AUDIOCNN: AUDIO EVENT CLASSIFICATION WITH DEEP LEARNING BASED MULTI-CHANNEL FUSION NETWORKS

A Thesis

in

Computer Science

Presented to the Faculty of the University of
Missouri–Kansas City in partial fulfillment
of the requirements for the degree
MASTER OF SCIENCE

by

Nagababu Chilukuri

B. Tech., JNTUK-University College of Engineering, Vizianagaram, India, 2018

Kansas City, Missouri

2020

AUDIOCNN: AUDIO EVENT CLASSIFICATION WITH DEEP LEARNING BASED MULTI-CHANNEL FUSION NETWORKS

Nagababu Chilukuri, Candidate for the Master of Science Degree
University of Missouri-Kansas City, 2020

ABSTRACT

In recent years, there is growing interest in environmental sound classification with a plethora of real-world applications, especially in audio fields like speech and music. Recent research works have proven spectral images based on deep learning models for better performance than standard methods. This thesis intends to design a fusion system by combining various audio features, including Spectrogram (SG), Chromagram (CG), and Mel Frequency Cepstral Coefficient (MFCC), for useful environmental sound classification. We propose the AudioCNN model based on a fusion network consisting of multiple Convolutional Neural Networks (CNN) with aggregation methods for various spectral image spectrogram features and audio-specific data augmentation techniques. We have conducted our extensive experiments with benchmark datasets, including Urbansound8k, ESC-50, and ESC-10, emotion datasets. We have obtained state-of-the-art results by outperforming the previous solutions. The experiment results show that combined features with lighter network CNN models outperform baseline environmental sound classification methods. The proposed Multi-Channel fusion network with data augmentation achieved competitive results on UrbanSound8K datasets compared to existing models.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Computing and Engineering, have examined a thesis titled "AudioCNN: Audio Event Classification with Deep Learning Based Multi-Channel Fusion Networks," presented by Nagababu Chilukuri, candidate for the Master of Science degree, and certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Yugyung Lee, Ph.D. (Committee Chair)
Department of Computer Science Electrical Engineering

Cory Beard, Ph.D.
Department of Computer Science Electrical Engineering

Ye Wang, Ph.D.
Department of Communication Studies

Content

ABSTRACT…… iii

ILLUSTRATIONS…….. vii

TABLES…….. viii

CHAPTER

1. INTRODUCTION……..…. 1

2. BACKGROUND…….……..…….. 4

2.1. Deep Learning……..... 4

2.2. Convolutional Neural Networks……..……... 5

2.2.1. Convolutional Layer……...… 6

2.2.2. Pooling Layer……..……..…. 7

2.3. Activation Functions……....….. 8

2.4. Transfer Learning……..……..… 11

2.5. Data Augmentation…….……...… 13

3. RELATED WORK……...…..… 17

3.1. Overview……..….. 17

3.2. Related Approaches……...……...…….. 18

4. METHODOLOGY…….. 20

4.1. Overview……..…..……..……...…… 20

4.2. Data Collection and Pre-Processing ……..……..… 21

4.3. Feature Selection…..…….……..… 23

4.4. Model Architectures…….……...… 27

4.5. Feature Fusion…… 29

5. RESULTS AND EVALUATION……...…… 31

5.1. Introduction…… 31

5.2. Evaluation Setup……..…….……..…… 31

5.3. Evaluation Metrics…….…… 31

5.4. Results and Discussions…….…….….. 32

6. CONCLUSION…….. 42

BIBLIOGRAPHY…… 43

VITA…….…… 46

ILLUSTRATIONS

Figure 1 Neural Network Architecture 5

Figure 2 Convoluted Feature Output with Kernel Filter ……..…….7

Figure 3 Max Pooling……....7

Figure 4 Different Activation Functions……..…...…….……..8

Figure 5 Overview of CNN Architecture with Many Convolutional Layers……..10

Figure 6 Traditional Image Data Augmentation Approach……..……...….14

Figure 7 Spectrogram-Based Image Data Augmentation Approach……...….....14

Figure 8 Noise Injection Example……..…..….15

Figure 9 Change In Time Shift Example……..……....15

Figure 10 Change In Pitch & Speed Example……..16

Figure 11 Overall Proposed Framework……..20

Figure 12 Urban Sound Taxonomy of UrbanSound8k Dataset……..…...…….22

Figure 13 The Three Types of Extracted Features…….…….……23

Figure 14 An Illustration of a Signal, after Multiplying with Overlapped Windows....24

Figure 15 A 12-Dimensional Feature Vector Representation of An Audio Signal…..26

Figure 16 The Architecture of Proposed CNN Model……..28

Figure 17 The Overall Framework of the Multi-Channel Fusion Network…….….30

Figure 18 Confusion Matrix…….32

Figure 19 Training Accuracy vs. Validation Accuracy of CNN Model……..33

Figure 20 Confusion Matrix Evaluated on the UrbanSound8K Dataset..……....35

Figure 21 Confusion Matrix Evaluated on the Emotion Dataset……..……..…….37

Figure 22 Training and Validation Accuracies with Transfer Learning……..……38

TABLES

Table 1 Information of Datasets 22

Table 2 Comparison of Different Input Feature Combinations on Us8k Dataset……34

Table 3 Comparison of Different Input Feature Combinations on ESC-50 Dataset...36

