{f_n}(mT) = |F(\omega_n, mT)|\cos\left(\omega_n, mT + T\sum_{i=0}^{m}\frac{\Delta\phi(\omega_n, lT)}{T}\right) \tag{3.16}$$

3.4 Feature scaling

Before giving input to the CNN network it is mandatory to scale the feature set for better performance. The tree based method does not require this kind of step. On the other hand, for gradient descent based optimization algorithm it is a must in order to have a performance boosting. By this way, the algorithm quickly finds the shortest path to the global minimization. Several types of minimization are possible like standardization (z-score normalization), min-max scaling, mean normalization. In this work, the z-score normalization is performed according to (3.17). The target of this scaling is to redistribute the features with mean, $\mu = 0$, and standard deviation, $\sigma = 1$.

$$y' = \frac{y - \bar{y}}{\sigma_y} \tag{3.17}$$

where, y and y' are the input data and rescaled data respectively, $\bar{y}$ is the mean of y, and σ_y is the standard deviation of y.

3.5 Train CNN model

In this study, two types of CNN architecture are proposed. These networks are classified as 1D and 2D networks based on their given input. The 2D architecture is more popular in ESC research than 1D. But this research's target is to develop a CNN model, where the input signal will be required minimal preprocessing as well as perform as like 2D model. This target is almost achieved and showed in the chapter 4 of this book. Table 3.1 represents the basic architecture of 1D CNN. The 1D CNN takes raw audio signal as input. The input size to the

first layer is 1 sec of audio file. After that five convolution blocks (denoted as ConvBlock-n; where n is the block sequence number) are used. These blocks are used as feature extractor modules. At the last stage, four dense layers are used as classifier module. Each ConvBlock consists of three layers – a 1D convolution layer, a Leaky rectified linear unit layer (Leaky ReLU), a batch normalization layer. The 1D convolution layer goes forward along with the time axis. Let, s be the input signal with n samples and k be the kernel with receptive field of size m then the convolution between s and k can be viewed as (18). The number of learned features by each layer is same as the number of filters in that layer. To add non-linearity after each convolution layer, leaky ReLU activation is applied on the previous layer output. Although ReLU is mostly used as activation function, leaky ReLU is used here for some specific advantage. The ReLU and leaky ReLU are mathematically defined by (3.19) and (3.20) respectively. From these equations it is clear that, the derivative of the ReLU will be always zero. On the other hand, Leaky ReLU is free from dying ReLU problem which occurs when the input value is negative. To utilize high learning rate and to reduce overfitting, batch normalization layer [46] is used here. This layer also replaces l_2-regularization and dropout layer. For a d dimensional layer where input is $x = (x^{(1)}, \ldots, x^{(d)})$, BN is done by (3.21). The output of this layer is like (3.22) where γ and β are also learned along model parameters.

$$z = (s * k)(i) = \sum_{j=1}^{m} k(j) \cdot s(i-j) \tag{3.18}$$

$$R(z) = \begin{cases} z, & z > 0 \\ 0, & z \leq 0 \end{cases} \tag{3.19}$$

$$R(z) = \begin{cases} z, & z > 0 \\ \alpha z, & z \leq 0 \end{cases} \tag{3.20}$$

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}} \tag{3.21}$$

$$z^{(k)} = \gamma^{(k)} \hat{x}^{(k)} + \beta^{(k)} \tag{3.22}$$

After each ConvBlock, a (4×1) maxpooling layer is employed to reduce the unnecessary features by downsampling. The maxpooling operation can be viewed as Fig. 3.6. Another

pooling layer called Global Average Pooling (GAP) is put before diving into the densely connected flatten layer. This is another kind of extreme pooling layer which only reduces the feature space in spatial domain but keeps the exact number of filters (depth). Mathematically speaking, if the previous layer's shape is $(h \times w \times d)$, then performing GAP on this layer will remake the shape into $(1 \times 1 \times d)$. In the dense block, four dense layers are used. Until the last dense layer, activation function is same as ConvBlock. At the last output layer, as prediction function, Softmax activation layer is used. This function gives the probability output for z input vector over K classes by (3.23). To train the whole network for training examples, Adam optimization [47] algorithm is used following Algorithm 1 [48]. It combines the idea from RMSProp and Momentum. As objective function or loss function, categorical cross-entropy is used as in (3.24). The whole network is trained by 300 times (epochs). To prevent overfitting, early stopping criteria was used for 30 epochs patience and 0.01 is set as learning rate. Same training strategy is also followed for 2D CNN model. Table 3.2 describes the different layers used in two dimensional model. In this type, the kernel operates both across the time and frequency axis over the input Gammatone spectrogram as well as the further layers. A same convention [r/s, f] × b is followed in both networks where it

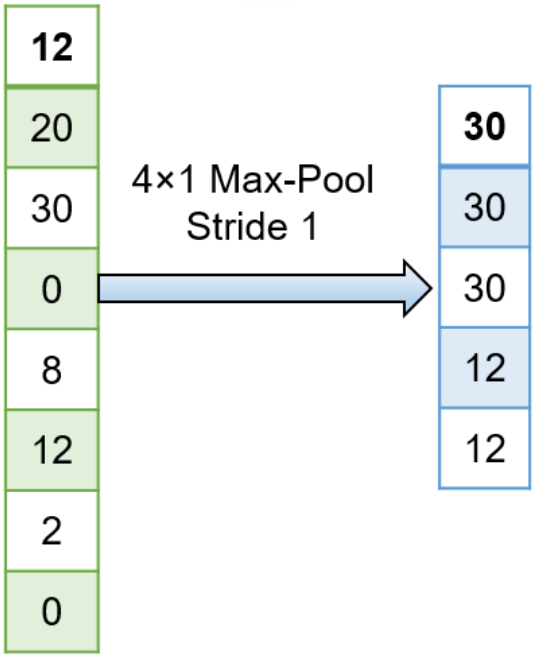

Fig. 3.6 Visualization of Maxpooling operation.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (3.23)$$

$$J = -\sum_{j=1}^{K} y_j log(z_j) \quad (3.24)$$

means [receptive field/strides, no. of filters] × repetition time. Although, in 1D network for each ConvBlock the stride number is kept same, for 2D network it alters respectively 3 and 1. For better visualization of the structure of proposed CNN models, Fig. 3.7 can be referred.

Table 3.1. Our proposed 1D CNN architecture for ESC.

Input	22500×1 time domain waveform
ConvBlock-1	[80/4, 48]×1
MaxPooling-1	4×1
ConvBlock-2	[3/1, 48]×2
MaxPooling-2	4×1
ConvBlock-3	[3/1, 96]×2
MaxPooling-3	4×1
ConvBlock-4	[3/1, 192]×2
MaxPooling-4	4×1
ConvBlock-5	[3/1, 384]×2
MaxPooling-5	4×1
GlobalAveragePooling-1	
Dense-1	[768]×1
Dense-2	[512]×1

Dense-3	[128]×1
Dense-4	[10]×1

Table 3.2. Our proposed 2D CNN architecture for ESC.

Input	(64 × 98 × 1) Gammatone spectrogram
ConvBlock-1	[8/1, 48]×1
ConvBlock-2	[3/s, 48]×2
MaxPooling-1	4×1
ConvBlock-3	[3/s, 96]×2
MaxPooling-2	4×1
ConvBlock-4	[3/s, 192]×2
MaxPooling-3	4×1
ConvBlock-5	[3/s, 384]×2
MaxPooling-4	4×1
GlobalAveragePooling-1	
Dense-1	[768]×1
Dense-2	[512]×1
Dense-3	[128]×1
Dense-4	[10]×1

Algorithm 1: Weight update using Adam optimizer by minimizing loss function J for learning rate α with the help of hyper parameter (β_1, β_2).

Step-1: $v_{dW} = \beta_1 v_{dW} + (1 - \beta_1)\frac{\partial J}{\partial W}$

Step-2: $s_{dW} = \beta_2 s_{dW} + (1 - \beta_2)(\frac{\partial J}{\partial W})^2$

Step-3: Bias correction, $v_{dw}^{updated} = \frac{v_{dW}}{1-(\beta_1)^t}$, $s_{dw}^{updated} = \frac{s_{dW}}{1-(\beta_2)^t}$

Step-4: Weight update: $W = W - \alpha \frac{v_{dw}^{updated}}{\sqrt{s_{dw}^{updated}} + \epsilon}$

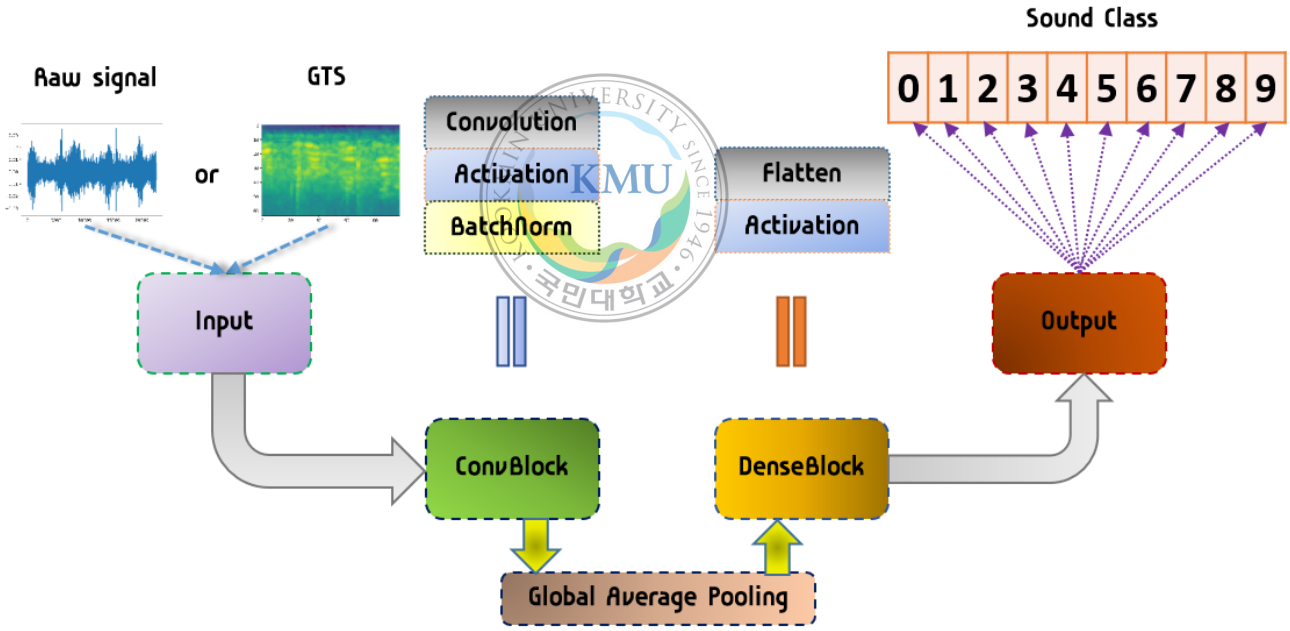

Fig. 3.7 Block diagram of proposed 1D and 2D CNN model for ESC system.

3.6 Test model

In the test phase, the evaluation is performed on the separated validation set at each fold. In training stage, input is generated by 1s framing. So, the test dataset's signal is also framed as similar. After predicting over several frames of a single signal, no decision aggregation rule like majority voting, summing rule is not applied.

Chapter 4

Dataset description, results and discussion

4.1 Dataset description

The proposed classification networks are trained and tested on two environment sound datasets. The datasets are collected from [39] and [40]. In order to prove the generalization of proposed models, two datasets are used. Total 20 classes are in these datasets. The total number of audio recordings are 9132.

4.1.1 ESC-10 dataset

The dataset, collected from [39], is called ESC-10 which consists of 400 labeled audio clips. Each clip is 5-second long and is sampled at 44.1 kHz rate. The total audio recordings are divided equally into 10 classes (40 examples per class). Fig. 4.1(a) demonstrates the distribution of classes for this dataset.

4.1.2 US-8K dataset

UrbanSound8K (US-8K) datasets from [40] has 8732 labeled sound excerpts. The time length of each clip is <=4-seconds and is sampled at 44.1 kHz rate. The audio clips are divided almost equally into 10 classes. The distribution of data is shown in Fig. 4.1(b).

4.2 Experimental setup

To train and test the model, and for performing other steps in the proposed ESC system, a desktop computer with windows 10 operating system (OS) is used. The hardware setup consists of an Intel-core i7 central processing unit (CPU), 16 Gigabytes(GB) random access memory (RAM), and a NVIDIA GTX- 2060 graphics unit processor (GPU). Python programming language is used for performing the simulation. The sound preprocessing is done by the librosa [49] library written for python programming language. The deep learning part is accomplished by popular ML

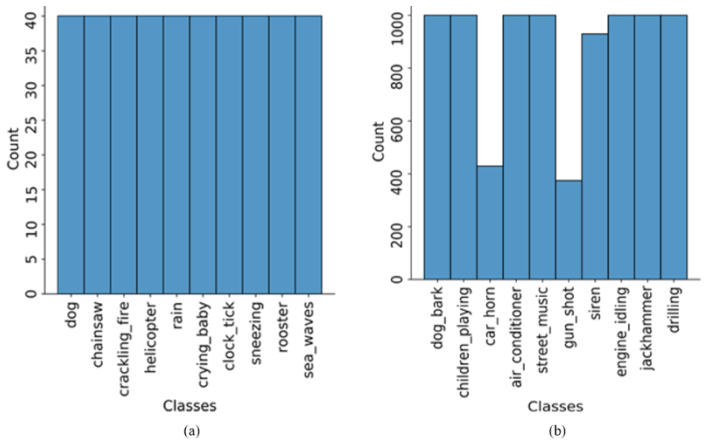

Fig. 4.1 Distribution of different classes in the two used datasets: (a) for ESC-10 dataset, and (b) for US-8K dataset.

framework Keras with TensorFlow backend. For K-fold cross validation, each time the whole dataset is splitted into two parts: train set (80%) and validation set (20%). The result here is shown for validation set. This is done by scikit-learn library [50]. Due to hard limitation on memory capacity, data augmentation only applied on only ESC-10 dataset. Moreover, as the number of data is low in this dataset, it was thought data augmentation would be useful for better learning by the CNN models.

4.3 About classification metrics

Among so many performance metrics, to measure the ability of a classifier deep learning model, six metrics are selected. They are: (a) accuracy (overall and per-class), (b) precision, (c) recall, (d) F_1-score, (e) Matthew correlation coefficient (MCC), and (f) Cohen's kappa coefficient (CKC). Besides this, several graphical analyses like receiver operating characteristic curve (ROC), precision recall (PR) curve and their respective area under curve (AUC) are calculated. Moreover, to analysis the failure cases by our proposed model are also discussed.

4.3.1 Confusion matrix

The most common tool to visualize the performance of a machine learning or statistical model is confusion matrix. As the name suggests, it is justified that how much the model

confuses between target classes. This metrics is helpful to derive the basic form of other performance metrics. Fig. 4.2 is the basic construction block for a confusion matrix which can

Confusion Matrix		Actual class	
		Class 1	Class 2
Predicted class	Class 1	True positive (TP)	False positive (FP)
	Class 2	False negative (FN)	True negative (TN)

Fig. 4.2. Example of building a confusion matrix basic block.

later be extended for more class. In this figure, each column represents the examples in an accurate class while each row represents the occasions in an anticipated class.