Table 4 Comparison of Different Input Feature Combinations on ESC-10 Dataset….36

Table 5 Information of Augmented Datasets……..…….……39

Table 6 Comparison of the Proposed Approach with Existing Models…...……...40

ACKNOWLEDGMENTS

I would like to thank Dr. Yugyung Lee for her valuable guidance and immense support throughout the research work as my advisor. Data analytics is growing very fast. Dr. Lee always keeps herself up to date with the latest research and encourages her students to work towards cutting-edge technologies. I am amazed by her positive energy and patience. Her vast experience, unparalleled knowledge, agile and prompt feedback, and smart ideas have immensely helped me put up the whole work. She is very patient in listening to all the new ideas, pragmatic in giving suggestions, and always doing the reality check.

I wish to thank the individuals of my thesis committee: Dr. Yugyung Lee, Dr. Cory Beard and, Dr. Ye Wan,g for generously offering their time, support, guidance, and goodwill throughout the preparation and review of this document. I would like to thank the University of Missouri-Kansas City for providing me with a platform to plan and execute my research work. It provided me with many opportunities to support myself and world-class facilities to research the machines available from January 2019 until Dec 2020.

Finally, I might want to thank my parents and family members for their support and guidance in every path of my life. They are the foundations of all my academic progress to date. I extend my gratitude and appreciation to all the friends and persons who co-operated with me in this journey.

CHAPTER 1

INTRODUCTION

Sound recognition systems are usually utilized for the assignments of discourse and music signal handling. In the meantime, natural sound recognition and the order have gotten much consideration as of late. There are different applications previously proposed in a wide variety of businesses, including monitoring [1], sound scene recognition for robot navigation [2], acoustic checking of the familiar and artificial environment [3]. In a carefully changed society, soundscape models make an exploration point of view in the intelligent city area. City commotion overseeing fundamentally adds to a sound and safe living condition in the vast urban areas. In movement driven frameworks, city sounds may enter the rising answers for creating and share venture experience. Helping innovations for individuals with handicaps and, specifically, route frameworks for daze or outwardly disabled individuals viably consolidate the urban sound models [4].

Until now, an assortment of sign preparing and AI methods applied to the problems, including matrix factorization, wavelet filterbanks [5], and most as of late profound neural systems. Specifically, profound convolutional neural networks (CNN) [6] are, on a fundamental level, very appropriate to the issue of ecological sound characterization: they are fit for catching vitality tweak designs over time and recurrence when applied to data sources like spectrograms, which is appearing to be a significant attribute for recognizing unique, frequently clamor like, sounds, for example, motors and jackhammers.

Using convolutional filters, the system should classify urban sounds even if the additional background noise is added. Most of the sound recognition systems fail to

recognize, for example, MFCC. However, the application of CNNs has been restricted for a specific part of the sound classification. Deep Neural Networks (DNN), which are very useful for high-dimensional and complex data learning, have recently been investigated for environmental acoustic classification. However, present deep CNNs spend much time in model tuning, which is not helpful for robustness and parameter update for a complex urban acoustic database.

When it comes to the audio features, they include waveform, spectrogram, Mel spectrogram, Log-spectrogram, and MFCC. These features are generated according to the characteristics of the human ear. They have different hearing sensitivities to sound waves of different frequencies. Its primary function is to highlight low frequencies and suppress high frequencies. What's more, the Mel-Spectrogram, Log-Spectrogram, and MFCC representations are quite visible in the images. Therefore, it is widely used in the ESC task. It is undeniable that compared with Mel-Spectrogram, Log-Spectrogram, and MFCC, the original audio features can fully express the audio information while the parts of other audios will cause a specific loss in the audio data. Thus, many teams are attempting to extract sound features from the original audio. In 2016, the Google team tried extracting features from the original audio. The model's input was changed to the time-frequency spectrum of complex signals to retain original audio information. Simultaneously, both the energy spectrum and the phase spectrum of the audio signal were preserved. However, their results showed that compared with Mel, LM, and MFCC, using original audio as input conferred no clear advantages in the performance.

The combination of the above audio functions and classifiers can achieve good recognition results in ESC. However, multiple restrictions still exist. For example, only single audio features are used as the input of a classifier. Single audio sources cannot

capture all the critical information about environmental audio events. In addition to the above restrictions, the accuracy and efficiency of ecological sound recognition are not good enough. There is a significant gap between the model performance and that of humans. The deficiency is evident only when the ESC systems operate in noisy and complex environments.