4.3.2 Accuracy

Accuracy is the primary measurement of any classification model. It is defined as the percentage of predictions that any model gets right. The mathematical formulation of classification is like (4.1). The maximum value for perfect classifier is 1.0 or 100% and for the worst it is 0.0 or 0%. On the other hand, "accuracy paradox" is related part in this topic. This term is mainly used for hugely unbalanced class where accuracy is not considered as a great tool. Therefore, in this research, other metrics are also incorporated especially precision, recall, and F1-score.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

$$Accuracy = \frac{True\ positive\ (TP) + True\ negative\ (TN)}{TP + TN + False\ positive\ (FP) + False\ negative\ (FN)} \tag{4.1}$$

4.3.3 Precision

Precision tries to answer the question "What extent of positive identifications by the model is really accurate?" This metric can be defined as (4.2). From this equation, it can also be observed that, a model can achieve highest precision 1.0 when there is no false positive case.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

4.3.4 Recall

Recall is the fraction measurement of actual positives that are identified correctly. (4.3) is the mathematical expression of recall. If there is no false negative case, the model's recall would be 1.0.

$$Recall = \frac{TP}{TP+FN} \tag{4.3}$$

4.3.5 F_1-score

To relate both the precision and recall at the same time, F-score (also known as F-measure) is used. The harmonic mean of precision and recall is identified as F_1-score or Soresen-Dice coefficient. According to (4.4), highest possible value for F_1-score is 1.0 when precision and recall both are 1.0 and F_1-score is 0.0 when either precision or recall is 0.0.

$$\boldsymbol{F_1 = 2 \times \frac{precsion \times recall}{precision + recall}} \tag{4.4}$$

4.3.6 Matthews correlation coefficient (mcc)

This is also famous as phi coefficient. First it was proposed at [51] for binary classification which was later developed for multiclass classification also. This metric is a great help when the classes are of different size. From the confusion matrix, mcc is defined as (4.5) which is a balanced measure. The highest value +1.0 means perfect prediction, 0.0 means random prediction and -1.0 indicates inverse prediction.

$$mcc = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{4.5}$$

4.3.7 Cohen's kappa coefficient (ckc)

To measure the level of agreement between two annotators, ckc metric is used. The highest possible value for this metric is 1.0 which means perfect agreement. If the ckc value is greater than 0.8, then the model is considered as pretty good classifier. The mathematical equation of ckc is (4.6).

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{4.6}$$

where, p_o is the relative observed agreement among raters and p_e is the hypothetical probability of chance agreement.

4.3.8 Receiver operating characteristic curve (ROC)

The diagnostic ability of a classifier system is analyzed by this ROC curve. It is a graphical representation of false positive rate (FPR) vs. true positive rate (TPR). The TPR or sensitivity is actually as same as recall. FPR or fall-out is the opposite of TPR which estimates the incorrectly classified positive example among all of the negative samples.

$$FPR = \frac{FP}{FP+TN} \tag{4.7}$$

Fig. 4.3 illustrates different situation under different conditions. Here, the point at (0,1) coordinates indicates the bet result i.e. perfect classifier. The diagonal line is for threshold on the decision. Any point on that line means random guess (here point B). The points above the red line indicates that the result is better than random guess (point A and C`). On the other hand, any point under the

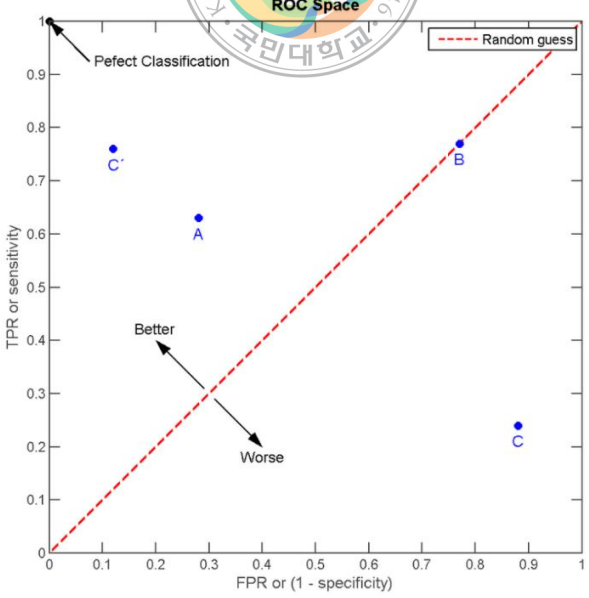

Fig. 4.3 Receiver operating characteristic (ROC) curve for different conditions [52].

diagonal line is acknowledged as worse result than random guess. So, the area under curve (AUC) for the perfect classifier will be 1.0.

4.3.9 Precision recall (PR) curve

Precision recall curve is the outcome of the tradeoff between precision and recall. The highest possible value for the AUC of PR curve is 1.0. Fig. 4.4 illustrates the best and worst case for PR curve.

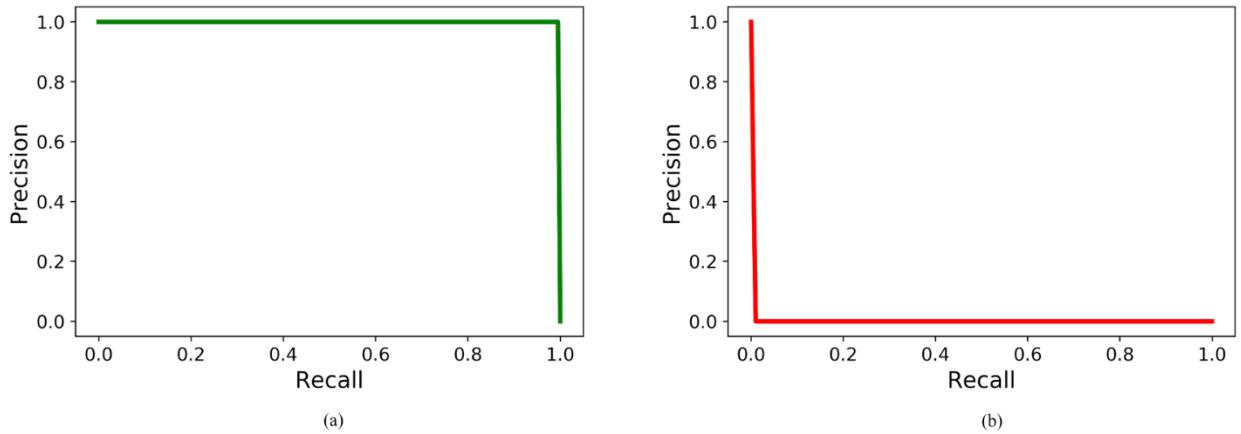

Fig. 4.4 Precision-recall curve for (a) ideal and (b) worst case.

4.4 Results and discussion

4.4.1 For raw input (1D CNN model)

In this section, the performance of the suggested 1D CNN model is analyzed i.e. how well it can learn the inherited features from the given audio signal without any kind of pre-processing. The results, shown here are obtained after classifying the validation dataset using trained 1D CNN model. Fig. 4.5 and 4.6 show the best and worst confusion matrices by 5-fold cross validation (CV). These matrices give an idea about how many classes are misclassified. For US-8K dataset, in each fold "*street_music*" is the most misclassified sound. This class is mostly predicted as "*children_playing*". If we look at the Fig. 4.6, "*sneezing*" is misclassified at maximum times for ESC-10 dataset. The model expects "*rooster*" as the same class as "*sneezing*". Table 4.1-4.5 and Table 4.6-4.10 tabulates the results for 5-fold CV for both datasets, respectively. In case of US-8K dataset, the model performs better than ESC-10 dataset. For US-8K dataset, the maximum accuracy is found at fold-3 whereas for ESC-10 dataset the maximum value is at fold-2. The other performance indicating parameters (precision, recall, F_1-score) in those results are similar as the understanding from confusion matrix. As the MCC

and CKC values of 0.80 are considered as excellent, in all cases the values are either over or near the threshold. The learning curves for proposed 1D CNN model are shown at Fig. 4.7 and 4.8. The maximum training accuracy for US-8K dataset is around 97% and for ESC-10 dataset it is 93%. Although the loss is less important in case of classification, loss is also relatively low for each case. Another important factor to determine the model's prediction accuracy is the analysis of receiver operating characteristic curve and to measure the area under ROC curve (AUC). Fig. 4.9 and Fig 4.10 depicts the ROC curves and theirs' AUC also for each dataset. For better visualization, in each fold the curves are drawn as 3 classes, 3 classes and last 4 classes in this manner. In each fold of US-8K dataset, AUC is found around 1.00. But among these "*street_music*" and "*children_playing*" classes' AUC is low compared to others. The same near to 1.00 AUC can be observed for ESC-10 dataset also where AUC for "*sneezing*" and "*rooster*" are less. Expected results are also found from model's prediction in terms of precision recall curves of their AUC. From PR curve, it is seen that the model's prediction capability for ESC-10 validation dataset is sometimes a little poorer than for US-8K dataset. While all the folds in US-8K exhibit same type of performance, in case of ESC-10 dataset in some folds (specially Fig. 4.12 (i, l)) model cannot perform so good. As no deep learning model still now is quite perfect, this study has also faced some unsuccessful cases. Some failure cases of misclassification in both datasets are shown at Fig. 4.13 and Fig. 4.14. Their input pattern is also shown for normalized amplitude vs. some samples. These figures do not necessarily imply the performance in the prior mentioned quality metrics because the disappointed cases are choiced randomly.

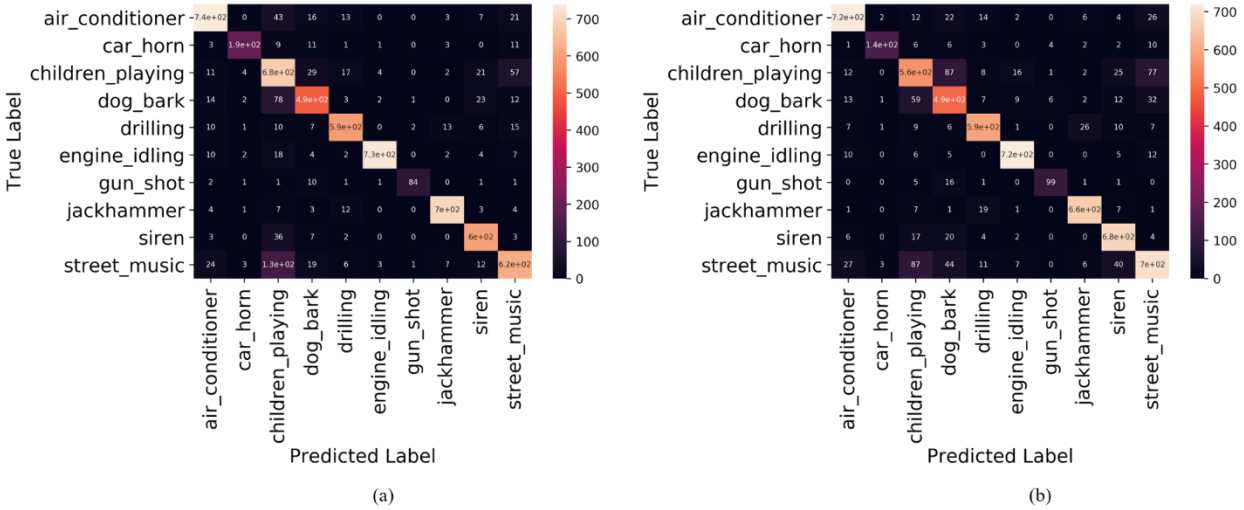

Fig. 4.5 Best and poorest Confusion matrices for 5-fold cross validation on US-8K dataset. (a)-(b) represents the output for fold-3 and fold-1 respectively for proposed 1D CNN network.

Table 4.1 Classification report of fold-1 for US-8K validation dataset (proposed 1D CNN).

Class	Precision	Recall	F_1-score	Accuracy	MCC	CKC
AI	90%	89%	0.90	85%	0.83	0.83
CH	95%	81%	0.88			
CP	73%	71%	0.72			
DB	70%	78%	0.74			
DR	90%	90%	0.90			
EI	95%	95%	0.95			
GS	90%	80%	0.85			
JH	94%	95%	0.94			
SI	87%	93%	0.90			
SM	80%	76%	0.78			

Table 4.2 Classification report of fold-2 for US-8K validation dataset (proposed 1D CNN).

Class	Precision	Recall	F_1-score	Accuracy	MCC	CKC
AI	87%	95%	0.91	86%	0.85	0.85

CH	96%	83%	0.89			
CP	77%	76%	0.71			
DB	80%	78%	0.79			
DR	94%	88%	0.91			
EI	96%	95%	0.95			
GS	98%	75%	0.85			
JH	94%	94%	0.94			
SI	83%	95%	0.89			
SM	73%	79%	0.76			

Table 4.3 Classification report of fold-3 for US-8K validation dataset (proposed 1D CNN).

Class	**Precision**	**Recall**	**F_1-score**	**Accuracy**	**MCC**	**CKC**
AI	90%	88%	0.89	87%	0.85	0.85
CH	93%	83%	0.88			
CP	67%	82%	0.74			
DB	82%	78%	0.80			
DR	91%	90%	0.91			
EI	99%	94%	0.96			
GS	95%	82%	0.88			
JH	96%	95%	0.96			
SI	89%	92%	0.90			
SM	83%	75%	0.79			

Table 4.4 Classification report of fold-4 for US-8K validation dataset (proposed 1D CNN).

Class	**Precision**	**Recall**	**F_1-score**	**Accuracy**	**MCC**	**CKC**
AI	94%	89%	0.82	86%	0.85	0.85
CH	82%	92%	0.87			

CP	69%	77%	0.73			
DB	83%	76%	0.80			
DR	91%	88%	0.89			
EI	93%	97%	0.95			
GS	88%	81%	0.84			
JH	91%	97%	0.94			
SI	86%	90%	0.88			
SM	84%	74%	0.79			

Table 4.5. Classification report of fold-5 for US-8K validation dataset (proposed 1D CNN).

Class	**Precision**	**Recall**	**F_1-score**	**Accuracy**	**MCC**	**CKC**
AI	81%	91%	0.86	85%	0.84	0.84
CH	92%	85%	0.88			
CP	81%	67%	0.73			
DB	74%	84%	0.79			
DR	93%	90%	0.91			
EI	95%	95%	0.95			
GS	89%	84%	0.86			
JH	91%	97%	0.94			
SI	90%	87%	0.88			
SM	77%	76%	0.76			

Table 4.6. Classification report of fold-1 for ESC-10 validation dataset (proposed 1D CNN).

Class	**Precision**	**Recall**	**F_1-score**	**Accuracy**	**MCC**	**CKC**
CS	87%	89%	0.88	80%	0.78	0.78
CT	94%	79%	0.86			

CF	92%	78%	0.84			
CB	90%	88%	0.89			
DB	81%	57%	0.67			
HL	73%	98%	0.84			
RA	91%	65%	0.76			
RO	79%	47%	0.59			
SW	71%	93%	0.81			
SN	62%	96%	0.76			

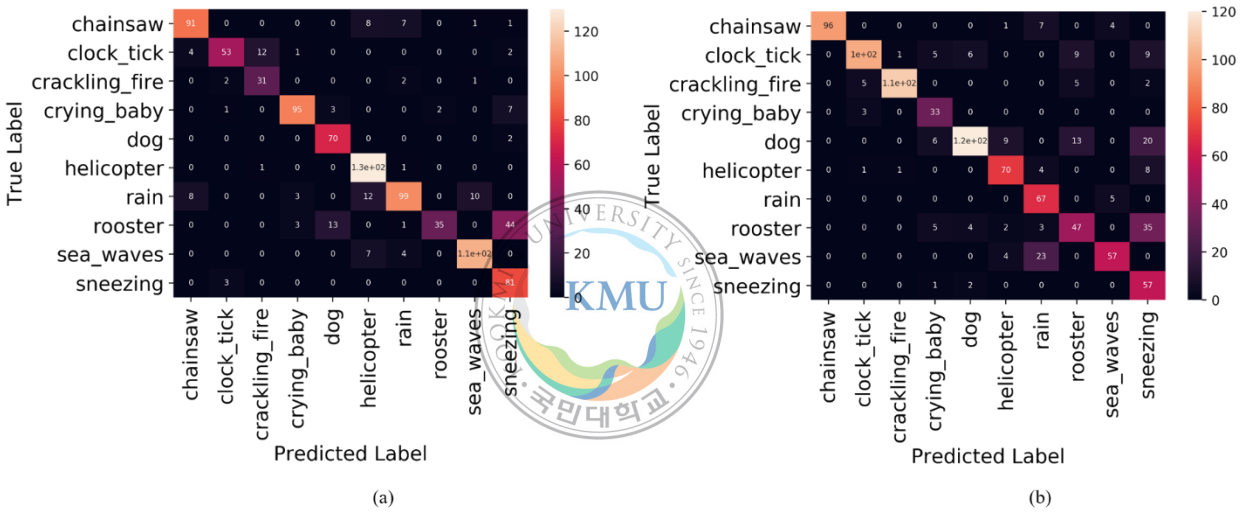

Fig. 4.6 Best and poorest confusion matrices for 5-fold cross validation on ESC-10 dataset. (a)-(b) represents the output for fold-2 and fold-3 respectively for proposed 1D CNN network.

Table 4.7 Classification report of fold-2 for ESC-10 validation dataset (proposed 1D CNN).

Class	Precision	Recall	F_1-score	Accuracy	MCC	CKC
CS	88%	84%	0.86	83%	0.81	0.81
CT	90%	74%	0.81			
CF	70%	86%	0.78			
CB	93%	88%	0.90			
DB	81%	97%	0.89			

HL	83%	98%	0.90			
RA	87%	75%	0.80			
RO	95%	36%	0.53			
SW	90%	91%	0.90			
SN	59%	96%	0.73			

Table 4.8 Classification report of fold-3 for ESC-10 validation dataset (proposed 1D CNN).

Class	**Precision**	**Recall**	**F_1-score**	**Accuracy**	**MCC**	**CKC**
CS	100%	89%	0.94	79%	0.77	0.76
CT	92%	77%	0.84			
CF	98%	90%	0.94			
CB	66%	92%	0.77			
DB	91%	71%	0.80			
HL	81%	83%	0.82			
RA	64%	93%	0.76			
RO	64%	49%	0.55			
SW	86%	68%	0.76			
SN	44%	95%	0.60			

Table 4.9 Classification report of fold-4 for ESC-10 validation dataset (proposed 1D CNN).

Class	**Precision**	**Recall**	**F_1-score**	**Accuracy**	**MCC**	**CKC**
CS	96%	86%	0.85	80%	0.78	0.78
CT	90%	74%	0.81			
CF	81%	88%	0.84			
CB	92%	92%	0.95			
DB	100%	60%	0.75			

HL	92%	71%	0.80			
RA	72%	92%	0.80			
RO	63%	54%	0.58			
SW	81%	88%	0.84			
SN	54%	91%	0.68			

Table 4.10 Classification report of fold-5 for ESC-10 validation dataset (proposed 1D CNN).

Class	**Precision**	**Recall**	**F_1-score**	**Accuracy**	**MCC**	**CKC**
CS	96%	90%	0.93	80%	0.78	0.78
CT	84%	90%	0.87			
CF	82%	100%	0.90			
CB	86%	86%	0.86			
DB	91%	71%	0.80			
HL	96%	81%	0.87			
RA	78%	86%	0.82			
RO	90%	53%	0.67			
SW	89%	73%	0.80			
SN	53%	92%	0.67			

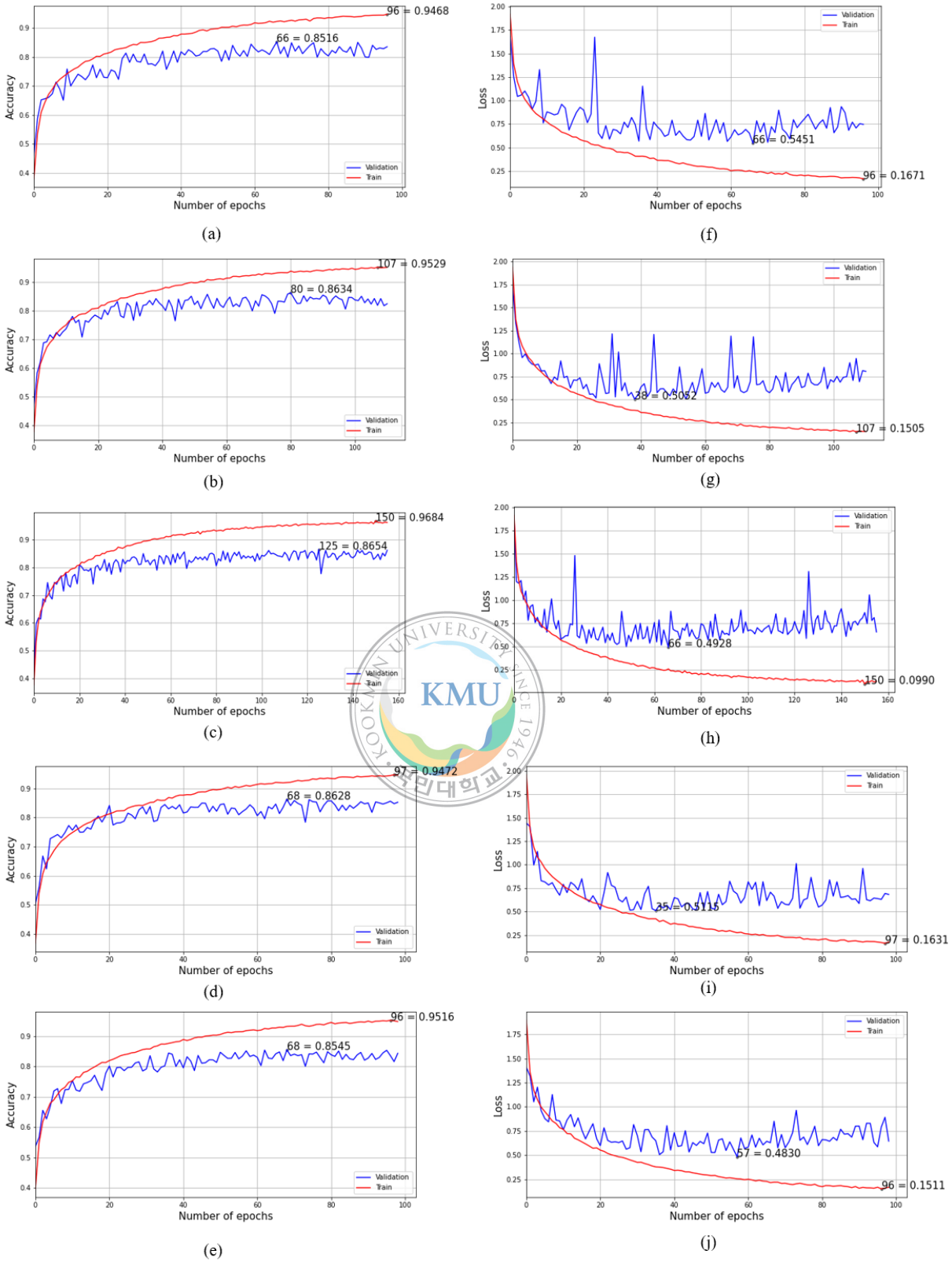

Fig. 4.7 Learning curves (accuracy and loss per epoch) for raw input of US-8k dataset to our proposed 1D CNN network. The graphs for 5-fold cross validation are shown: (a)-(e) accuracy

per epoch for fold 1-5, and (f)-(j) loss per epoch for fold 1-5. The best accuracy and minimum loss are marked in the respective figure.

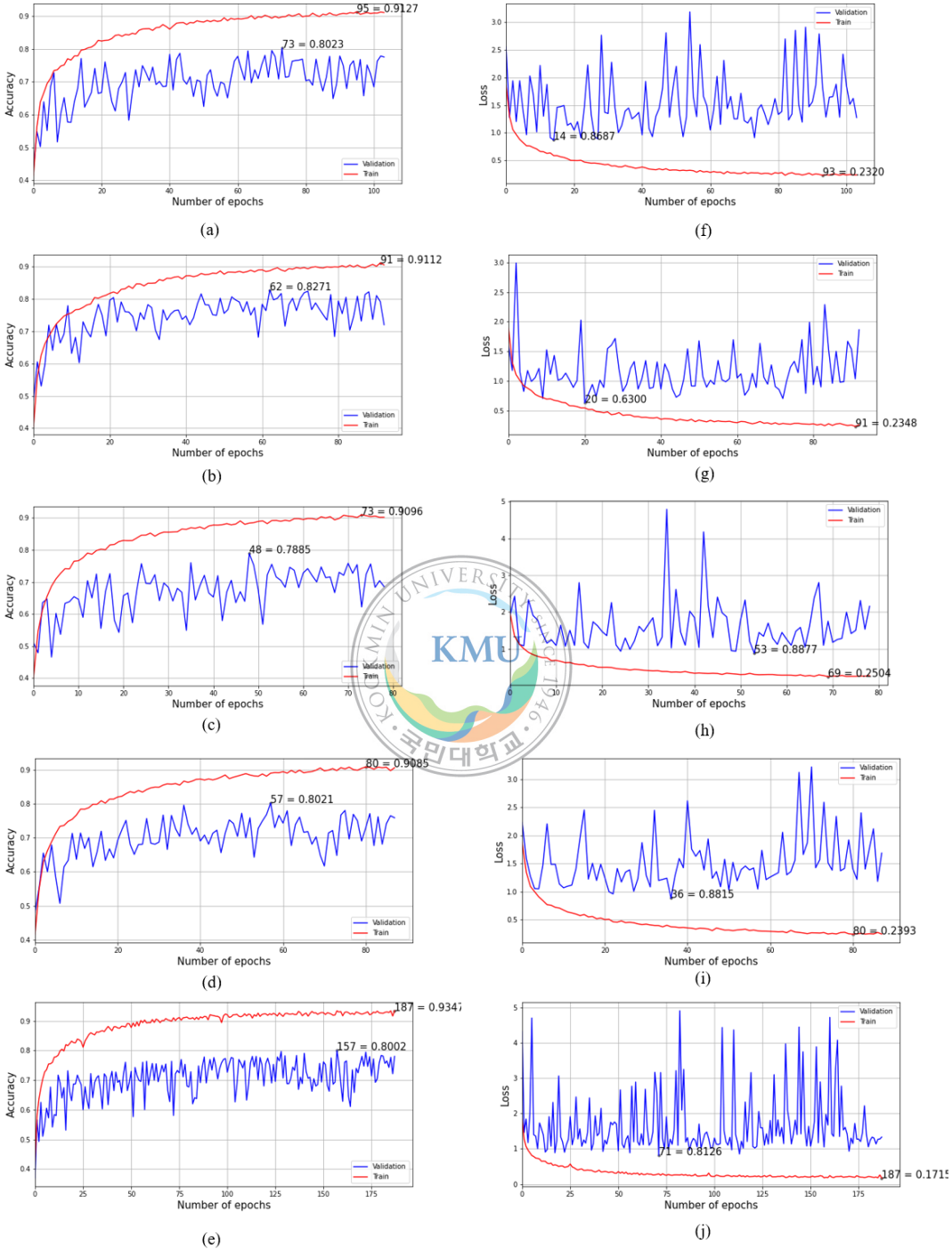

Fig. 4.8 Learning curves (accuracy and loss per epoch) for raw input of ESC-10 dataset to our proposed 1D CNN network. The graphs for 5-fold cross validation are shown: (a)-(e) accuracy

per epoch for fold 1-5, and (f)-(j) loss per epoch for fold 1-5. The best accuracy and minimum loss are marked in the respective figure.

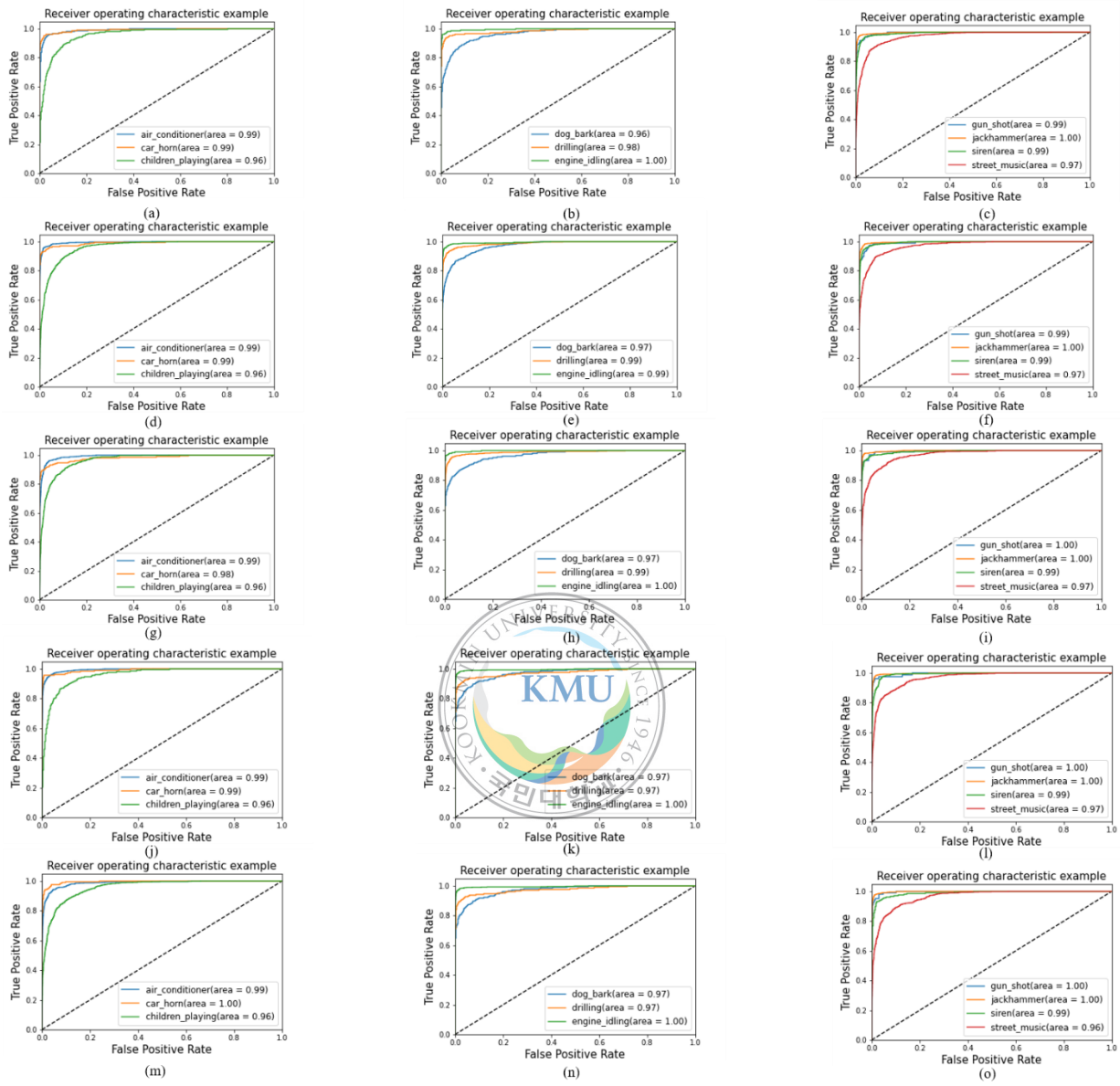

Fig. 4.9 ROC curve and respective AUC for raw input of US-8K dataset to our proposed 1D CNN network. The graphs for 5-fold cross validation are shown. Each row represents the ROC and AUC of 10-class for 5-fold cross validation.