In our experiments, four datasets are utilized: ESC-10 [7], ESC-50 [7], UrbanSound8k (Us8k) [8], and Emotion dataset. For deep learning tasks, large samples would be required for accuracy and better performance. To avoid over-fitting problems, data augmentation is necessary for these learning tasks. Data augmentation can provide more training data to reduce the possibility of overfitting in training and increase the models' accuracy and performance. In this study, significant data augmentation is presented, which significantly improves the performance of learning.

In this study, three different spectral image spectrogram features are extracted from audio clips. This approach includes various components such as feature extraction from ConvNets, model training, and features aggregation. Spectral features introduced for this thesis are spectrogram (SG), Chromagram and, MFCC. These features are trained with three lighter network CNNs, respectively. Finally, the outputs derived from the Global Average Pooling layer from three CNNs are fused with different aggregation methods. The experimental results indicate that the AudioCNN model outperforms the existing models on the Urbansound8K dataset.

CHAPTER 2

BACKGROUND

This chapter gives all the background information about the various elements and components used in this work. It provides an overview of transfer learning and data augmentation, which will help understand this work better.

2.1. Deep Learning

Deep Learning is about learning multiple representations and abstractions that compute nonlinear functions and understand highly complex data. Each input from previous layers is passed through these activation functions, and transformed data is passed through the next layers. Each layer consists of neurons that are the fundamental basic units of the network and have various connections to the same layer and dependent layers.

Essentially, deep neural networks have been inspired by human brain architecture. These output signals can trigger a few or several neurons to perform a particular section. The functionality is mimicked in building the architecture of the neural network. Neural network algorithms are models that represent the human brain and are good at identifying nonlinear patterns or patterns to be reused. These can be described as graphs in which nodes are neurons, and connections between them have directed edges with specific weights. An example of an artificial neural network is shown in Figure 1.

The ability to learn quickly, be useful, and suitable for various other tasks makes them powerful. So, the information in an artificial neural network flows in two ways. When the model is being trained with information in the dataset that is fed

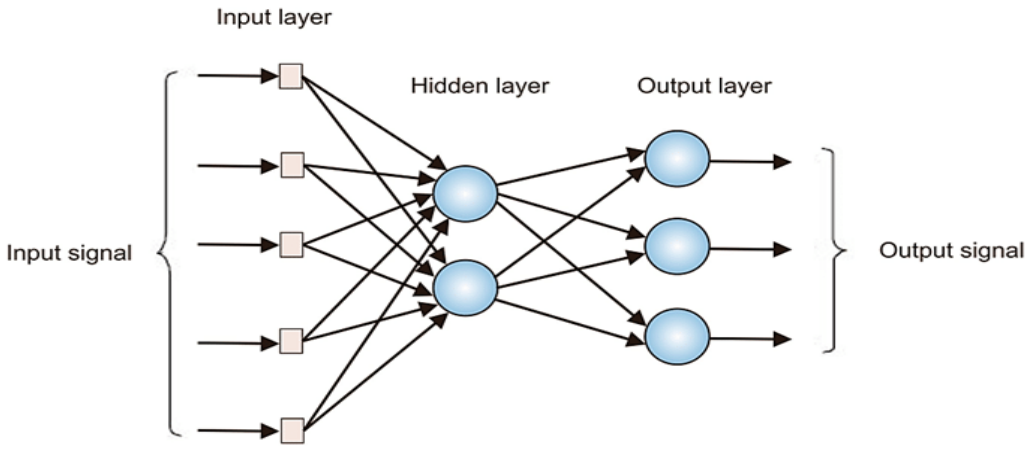


Figure 1: Neural Network Architecture

into the network, or after being trained, the model is tested. The data is fed from the input layer and passes on through subsequent layers, hidden layers, and then final layers. This neural network shown above is popularly called a feed-forward neural network. The neurons receive input from the neurons to the left, then the inputs are multiplied by the weights, and they proceed along. The process by which neural networks learn things is called backpropagation [9]. The actual expected output is differentiated from the network's output, and the weights are adjusted in the neurons accordingly. A deep neural network using backpropagation is responsible for learning and better performance with iterations.

2.2. Convolutional Neural Networks

A conventional neural network comprises a few layers with neurons where every neuron in each layer is associated with all previous and next layers. These fully connected layers form a network. The first and last layers are named "input layer" and "output layer," respectively. The layers in between are known as "hidden layers." The neural network takes in an input vector and transforms it by passing it through the hidden layers. For conventional neural networks, if the size of data increased or hidden layers are added, the network's complexity will result in extensive parameter

calculations with heavy computational use, which results in overfitting problems without any meaningful accuracy loss results.

Convolutional Neural Networks [10], on the other hand, have only limited connections with previous layers of neurons, and these networks have only a few fully connected layers. This encourages a type of local spatial relationship in the data. In other words, a neuron can only "see" a small portion of the layer before it and is unaffected by changes in different regions and automatically forms a hierarchy of features when multiple layers are stacked, which increases abstraction from low-level to high-level. This medium concludes that the input data is gradually learned from the first layer to the last layer and makes decisive conclusions.