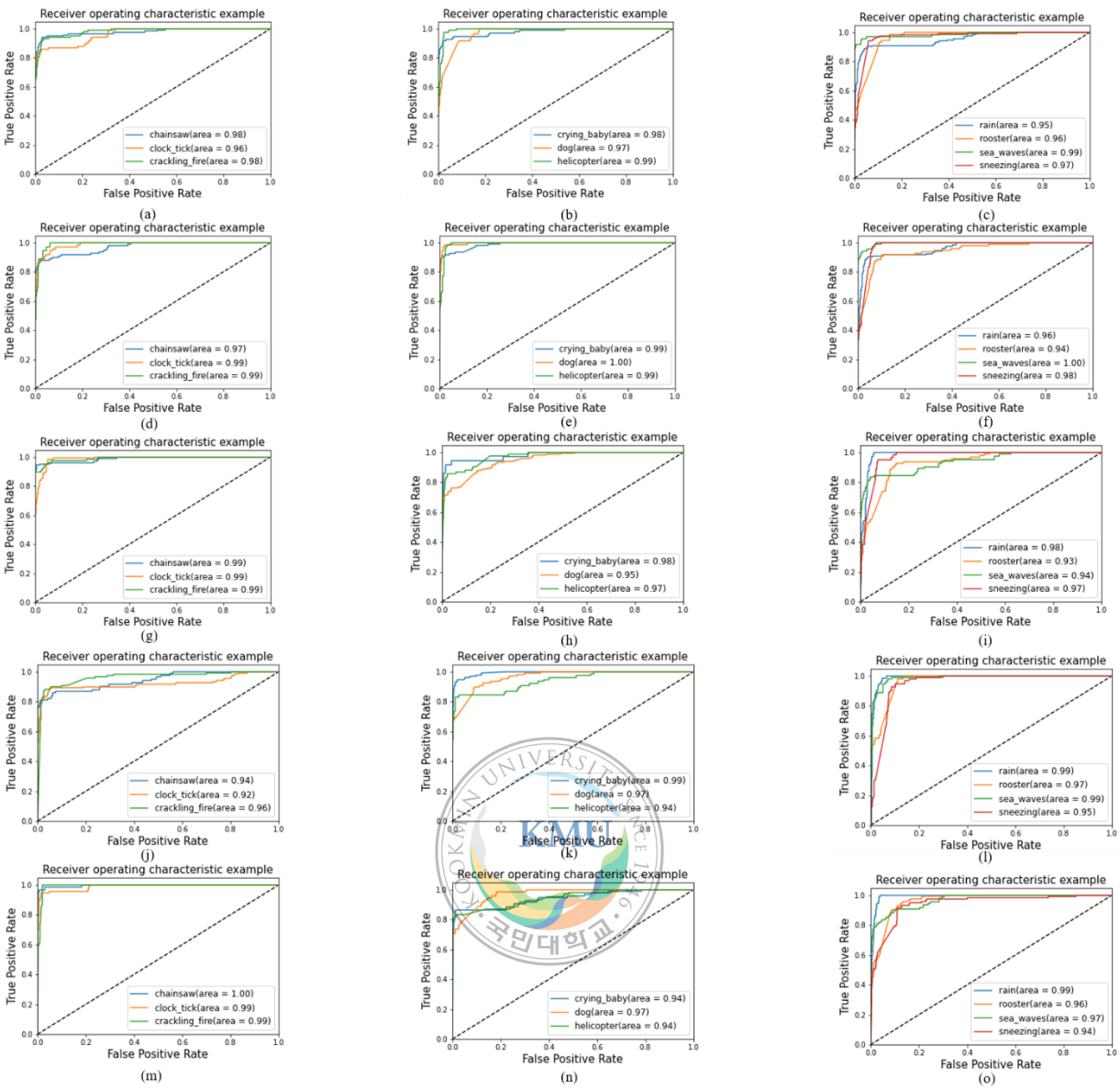

Fig. 4.10 ROC curve and respective AUC for raw input of ESC-10 dataset to our proposed 1D CNN network. The graphs for 5-fold cross validation are shown. Each row represents the ROC and AUC of 10-class for 5-fold cross validation.

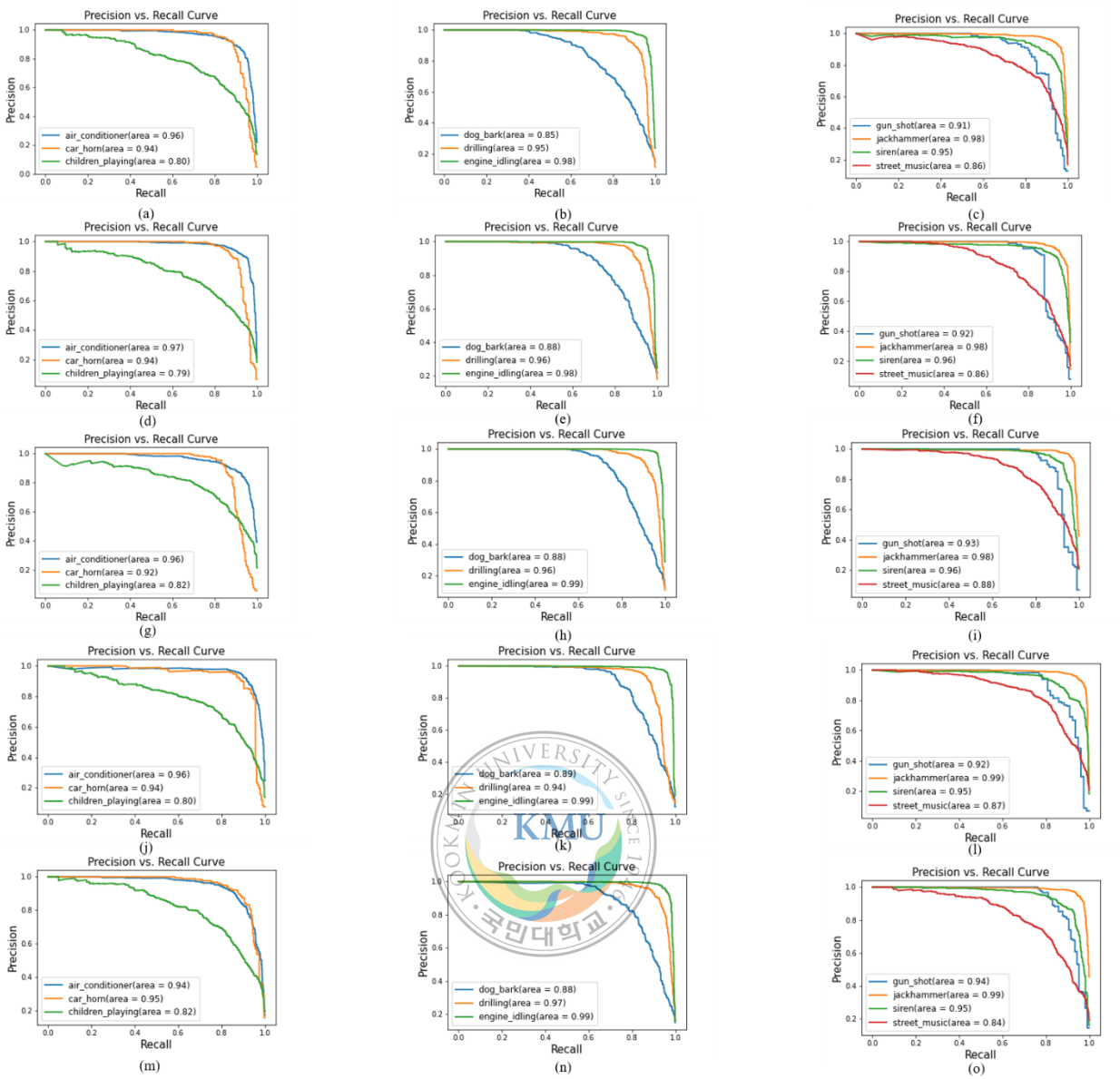

Fig. 4.11 Precision vs. Recall (PR) curve and respective area under curve (AUC) for raw input of US-8K dataset to our proposed 1D CNN network. The graphs for 5-fold cross validation are shown. Each row represents the PR and AUC of 10-class for 5-fold cross validation.

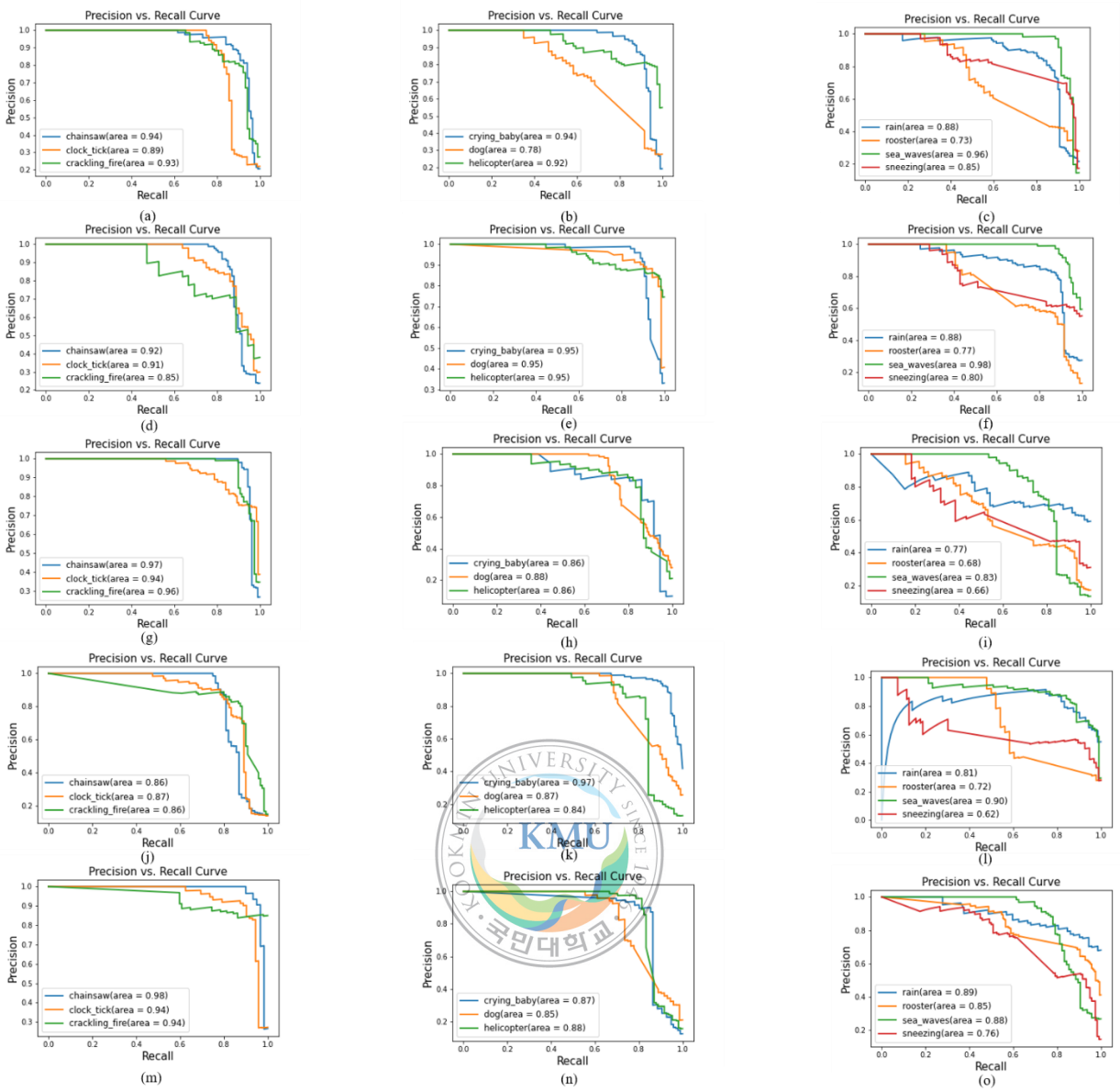

Fig. 4.12 Precision vs. Recall (PR) curve and respective area under curve (AUC) for raw input of ESC-10 dataset to our proposed 1D CNN network. The graphs for 5-fold cross validation are shown. Each row represents the PR and AUC of 10-class for 5-fold cross validation.

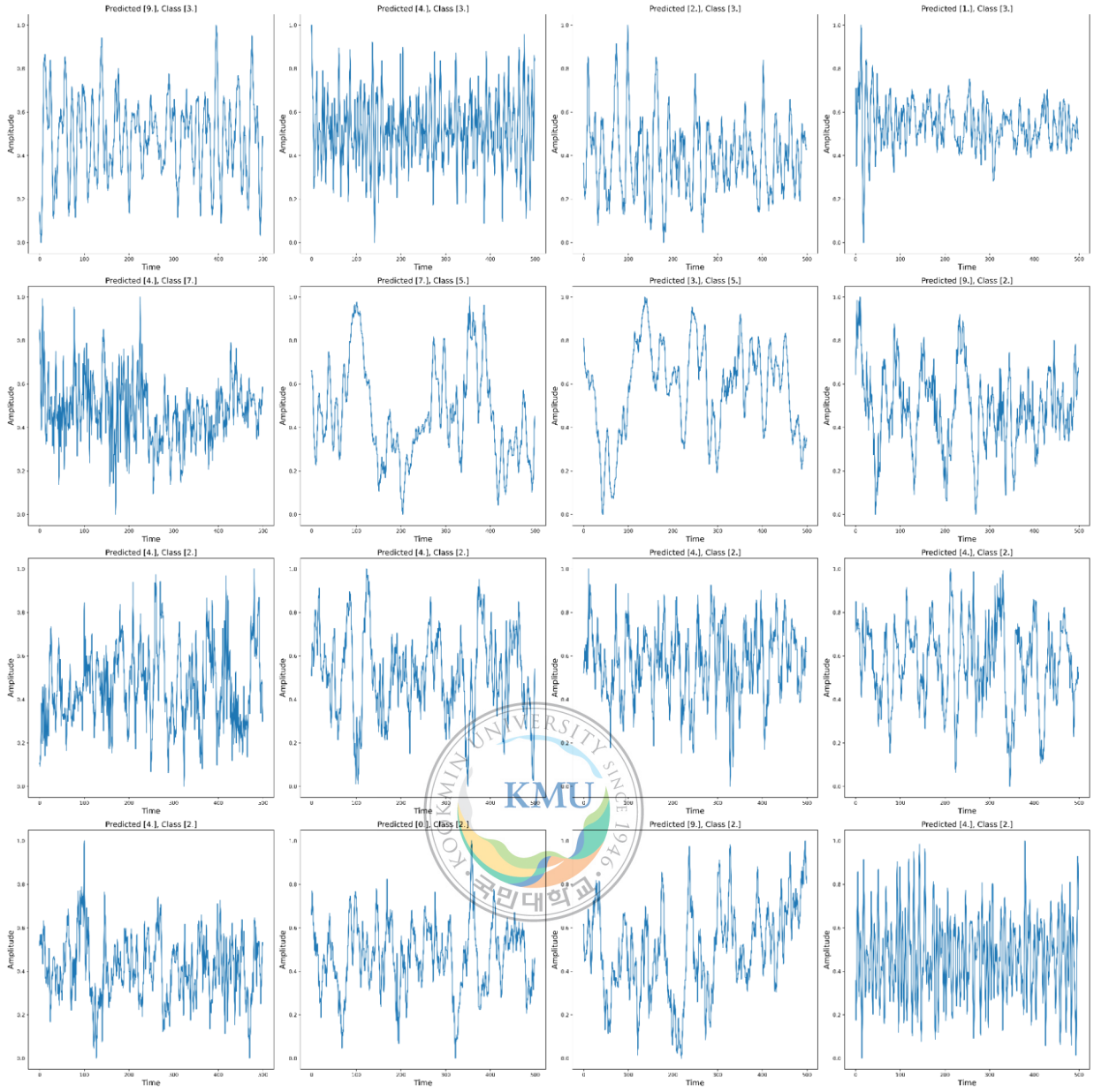

Fig. 4.13 Some misclassified classes by our proposed 1D network tested on US-8K dataset.

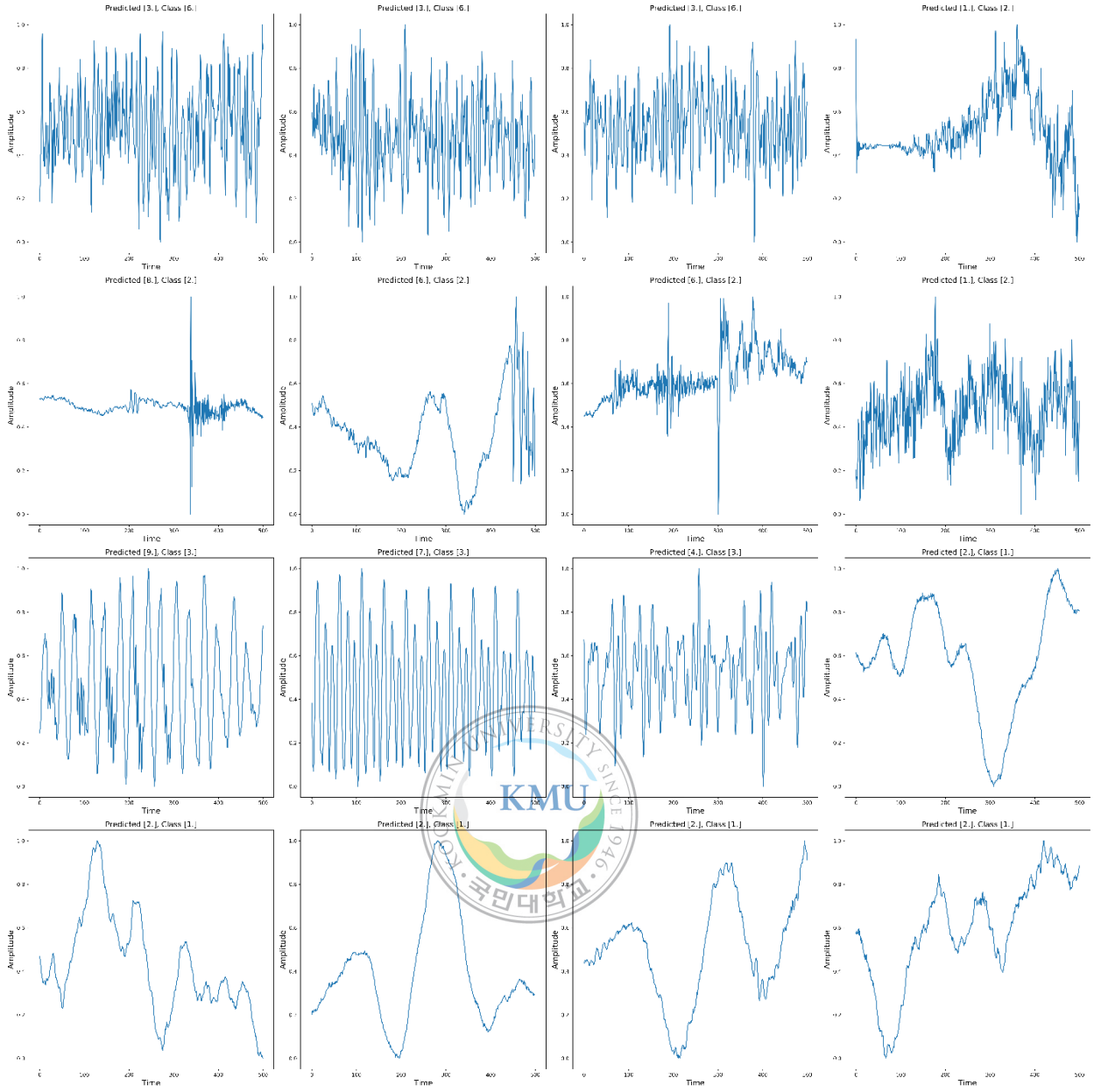

Fig. 4.14 Some misclassified classes by our proposed 1D network tested on ESC-10 dataset.

4.4.2 For Gammatone spectrogram input (2D CNN model)

In this part, the performance of the recommended 2D CNN model is dissected for example how well it can deal with the pre-processed input (Gammatone spectrogram). The outcomes, appeared here are gotten after classifying the validation dataset utilizing proposed 2D CNN model. Fig. 4.15 and 4.16 show the best and worst confusion matrices by 5-fold cross validation (CV). These matrices give an idea about how many classes are misclassified. Some results are found as same as the findings in case of 1D CNN model. For US-8K dataset, in each fold

"*street_music*" and "*children_playing*" are the most misclassified sounds. These sounds are outdoor sounds. So, probably interaction with some other common sounds in the signal makes difficulty to classify it correctly. Again, if we look at the Fig. 4.16, many times "*rooster*" is misclassified as "*sneezing*". Table 4.11-4.15 and Table 4.16-4.20 organizes the results for 5-fold CV for both datasets respectively. In case of US-8K dataset, the model performs better than ESC-10 dataset. For US-8K dataset, same maximum accuracy (91%) is found at fold-3 and fold-4 whereas for ESC-10 dataset the maximum value of accuracy (84%) is at fold-2. The other performance indicating parameters (precision, recall, F_1-score) in those results are similar as the understanding from confusion matrix. When number of false positives increases, precision goes down and increasing the number of false negative results in low recall. Looking at the classification report tables (Table-4.11~4.20), low precision and recall are found at the cases of high number of misclassification. As the MCC and CKC values of 0.80 are considered as excellent, in all cases the values are either over or near this threshold. The learning curves for proposed 2D CNN model are shown at Fig. 4.17 and 4.18. The maximum training accuracy for US-8K dataset is around 96% and for ESC-10 dataset it is 97%. Although the loss is less important in case of classification, loss is also relatively low for each case. Fig. 4.19 and Fig 4.20 depicts the ROC curves and theirs' AUC also for each dataset. The same manner for curve drawing is followed here as 1D CNN network. In each fold of US-8K dataset, AUC is found around 1.00. The same near to 1.00 AUC can be observed for ESC-10 dataset also where AUC for "*sneezing*" and "*rooster*" are less. Expected results are also found from model's prediction in terms of precision recall curves of their AUC. From PR curve, it is seen that the model's prediction capability for ESC-10 validation dataset is sometimes a little poorer than for US-8K dataset. While all the folds in US-8K exhibit same type of performance, in case of ESC-10 dataset in some folds (specially Fig. 4.22 (f, i)) model cannot perform so good. Some disappointment instances of misclassification in both datasets are appeared at Fig. 4.23 and Fig. 4.24. Their input spectrogram (GTS) is likewise appeared for filter index versus time steps. These figures don't really suggest the exhibition in the earlier referenced quality measurements on the grounds that the unwanted cases are picked arbitrarily.

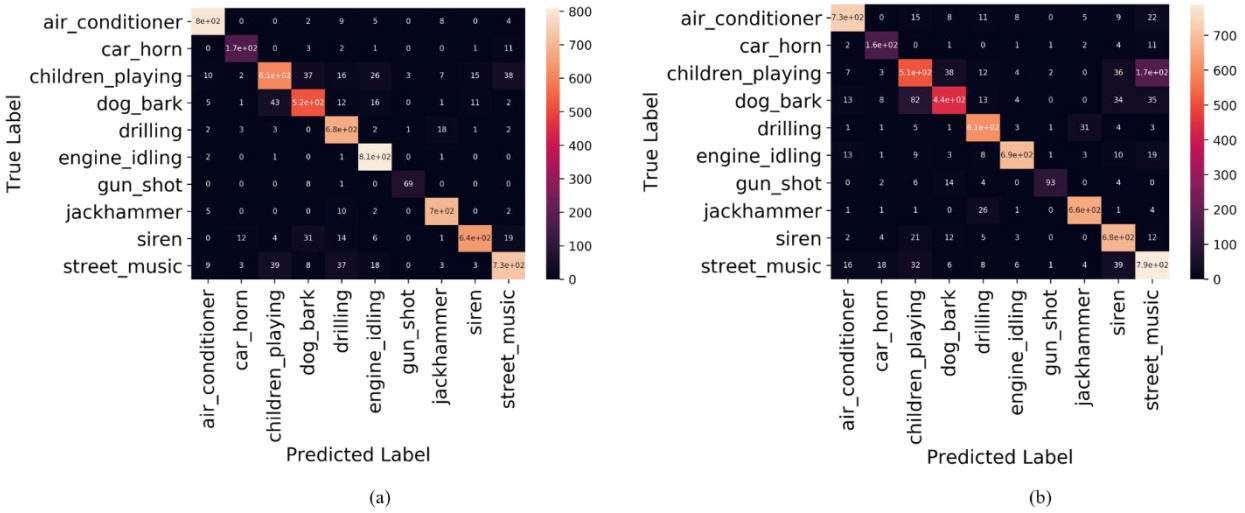

Fig. 4.15 Best and poorest confusion matrices for 5-fold cross validation on US-8K dataset. (a)-(b) represents the output for fold-4 and fold-1 respectively for proposed 2D CNN network.

Table 4.11. Classification report of fold-1 for US-8K validation dataset (proposed 2D CNN).

Class	Precision	Recall	F_1-score	Accuracy	MCC	CKC
AI	93%	90%	0.92	85%	0.83	0.83
CH	81%	88%	0.84			
CP	75%	65%	0.70			
DB	84%	70%	0.76			
DR	88%	92%	0.90			
EI	96%	91%	0.93			
GS	94%	76%	0.84			
JH	94%	95%	0.94			
SI	83%	92%	0.87			
SM	74%	86%	0.80			

Table 4.12 Classification report of fold-2 for US-8K validation dataset (proposed 2D CNN).

Class	Precision	Recall	F_1-score	Accuracy	MCC	CKC
AI	96%	91%	0.94	89%	0.88	0.88

CH	89%	88%	0.88			
CP	81%	82%	0.81			
DB	84%	81%	0.82			
DR	90%	90%	0.90			
EI	96%	96%	0.96			
GS	95%	84%	0.89			
JH	95%	94%	0.94			
SI	84%	97%	0.90			
SM	88%	80%	0.84			

Table 4.13 Classification report of fold-3 for US-8K validation dataset (proposed 2D CNN).

Class	**Precision**	**Recall**	**F_1-score**	**Accuracy**	**MCC**	**CKC**
AI	94%	96%	0.95	89%	0.88	0.88
CH	91%	82%	0.86			
CP	86%	73%	0.79			
DB	72%	87%	0.79			
DR	92%	93%	0.93			
EI	95%	96%	0.96			
GS	94%	81%	0.87			
JH	93%	97%	0.95			
SI	97%	85%	0.91			
SM	85%	88%	0.87			

Table 4.14 Classification report of fold-4 for US-8K validation dataset (proposed 2D CNN).

Class	**Precision**	**Recall**	**F_1-score**	**Accuracy**	**MCC**	**CKC**
AI	96%	97%	0.96	91%	0.90	0.90

CH	89%	90%	0.90			
CP	87%	80%	0.83			
DB	85%	85%	0.85			
DR	87%	95%	0.91			
EI	91%	99%	0.95			
GS	95%	88%	0.91			
JH	95%	97%	0.96			
SI	95%	88%	0.92			
SM	90%	86%	0.88			

Table 4.15 Classification report of fold-5 for US-8K validation dataset (proposed 2D CNN).

Class	Precision	Recall	F_1-score	Accuracy	MCC	CKC
AI	93%	96%	0.94	91%	0.90	0.90
CH	97%	92%	0.94			
CP	83%	85%	0.84			
DB	80%	89%	0.84			
DR	91%	94%	0.92			
EI	98%	99%	0.98			
GS	95%	84%	0.89			
JH	93%	96%	0.95			
SI	97%	92%	0.94			
SM	91%	78%	0.84			

Table 4.16 Classification report of fold-1 for ESC-10 validation dataset (proposed 2D CNN).

Class	Precision	Recall	F_1-score	Accuracy	MCC	CKC
CS	94%	97%	0.96	82%	0.81	0.81

CT	97%	69%	0.81			
CF	92%	90%	0.91			
CB	88%	84%	0.86			
DB	86%	52%	0.65			
HL	64%	100%	0.78			
RA	92%	84%	0.88			
RO	84%	44%	0.58			
SW	96%	94%	0.95			
SN	64%	89%	0.74			

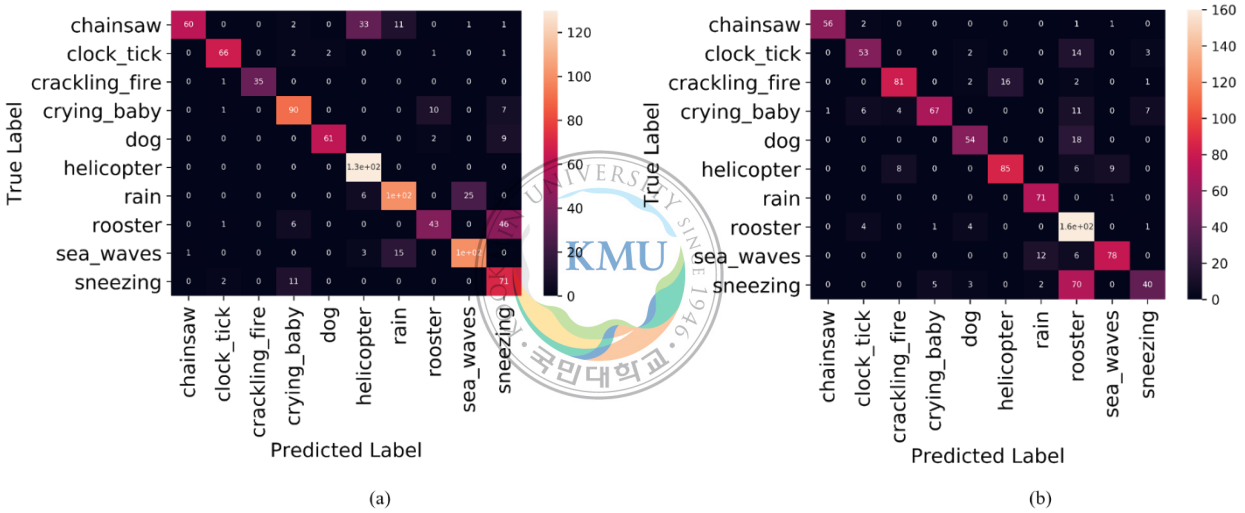

Fig. 4.16 Best and poorest confusion matrices for 5-fold cross validation on ESC-10 dataset. (a)-(b) represents the output for fold-2 and fold-5 respectively for proposed 2D CNN network.

Table 4.17 Classification report of fold-2 for ESC-10 validation dataset (proposed 2D CNN).

Class	Precision	Recall	F_1-score	Accuracy	MCC	CKC
CS	98%	56%	0.71	84%	0.77	0.77
CT	93%	92%	0.92			
CF	100%	97%	0.99			
CB	81%	83%	0.82			

DB	97%	85%	0.90			
HL	76%	100%	0.86			
RA	80%	77%	0.78			
RO	77%	45%	0.57			
SW	80%	84%	0.82			
SN	53%	85%	0.65			

Table 4.18 Classification report of fold-3 for ESC-10 validation dataset (proposed 2D CNN).

Class	Precision	Recall	F_1-score	Accuracy	MCC	CKC
CS	98%	86%	0.92	79%	0.77	0.76
CT	83%	90%	0.87			
CF	91%	80%	0.85			
CB	88%	78%	0.82			
DB	94%	68%	0.79			
HL	86%	85%	0.85			
RA	67%	100%	0.80			
RO	77%	45%	0.57			
SW	97%	77%	0.86			
SN	34%	90%	0.52			

Table 4.19 Classification report of fold-4 for ESC-10 validation dataset (proposed 2D CNN).