2.2.1. Convolutional Layer

The Convolution can be referred to as an operation between two matrices. One matrix is a receptive field with a restricted portion, which is input. The other matrix is filter or kernel, which is a set of learnable parameters. Generally, these kernels are spatially smaller than input size, but it does have the same depth. For images, it accepts a three-dimensional matrix(width, height, color channels) with the more common use of two-dimensional Convolution that results in outputs with a three-dimensional matrix. When a specific feature is detected, the network will learn filters that are activated.

Convolutional Neural Networks have a sparse interaction, which is each filter of a convolutional layer. In CNN's, local regions of an input signal are shared with equal weights. This property is called parameter sharing. Each output feature map describes different detected features, which means changes in the input result in the same outputs. This property comes under equivariance or translation.

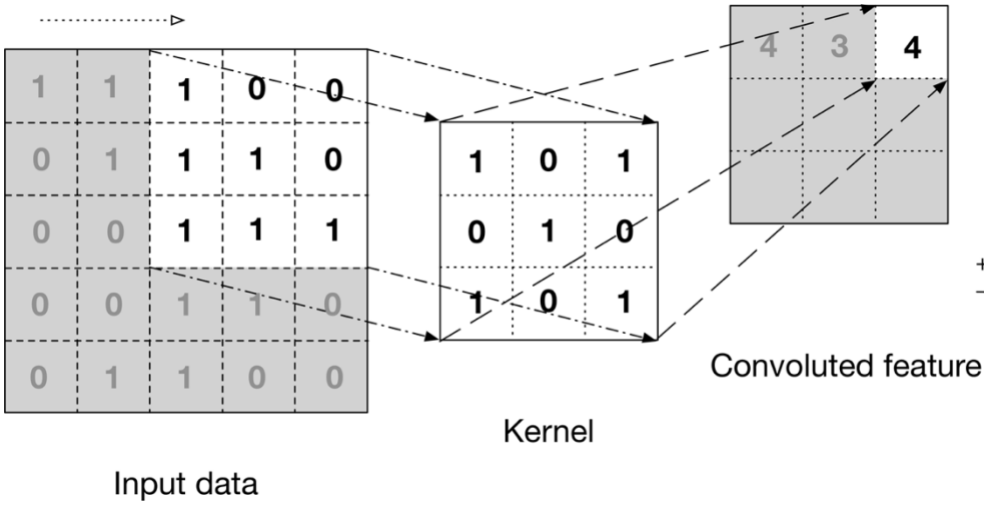


Figure 2: Convoluted Feature Output with Kernel Filter

As shown in Figure 2, the filter size of nxn is sliding across the height and width of an image-producing resultant convoluted feature with a two-dimensional representation of an image known as an activation map.

2.2.2. Pooling Layer

After the convolutional stage, activation functions are used to produce nonlinear representations. Further pooling layers are applied—this pooling implementation helps in better computation and decreased weights and parameters. A typical pooling layer function can use several operations on a rectangular neighborhood, such as maximum, average, L^2 norm. Figure 3 shows a simple max-pooling example.

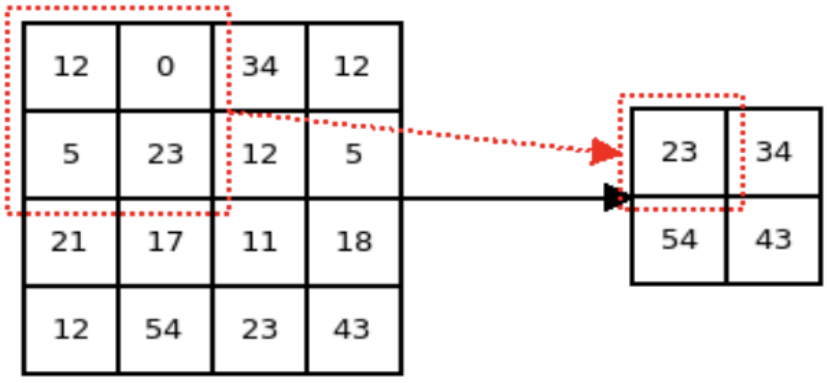


Figure 3: Max Pooling

Out of these operations, the MAX pooling operation is applied in most practical applications, and it returns the maximum values from the rectangular neighborhood. Pooling operations have a computational cost of the network. Nevertheless, pooling remains translation invariance, which means detecting if the feature is present regardless of its appearance.

2.3. Activation Functions

Activation functions [11] illustrates the input-output value relations in a linear and nonlinear way. Each neuron is attached with an activation function, and it should be activated or not based on relevance to model prediction with each neuron's inputs.

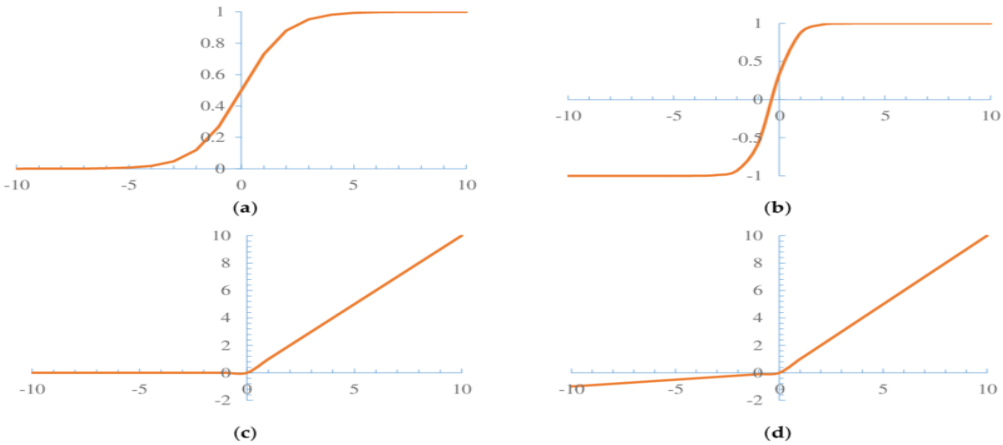


Figure 4: Different Activation Functions

Activation functions normalize each neuron's output between 0 and 1 or between -1 and 1.An additional characteristic with activation function is it must be computationally efficient because it deals with millions of calculations with each sample inputs. Nonlinear activation functions give the power to be more flexible in elaborating arbitrary relations. Here, described most commonly used activation functions. Figure 4 represents the different activation functions.

Sigmoid Function

The sigmoid function takes the form:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

It takes a real number, x, input values give the output ranges between 0 and 1. Therefore, this sigmoid function is mainly used for models to predict the probability as an output. Its shape is shown in Figure 4(a).

Hyperbolic Tangent Activation Function(Tanh)

The tanh function is similar to the sigmoid function. The range of tanh function is from -1 and 1. It takes the form:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

The tanh function's advantage is that output values will be either negative or positive, which means output for the next layers may be the same sign, and it is symmetric around the origin. Its shape is shown in figure 4(b).

Rectified Linear Unit (ReLU)

Rectified Linear Unit takes the form:

$$f(x) = \begin{cases} 0, & for \;\; x < 0 \\ x, & for \;\; x \geq 0 \end{cases}$$

It is the most popular and commonly used activation function in all the convolutional neural networks, and it computes the function f(x)=max(0,x). The ReLU activation function's issue is that all the negative input values are turned immediately into zero, which affects the model's ability to fit or train the data correctly. Its shape is shown in Figure 4(c).

Leaky rectified Linear Unit (ReLU)

Leaky ReLU takes the form:

$$f(x) = \begin{cases} \alpha x, & for \quad x < 0 \\ x, & for \quad x \geq 0 \end{cases}$$

It is an improved version of the ReLU function. The ReLU activation function gives output zero for the negative input values, which deactivates the region's neurons. Leaky ReLU addresses the problem by defining the extremely small linear component α, whose value is 0.01. Its shape is shown in Figure 4(d).

Softmax Function

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}$$

The sigmoid activation function is often referred to as a combination of multiple sigmoids. We know the sigmoid activation function returns the probabilities for a data point belonging to a particular class. In contrast, the softmax function returns the probability for a data point belonging to each class. For multiclass classification problems softmax function is applied.

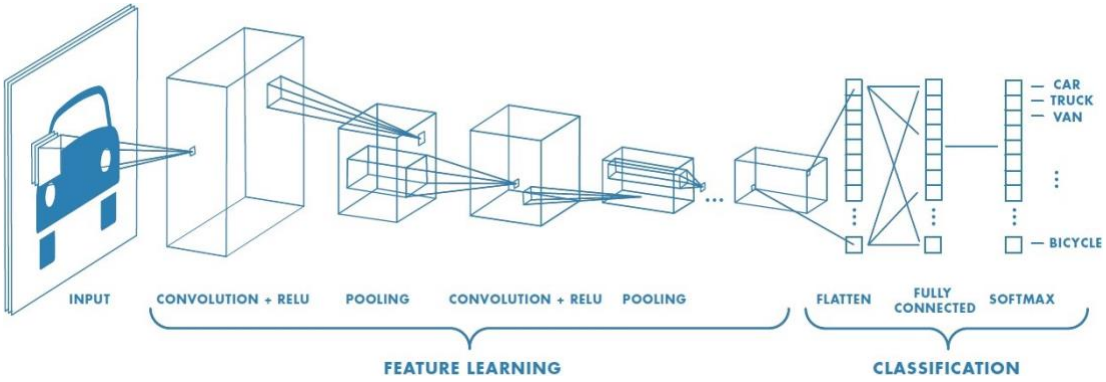


Figure 5: Overview of CNN Architecture with Many Convolutional Layers

Technically, deep learning CNN models are applied to image data. Each input image is passed through convolutional layers, which contain filters (kernel), Pooling layers, Fully connected layers (FC), and softmax layer at the end to get the probabilistic

outputs between 0 and 1. The full overview of CNN architecture is shown in Figure 5, which processes the input image and classifies it based on values.