Class	Precision	Recall	F_1-score	Accuracy	MCC	CKC
CS	90%	73%	0.80	79%	0.75	0.75
CT	91%	72%	0.80			
CF	72%	68%	0.70			
CB	76%	95%	0.85			

DB	89%	63%	0.74			
HL	90%	79%	0.84			
RA	74%	89%	0.81			
RO	92%	46%	0.61			
SW	82%	94%	0.87			
SN	55%	83%	0.66			

Table 4.20 Classification report of fold-5 for ESC-10 validation dataset (proposed 2D CNN).

Class	**Precision**	**Recall**	**F_1-score**	**Accuracy**	**MCC**	**CKC**
CS	98%	93%	0.96			
CT	92%	74%	0.77			
CF	87%	79%	0.83			
CB	92%	70%	0.79			
DB	83%	75%	0.79	77%	0.75	0.74
HL	84%	79%	0.81			
RA	84%	99%	0.90			
RO	55%	94%	0.70			
SW	88%	81%	0.84			
SN	77%	33%	0.47			

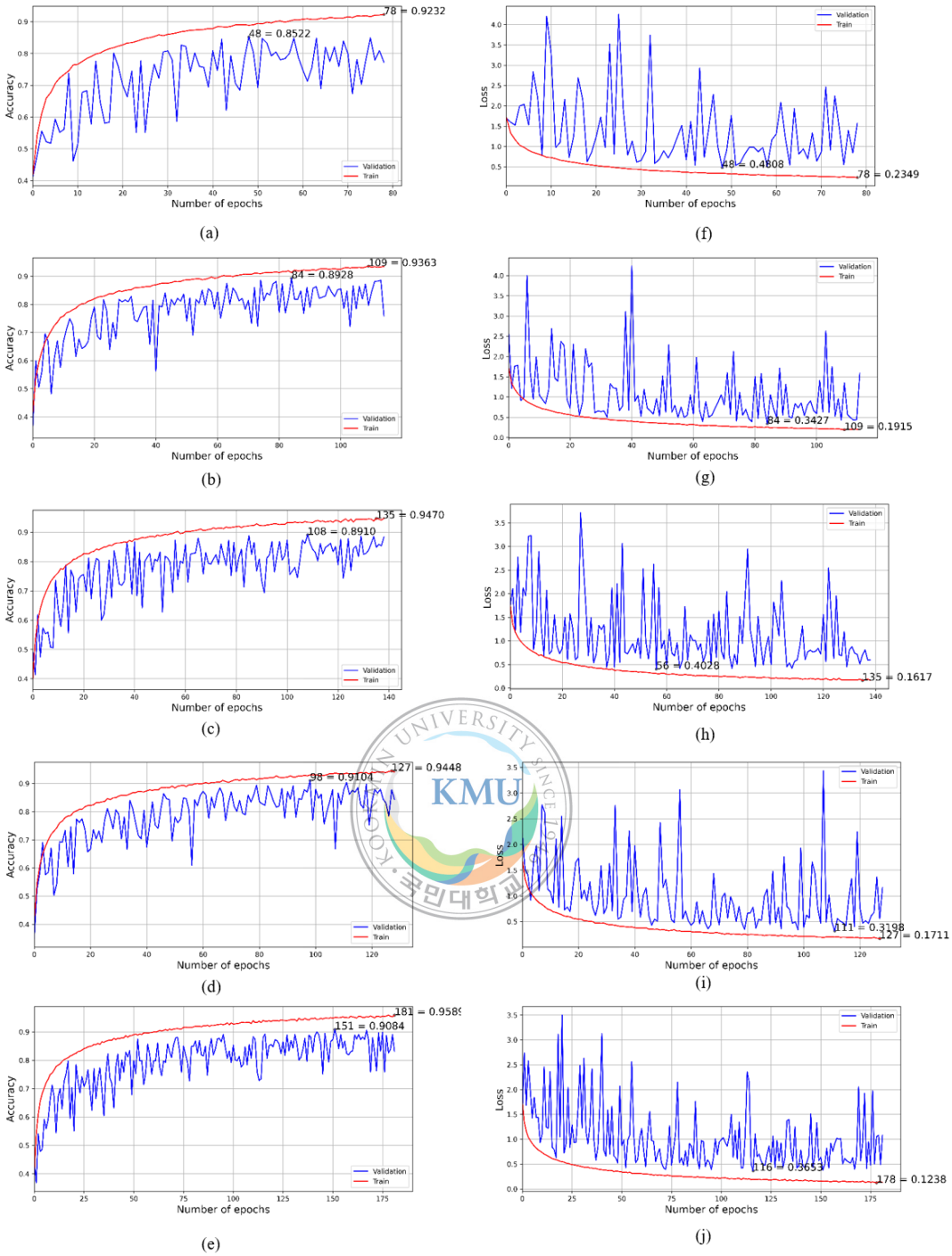

Fig. 4.17 Learning curves (accuracy and loss per epoch) for Gammatone spectrogram input of US-8K dataset to our proposed 1D CNN network. The graphs for 5-fold cross validation are shown: (a)-(e) accuracy per epoch for fold 1-5, and (f)-(j) loss per epoch for fold 1-5. The best accuracy and minimum loss are marked in the respective figure.

Fig. 4.18 Learning curves (accuracy and loss per epoch) for Gammatone spectrogram input of ESC-10 dataset to our proposed 2D CNN network. The graphs for 5-fold cross validation are

shown: (a)-(e) accuracy per epoch for fold 1-5, and (f)-(j) loss per epoch for fold 1-5. The best accuracy and minimum loss are marked in the respective figure.

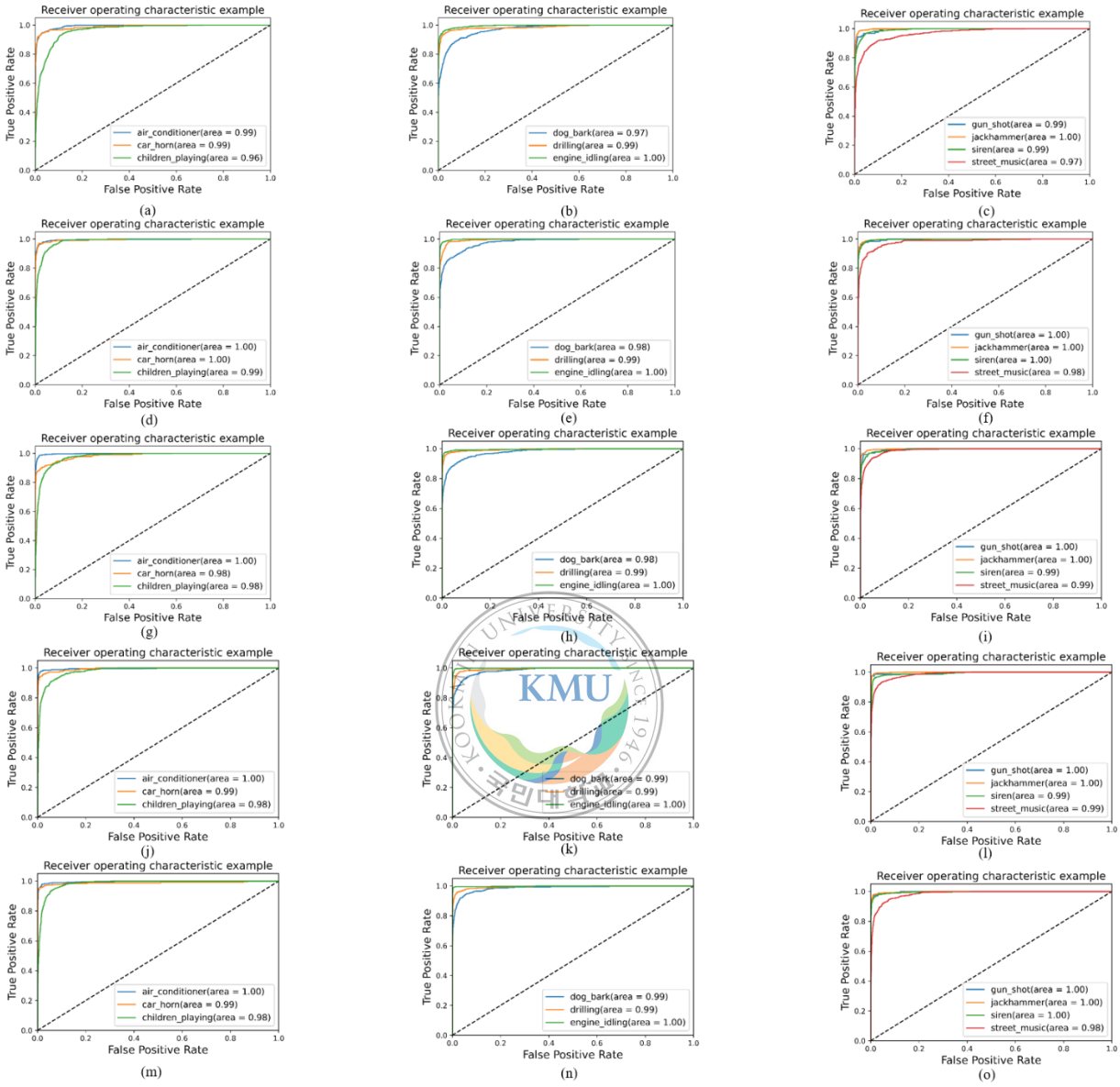

Fig. 4.19 ROC curve and respective AUC for Gammatone spectrogram input of US-8K dataset to our proposed 2D CNN network. The graphs for 5-fold cross validation are shown. Each row represents the ROC and AUC of 10-class for CV.

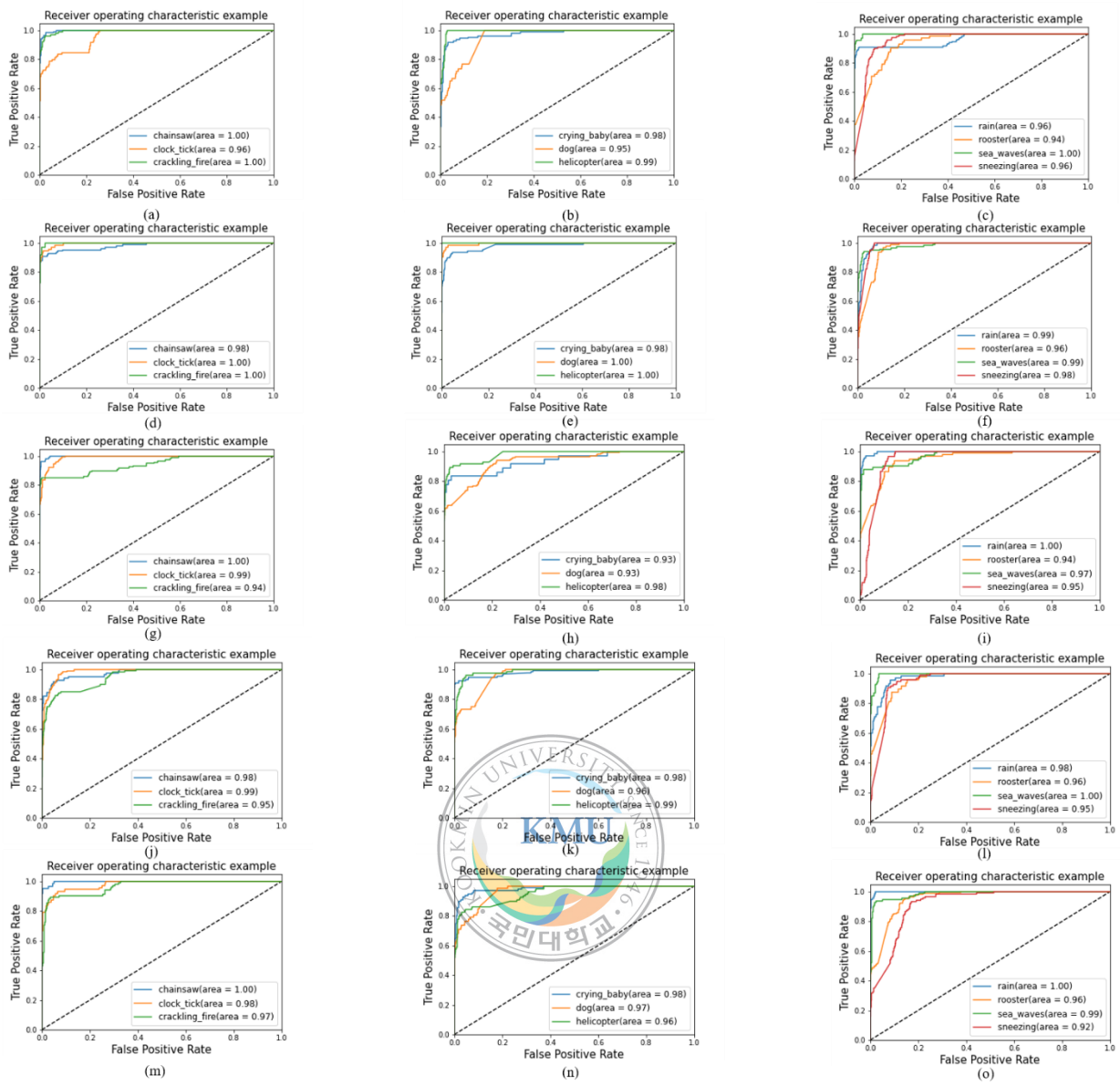

Fig. 4.20 ROC curve and respective AUC for Gammatone spectrogram input of ESC-10 dataset to our proposed 2D CNN network. The graphs for 5-fold cross validation are shown. Each row represents the ROC and AUC of 10-class for 5-fold cross validation.

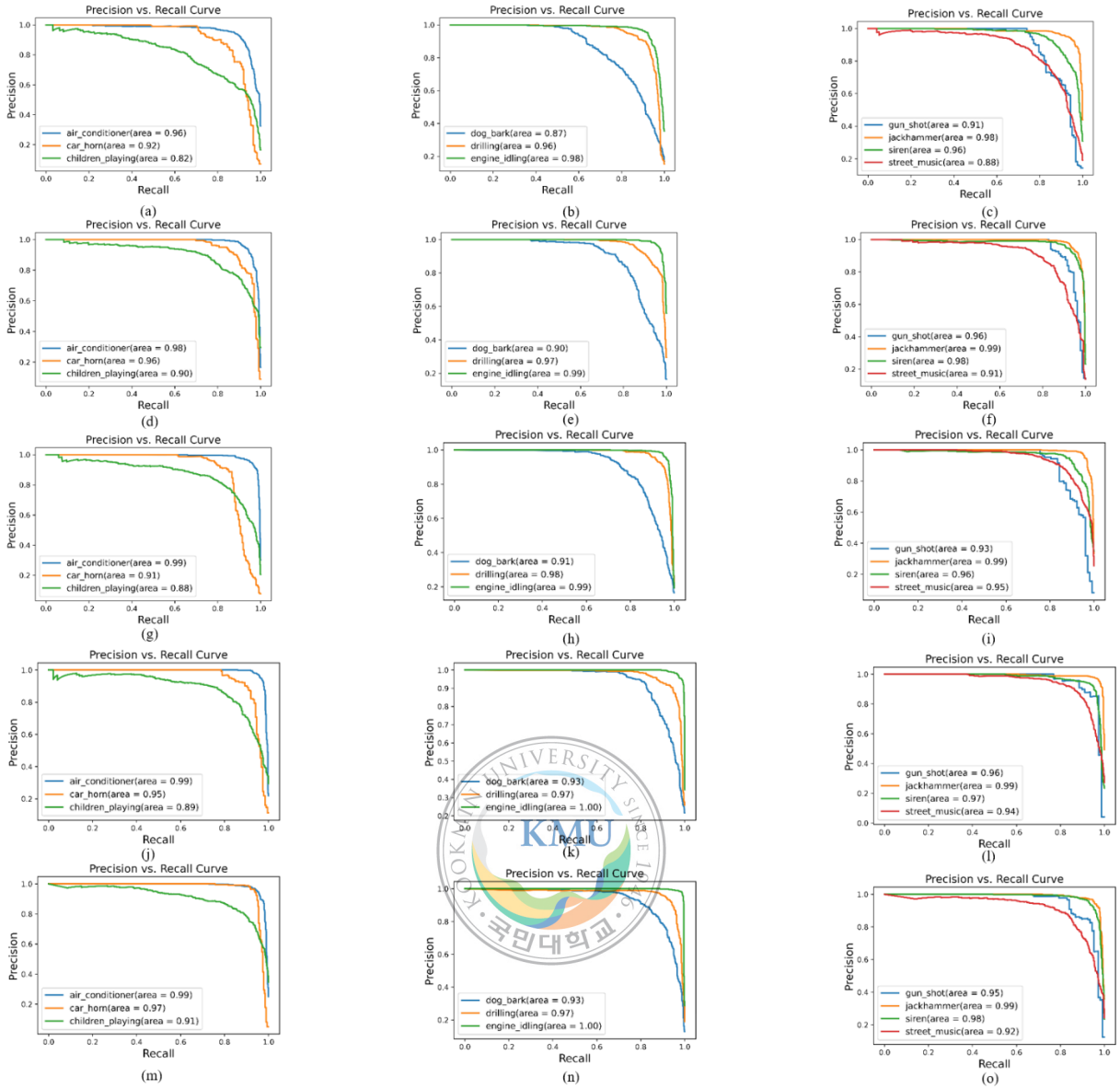

Fig. 4.21 Precision vs. Recall (PR) curve and respective area under curve (AUC) for gammatone spectrogram input of US-8K dataset to our proposed 2D CNN network. The graphs for 5-fold cross validation are shown. Each row represents the PR and AUC of 10-class for 5-fold cross validation.