2.4. Transfer Learning

Transfer learning, used in machine learning where the knowledge of an already trained model is applied to different source or domain task problems. For example, the classifier learned to detect the cars can be used as a pre-trained model to recognize other objects like motorcycles in another model. The general idea is to use a trained model on a large dataset with more classes to a new task with limited data samples and few classes. Transfer learning is applied mostly in computer vision and natural language processing tasks like sentiment analysis, which requires large computational resources. Keras Framework [12] provides several pre-trained models that can be used for transfer learning with ImageNet [13] as source task, feature extraction, prediction, and fine-tuning. Some of the popular pre-trained models are VGG16 [14], ResNet50 [15], Inception-v3 [16], Xception [17], DenseNet [18].

VGG [14] network was released in early 2015 by the Visual Geometry Group at Oxford University, which supersedes the previous network AlexNet, the first convolutional neural network trained on ImageNet [9] dataset by improving its performance and accuracy. AlexNet architecture is built with different filter sizes(11x11,5x5,3x3) consisting of 5 convolutional layers, followed by three fully connected layers. VGG [14] network was constructed with a fixed filter size(3x3) over every layer with a stride value of 1, which detects the small details in the images, leading to an increase in performance accuracy.

VGG [14] takes an input size of 224x224. As input goes deeper into the network, the backpropagation technique is applied to reduce loss and adjust the weights based on misclassified data. When the network is deep with multiple layers

stacked, there is a vanishing gradient problem. The input is passed through various layers with activation functions, gradients of the loss function approach to zero, making the network hard to learn. This saturation problem has led to the implementation of a new network called ResNet [15].

ResNet [15] was first introduced by Microsoft research in the year 2015. This network's core idea is to solve the vanishing gradient problem by implementing a new approach called the "identity shortcut connection," where the outputs of previous layers are skipped and introduces a shortcut that forces them to learn the residual between the input and output instead of directing mapping. ResNet [15] consists of around 11 million parameters, with 18 convolutional layers followed by a fully-connected layer and softmax. ResNet [15] has different versions like ResNet-18, ResNet-50, ResNet-101, and ResNet-152, it denotes the no of layers in the network.

The Google Research team developed Inception-v3 [16]. The core idea is to apply convolutions with different filter sizes on the same input and concatenate them together. Dealing with vanishing gradient problems and better convergence auxiliary classifiers are also attached before the end of the network, which pushes the useful gradients to the lower layers to make them immediately useful. Identifying the small and global features present in the images inception introduces various filter sizes like 1x1,3x3,5x5. Large filter sizes capture global features, and small filter sizes detect other small features in the images.

Xception [17] stands for the Extreme version of inception, introduced by Google. The Xception architecture is a linear stack of depth-wise separable convolutional layers with residual connections. Inception-v3, depth-wise separable Convolution (depth-wise Convolution followed by pointwise Convolution) with a large no of towers is applied. This observation leads the authors to propose a novel

deep convolutional neural network inspired by inception, where the inception modules are replaced with depth-wise separable convolutions. Xception significantly outperforms Inception-v3 on a large image classification dataset consisting of 350 million images and slightly outperforms Inception-v3 on the ImageNet dataset.

DenseNet [18] (Densely Connected Convolutional Networks) is a logical extension of ResNet, where residual learning is learned by merging previous layers with future layers. In DenseNet, each layer connects with every other layer in a feed-forward fashion with concatenating outputs from the earlier layers instead of using the summation. Generally, traditional convolutional networks with L layers have L connections, one between each layer and its subsequent layer. Whereas in DenseNet has L(L+1)/2 connections. For each layer, the feature-maps of all former layers are used as inputs and, its feature-maps are utilized as inputs to all subsequent layers. DenseNet solves the vanishing gradient problem and strengthens feature propagation. This dense connectivity pattern requires no need to relearn redundant feature maps that require fewer parameters than traditional convolutional networks.

2.5 Data Augmentation

However, hyperparameters and model architecture are the best factors for building a useful model. Instinctively, lack of data is one of the common issues in an actual deep learning problem. Data augmentation helps generate synthetic data from the existing data set such that the generalization capability of the model can be improved. The most traditional practice for image data augmentation is to perform image modifications and affine transformations such as Zoom range, Width shift, Fill mode, Brightness range, Rotation angle, Height shift, Shear range, Reflection, and Horizontal flip. These transformations are adopted until the desired no of training data is generated. The flow of the traditional augmentation approach is shown in Figure 6.

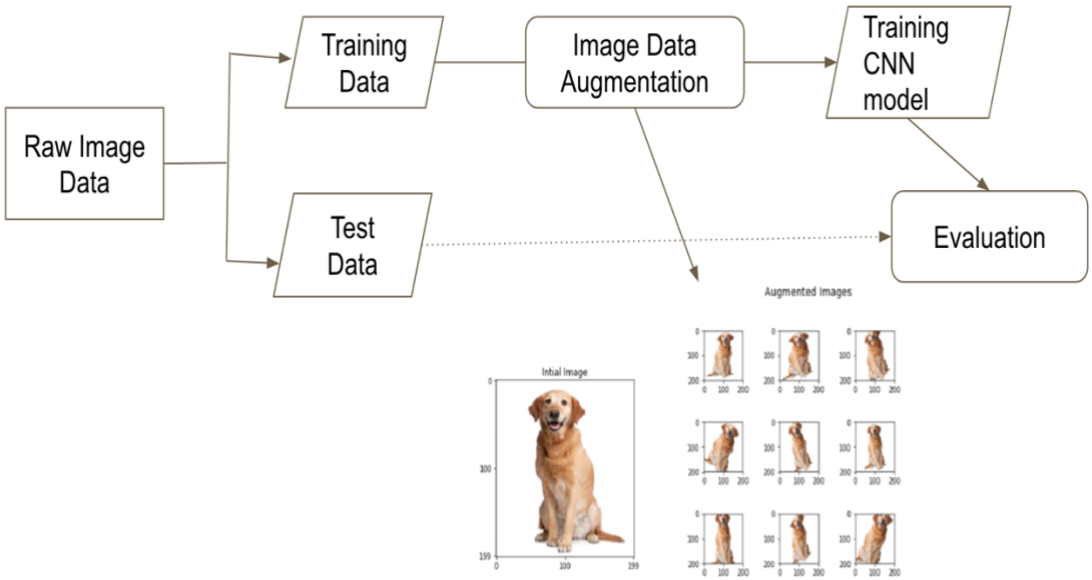


Figure 6: Traditional Image Data Augmentation Approach

This approach aims to improve the data that the system can learn different input patterns while training. Thus, we proposed to consider variations of data that relate to spectrogram-based augmentations. This approach is shown in Figure 7.

For audio data augmentation, several techniques can be adopted, such as noise injection, shifting time, changing pitch, and speed.

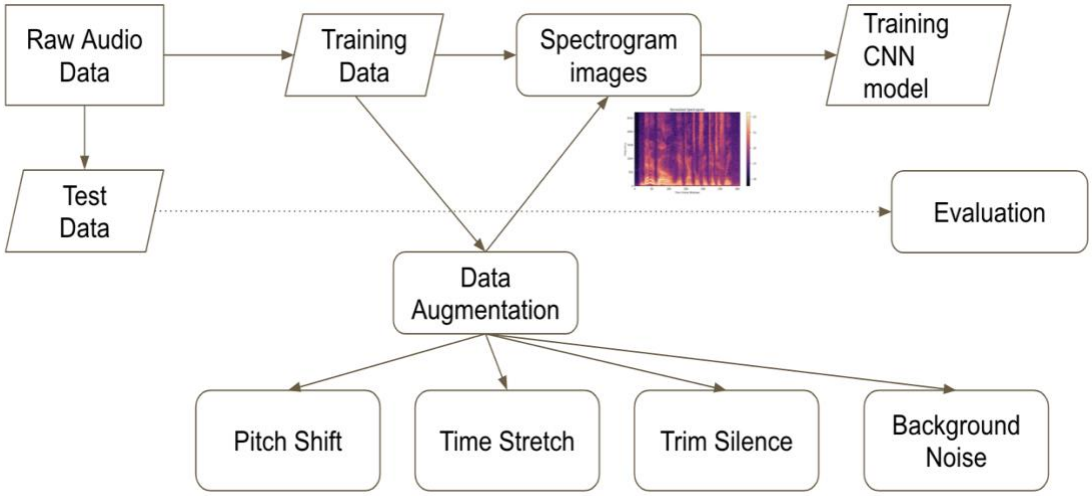


Figure 7: Spectrogram-Based Image Data Augmentation Approach.

In this section, various data augmentation methods are adopted. These variations were considered after observing the baseline model's first results and the

results after evaluating noise and amplitude corrupted inputs. The model was then trained on these variations to study different behaviors. The various audio data augmentation approaches are described below.

Noise Injection

This technique adds background noise to the original audio by adding a random value to data using NumPy. Figure 8 exhibits a simple noise injection example.