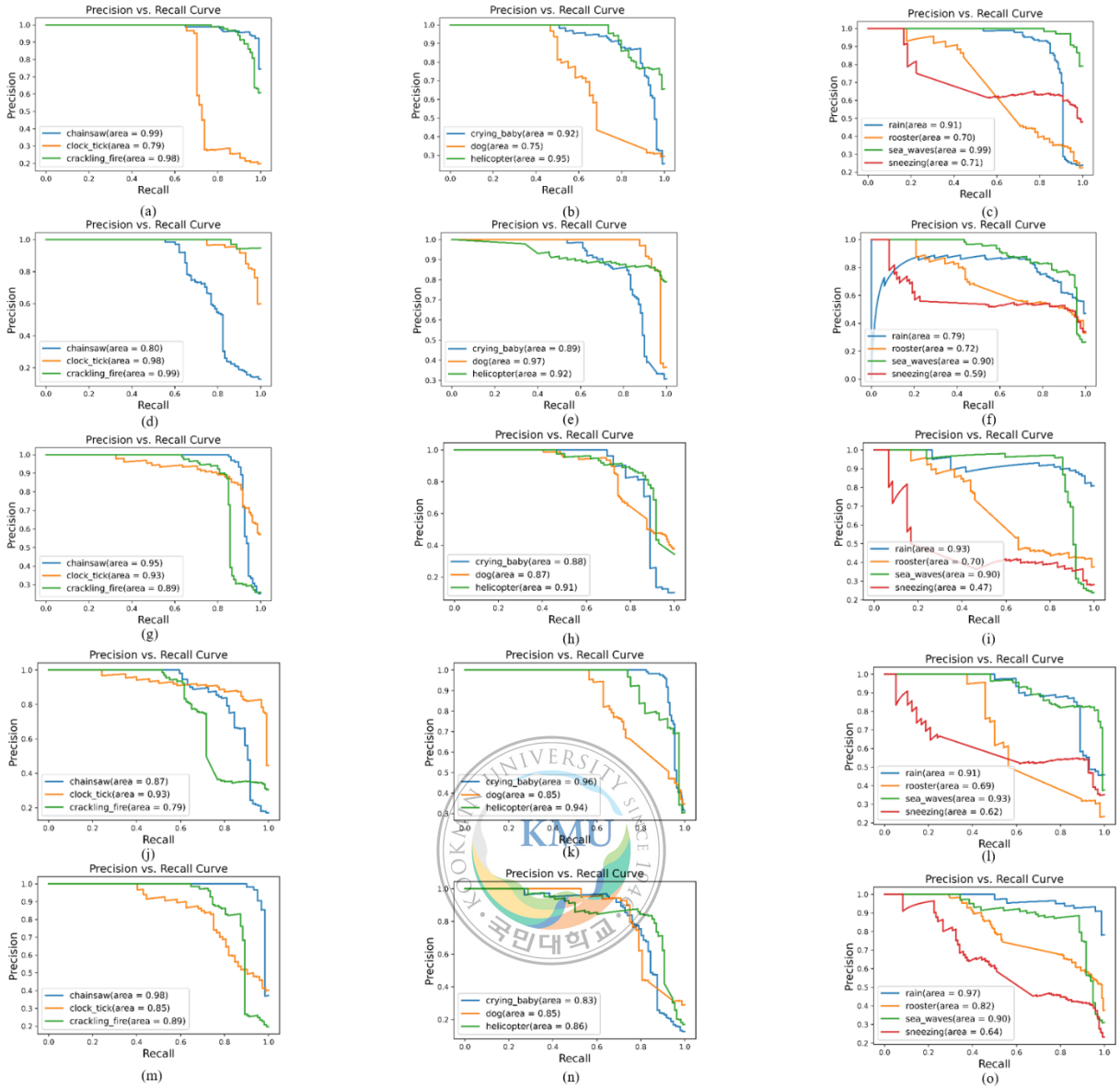

Fig. 4.22 Precision vs. Recall (PR) curve and respective area under curve (AUC) for gammatone spectrogram input of ESC-10 dataset to our proposed 2D CNN network. The graphs for 5-fold cross validation are shown. Each row represents the PR and AUC of 10-class for 5-fold cross validation.

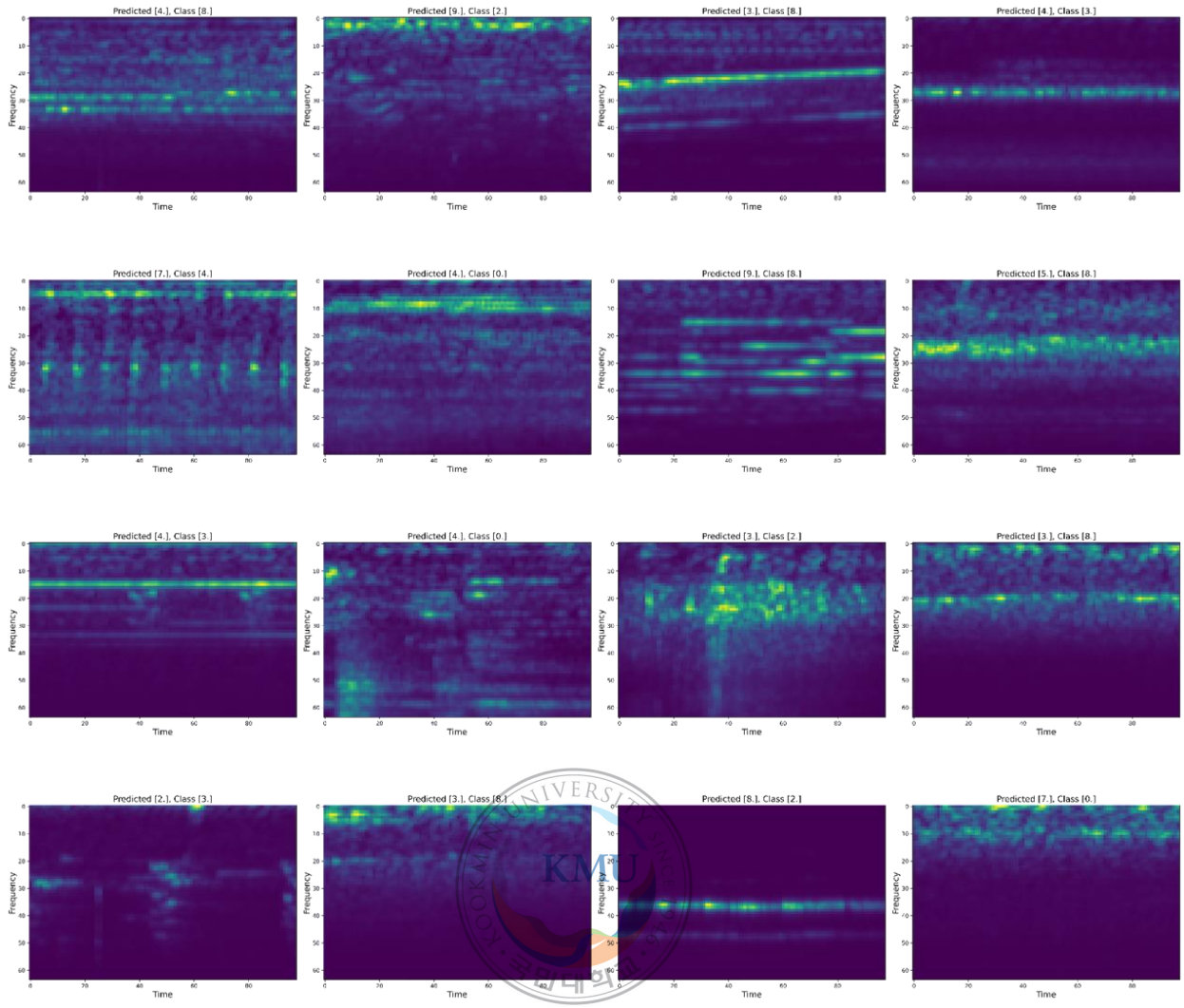

Fig. 4.23 Some misclassified classes by our proposed 2D network tested on US-8K dataset.

4.5 Overall performance analysis

In previous sections, the results are analyzed by evaluating various performance matrices. To visualize the overall classification accuracy and loss for both datasets and models as a boxplot, Fig. 4.25 would be quite helpful. Here, it can be seen from Fig. 4.25(a) that the classification accuracy is highest (91%) for 2D CNN on US-8K dataset. The lowest accuracy is for 1D CNN on ESC-10 dataset. The accuracy for 1D and 2D CNN on ESC-10 dataset is kind of same. This same nature is also observed in classification loss plot at Fig. 4.25(b). However, in general the proposed 1D and 2D CNN predicts superior for US-8K dataset than for ESC-10 dataset. One possible reason of it may be lack of supporting example per class in ESC-10. To improve the

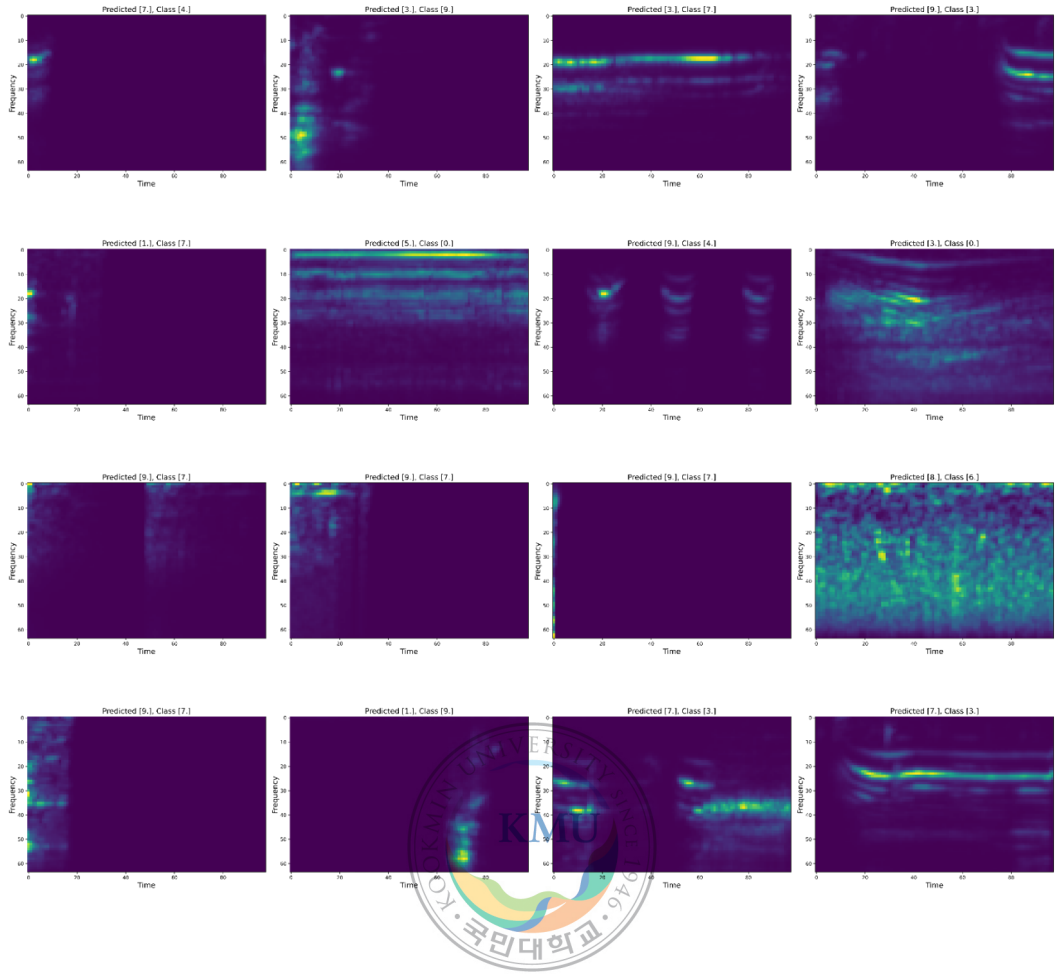

Fig. 4.24 Some misclassified classes by our proposed 2D network tested on ESC-10 dataset.

number of training example, data augmentation attempt has been taken. Otherwise the result would degrade more. Table 4.21 enlists the comparison of the proposed models here against other state-of-the art methods. The methods are arranged in descending order in terms of accuracy. Among 2D input representation methods, proposed 2D CNN model is competitive with a mean accuracy of 89%. Comparing with another recent 1D CNN based method [27], proposed 1D model's accuracy is very close to them. Considering number of less model parameters, proposed models are ahead of many other approaches. However, to measure the computational load on hardware, floating point operations per second (FLOPS) is also calculated. The highest FLOPS for 2D CNN model is found around 6.96 Giga and for 1D CNN it is around 0.163G. Measuring all the aspects of the proposed models, it can be confidently

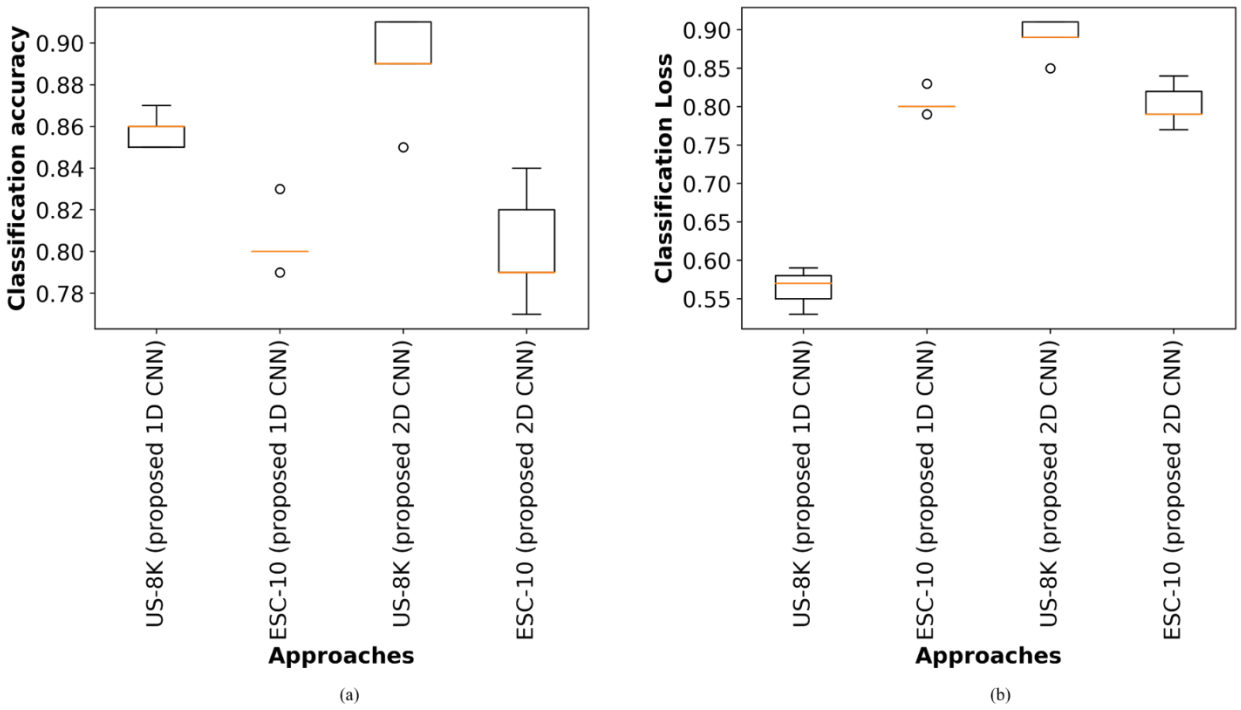

Fig. 4.25 5-fold cross validation boxplot for proposed 1D and 2D CNN architecture: overall classification (a) accuracy, and (b) loss for both datasets (US-8K and ESC-10). Orange line means median value here.

Table 4.21 Different approaches' vs. proposed model's mean accuracy on US-8K dataset.

Methods	Input shape	Mean accuracy	# of Parameters
TSCNN-DS [53]	2D	**97%**	15.9 million
GoogleNet [34]	2D	93%	6.7 million
DS-CNN [33]	1D-2D	92%	NA
AbdoliCNN (GT) [27]	1D	89%	0.55 million
Proposed 2D CNN	2D	**89%**	**1.8 million**
AdboliCNN [27]	1D	87%	0.256 million
Proposed 1D CNN	1D	**86%**	**1.6 million**
EnvNet-v2 [26]	1D	78%	101 million
PiczakCNN [19]	2D	73%	26 million

SB-CNN [21]	2D	73%	**0.241 million**
M18CNN [23]	1D	72%	3.7 million

⁎ GT: Gammatone

• Best values for mean accuracy and no. parameters are in bold font. The values for proposed models are in blue colored bold font.

said that the proposed model is quite competitive with other methods. Moreover, another interesting and useful point is that very close accuracy to 2D CNN can be found using the advised 1D CNN model which requires no heavily engineered pre-processing steps and has also less computation burden on hardware.

Chapter 5

Conclusion

5.1 Summary

In this research, the main motivation was to develop a 1D CNN model for environmental sound classification which does not require any preprocessing. For this purpose, another 2D CNN model was proposed to classify the signal from their given Gammatone spectrogram. After comparing the two models on same datasets but for different input representation, it is decided by means of accuracy that the 1D CNN model can learn the discriminative features from the raw signal input. In all quality metrics, 1D CNN model performs more or less same as 2D CNN model. Moreover, 1D representation saves heavy pre-processing and by this way computational cost. Both CNN architectures (1D and 2D) performs close to state-of-the-art methods. Moderate parameters and low FLOPS make sure the faster inference on the test or real world dataset. However, to achieve this type of accuracy and performance, effective data augmentation methods were used. Other techniques, related to this ESC task were also discussed in detail here specifically how filter bank representation of input signal is more related to the human ear architecture and how CNN is related to the human auditory system are broadly discussed. Overall, a logical, theoretical, and empirical analysis suggests the compatibility of this research's outcome with many other prior methods.

5.2 Failures

Although the suggested approach and models succeeded in maximum cases, sometimes some failure cases also occurred. Some sounds are so similar to one to another sometimes human also are misled. So, the proposed model also found it tough to find discriminative features from the given representation. Another important point to note here that the applied data augmentation procedure is not enough. The dataset size of ESC-10 is basically small and after augmentation the size did not become necessarily large.

5.3 Future research

The presented research leaves a room for improvement in several sections. First of all, effective representation of input features is so important. If the input is in raw waveform, then efficient kern el design is necessary inside the CNN model. So, this is one type of plan that will be implemented in future. Secondly, due to data scarcity data augmentation is really a necessary step. So, in future an attempt will be taken to implement other types of augmentation which are related to real world scenario. At last, more effort will be given to design the model size compact so that it will fit to the low-end processor device for IOT application.

References

[1] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, "Deep Learning for Audio Signal Processing," *IEEE J. Sel. Top. Signal Process.*, vol. 13, no. 2, pp. 206–219, May 2019, doi: 10.1109/JSTSP.2019.2908700.

[2] S. Chachada and C.-J. Kuo, "Environmental sound recognition: A survey," in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, Oct. 2013, pp. 1–9, doi: 10.1109/APSIPA.2013.6694338.

[3] M. Crocco, M. Cristani, A. Trucco, and V. Murino, "Audio Surveillance: A Systematic Review," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 1–46, May 2016, doi: 10.1145/2871183.

[4] Y. LeCun *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989, doi: 10.1162/neco.1989.1.4.541.

[5] G. Hinton *et al.*, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012, doi: 10.1109/MSP.2012.2205597.

[6] C. J. Plack, *The sense of hearing, 2nd ed*. New York, NY, US: Psychology Press, 2014, pp. x, 296.

[7] J. B. Allen, "How do humans process and recognize speech?," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 567–577, Oct. 1994, doi: 10.1109/89.326615.

[8] A. J. E. Kell, D. L. K. Yamins, E. N. Shook, S. V. Norman-Haignere, and J. H. McDermott, "A Task-Optimized Neural Network Replicates Human Auditory Behavior, Predicts Brain Responses, and Reveals a Cortical Processing Hierarchy," *Neuron*, vol. 98, no. 3, pp. 630-644.e16, May 2018, doi: 10.1016/j.neuron.2018.03.044.

[9] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Feature Learning With Matrix Factorization Applied to Acoustic Scene Classification," *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 6, pp. 1216–1229, Jun. 2017, doi: 10.1109/TASLP.2017.2690570.

[10] S. Chu, S. Narayanan, and C.-C. J. Kuo, "Environmental sound recognition with time-frequency audio features," *IEEE Trans. Audio Speech Lang. Process.*, vol. 17, no. 6, pp. 1142–1158, Aug. 2009.

[11] P. Dhanalakshmi, S. Palanivel, and V. Ramalingam, "Classification of audio signals using AANN and GMM," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 716–723, Jan. 2011, doi: 10.1016/j.asoc.2009.12.033.

[12] J. T. Geiger and K. Helwani, "Improving event detection for audio surveillance using Gabor filterbank features," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, Aug. 2015, pp. 714–718, doi: 10.1109/EUSIPCO.2015.7362476.

[13] J. Wang, H. Lee, J. Wang, and C. Lin, "Robust Environmental Sound Recognition for Home Automation," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 1, pp. 25–31, Jan. 2008, doi: 10.1109/TASE.2007.911680.

[14] J. Wang, Y. Lee, C. Lin, E. Siahaan, and C. Yang, "Robust Environmental Sound Recognition With Fast Noise Suppression for Home Automation," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1235–1242, Oct. 2015, doi: 10.1109/TASE.2015.2470119.

[15] B. Ghoraani and S. Krishnan, "Time–Frequency Matrix Feature Extraction and Classification of Environmental Audio Signals," *IEEE Trans. Audio Speech Lang. Process.*, vol. 19, no. 7, pp. 2197–2209, Sep. 2011, doi: 10.1109/TASL.2011.2118753.

[16] E. Akbal, "An automated environmental sound classification methods based on statistical and textural feature," *Appl. Acoust.*, vol. 167, p. 107413, Oct. 2020, doi: 10.1016/j.apacoust.2020.107413.

[17] S. Chandrakala and S. L. Jayalakshmi, "Generative Model Driven Representation Learning in a Hybrid Framework for Environmental Audio Scene and Sound Event Recognition," *IEEE Trans. Multimed.*, vol. 22, no. 1, pp. 3–14, Jan. 2020, doi: 10.1109/TMM.2019.2925956.

[18] X. Valero and F. Alias, "Gammatone Cepstral Coefficients: Biologically Inspired Features for Non-Speech Audio Classification," *IEEE Trans. Multimed.*, vol. 14, no. 6, pp. 1684–1689, Dec. 2012, doi: 10.1109/TMM.2012.2199972.

[19] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep. 2015, pp. 1–6, doi: 10.1109/MLSP.2015.7324337.

[20] J. Pons and X. Serra, "Randomly weighted CNNs for (music) audio classification," *ArXiv180500237 Cs Eess*, Feb. 2019, Accessed: Feb. 10, 2020. [Online]. Available: http://arxiv.org/abs/1805.00237.

[21] J. Salamon and J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 279–283, Mar. 2017, doi: 10.1109/LSP.2017.2657381.

[22] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 4624–4628, doi: 10.1109/ICASSP.2015.7178847.

[23] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 421–425, doi: 10.1109/ICASSP.2017.7952190.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *ArXiv151203385 Cs*, Dec. 2015, Accessed: Nov. 29, 2020. [Online]. Available: http://arxiv.org/abs/1512.03385.

[25] 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)*, Mar. 2017, pp. 2721–2725, doi: 10.1109/ICASSP.2017.7952651.

[26] Y. Tokozume, Y. Ushiku, and T. Harada, “Learning from Between-class Examples for Deep Sound Recognition,” *ArXiv171110282 Cs Eess Stat*, Feb. 2018, Accessed: Nov. 29, 2020. [Online]. Available: http://arxiv.org/abs/1711.10282.

[27] S. Abdoli, P. Cardinal, and A. Lameiras Koerich, “End-to-end environmental sound classification using a 1D convolutional neural network,” *Expert Syst. Appl.*, vol. 136, pp. 252–263, Dec. 2019, doi: 10.1016/j.eswa.2019.06.040.

[28] H. Park and C. D. Yoo, “CNN-Based Learnable Gammatone Filterbank and Equal-Loudness Normalization for Environmental Sound Classification,” *IEEE Signal Process. Lett.*, vol. 27, pp. 411–415, 2020, doi: 10.1109/LSP.2020.2975422.

[29] É. Bavu, A. Ramamonjy, H. Pujol, and A. Garcia, “TimeScaleNet: A Multiresolution Approach for Raw Audio Recognition Using Learnable Biquadratic IIR Filters and Residual Networks of Depthwise-Separable One-Dimensional Atrous Convolutions,” *IEEE J. Sel. Top. Signal Process.*, vol. 13, no. 2, pp. 220–235, May 2019, doi: 10.1109/JSTSP.2019.2908696.

[30] Y. Aytar, C. Vondrick, and A. Torralba, “SoundNet: learning sound representations from unlabeled video,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, Dec. 2016, pp. 892–900, Accessed: Dec. 01, 2020. [Online].

[31] A. Khamparia, D. Gupta, N. G. Nguyen, A. Khanna, B. Pandey, and P. Tiwari, “Sound Classification Using Convolutional Neural Network and Tensor Deep Stacking Network,” *IEEE Access*, vol. 7, pp. 7717–7727, 2019, doi: 10.1109/ACCESS.2018.2888882.

[32] Y. Su, K. Zhang, J. Wang, and K. Madani, “Environment Sound Classification Using a Two-Stream CNN Based on Decision-Level Fusion,” *Sensors*, vol. 19, no. 7, Art. no. 7, Jan. 2019, doi: 10.3390/s19071733.

[33] S. Li, Y. Yao, J. Hu, G. Liu, X. Yao, and J. Hu, “An Ensemble Stacked Convolutional Neural Network Model for Environmental Event Sound Recognition,” *Appl. Sci.*, vol. 8, no. 7, p. 1152, Jul. 2018, doi: 10.3390/app8071152.

[34] V. Boddapati, A. Petef, J. Rasmusson, and L. Lundberg, “Classifying environmental sounds using image recognition networks,” *Procedia Comput. Sci.*, vol. 112, pp. 2048–2056, Jan. 2017, doi: 10.1016/j.procs.2017.08.250.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[36] C. Szegedy *et al.*, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[37] Z. Zhang, S. Xu, S. Zhang, T. Qiao, and S. Cao, “Attention based convolutional recurrent neural network for environmental sound classification,” *Neurocomputing*, Sep. 2020, doi: 10.1016/j.neucom.2020.08.069.

[38] 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)*, Apr. 2015, pp. 171–175, doi: 10.1109/ICASSP.2015.7177954.

[39] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd ACM international conference on Multimedia*, Brisbane, Australia, Oct. 2015, pp. 1015–1018, doi: 10.1145/2733373.2806390.

[40] J. Salamon, C. Jacoby, and J. P. Bello, "A Dataset and Taxonomy for Urban Sound Research," in *Proceedings of the 22nd ACM international conference on Multimedia*, Orlando, Florida, USA, Nov. 2014, pp. 1041–1044, doi: 10.1145/2647868.2655045.

[41] J. Gibson, M. V. Segbroeck, and S. S. Narayanan, "Comparing Time-Frequency Representations for Directional Derivative Features," p. 4.

[42] R. Patterson, I. Nimmo-smith, J. Holdsworth, P. Rice, and M. Qoad, *Cambride,e CB2 2EF*. 1987.

[43] B. R. Glasberg and B. C. J. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hear. Res.*, vol. 47, no. 1, pp. 103–138, Aug. 1990, doi: 10.1016/0378-5955(90)90170-T.

[44] "Gammatone-like spectrograms." https://www.ee.columbia.edu/~dpwe/resources/matlab/gammatonegram/ (accessed Nov. 25, 2020).

[45] J. L. Flanagan and R. M. Golden, "Phase vocoder," *Bell Syst. Tech. J.*, vol. 45, no. 9, pp. 1493–1509, Nov. 1966, doi: 10.1002/j.1538-7305.1966.tb01706.x.

[46] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *ArXiv150203167 Cs*, Mar. 2015, Accessed: Dec. 07, 2020. [Online]. Available: http://arxiv.org/abs/1502.03167.

[47] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *ArXiv14126980 Cs*, Jan. 2017, Accessed: Dec. 07, 2020. [Online]. Available: http://arxiv.org/abs/1412.6980.

[48] "Optimizers — ML Glossary documentation." https://ml-cheatsheet.readthedocs.io/en/latest/optimizers.html#adam (accessed Dec. 07, 2020).

[49] B. McFee *et al.*, "librosa: Audio and Music Signal Analysis in Python," Austin, Texas, 2015, pp. 18–24, doi: 10.25080/Majora-7b98e3ed-003.

[50] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011.

[51] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochim. Biophys. Acta BBA - Protein Struct.*, vol. 405, no. 2, pp. 442–451, Oct. 1975, doi: 10.1016/0005-2795(75)90109-9.

[52] "Receiver operating characteristic," *Wikipedia*. Nov. 19, 2020, Accessed: Nov. 28, 2020. [Online]. Available:

https://en.wikipedia.org/w/index.php?title=Receiver_operating_characteristic&oldid=989528442

[53] Y. Su, K. Zhang, J. Wang, and K. Madani, “Environment Sound Classification Using a Two-Stream CNN Based on Decision-Level Fusion,” *Sensors*, vol. 19, no. 7, Apr. 2019, doi: 10.3390/s19071733.