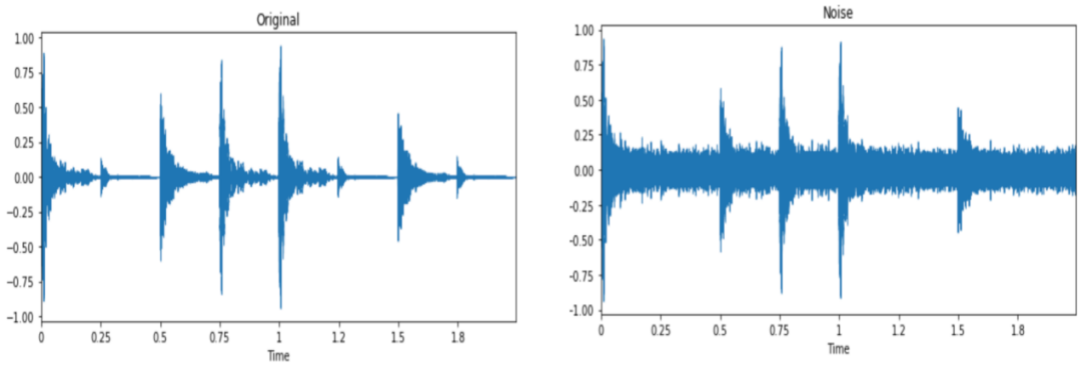


Figure 8: Noise Injection Example

Time Shifting

The idea is to shift the starting point of the audio slightly, then pad it to the original length. If shifting happens to fast forward, then the first part will become silence. If activity occurs back ahead, then the last audio element will become silent. Figure 9 shows a simple change in time shift example.

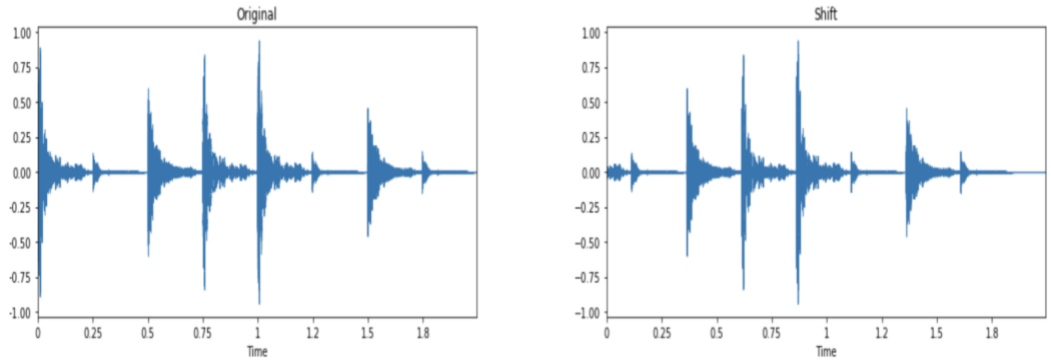


Figure 9: Change In Time Shift Example

Changing Pitch & Speed

The technique is to change pitch randomly and stretches time series by a fixed rate. The Time_Stretch function will take inputs as wave samples and factor by which to stretch. The Pitch_Shift process will take inputs as wave sample, sample rate, and several steps to shift pitch. Figure 10 shows a simple example of the change in pitch and speed example.

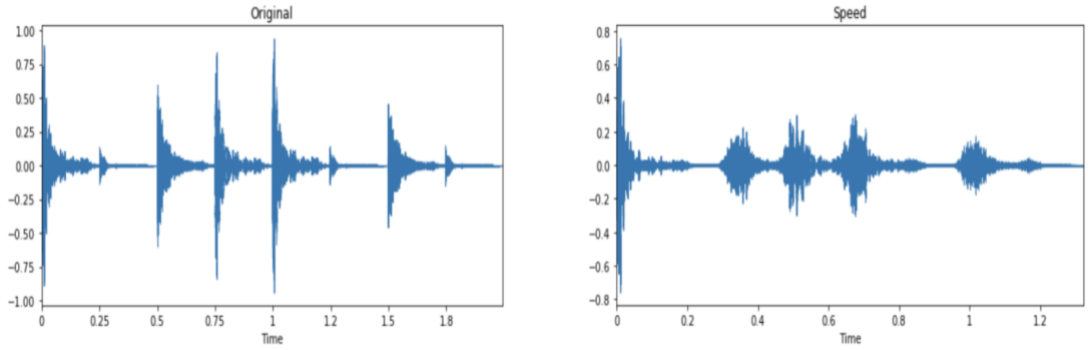


Figure 10: Change In Pitch & Speed Example

In this approach, augmentation is employed with each variation implemented on the whole audio dataset individually. For ESC-50, there are 2000 original audio clips, 2000 generated from background noise, 2000 generated from time shift, 2000 caused from the change in pitch, 2000 generated from the difference in speed & dynamic range. Thus, there is a total of 10,000 audio files. For ESC-10, there are 400 original audio clips, 400 generated from background noise, 400 generated from time shift, 400 caused by the change in pitch, 400 induced from the difference in speed & dynamic range. Thus, there is a total of 2000 audio clips. For the Us8k dataset, there are ten folds, with each fold around 1000 audio clips. For this study, only three folds are considered, which gives us around 3000 original audio clips. Thus, there are a total of 23,732 audio clips.

CHAPTER 3

RELATED WORK

This study aims to research the performance of deep convolutional networks designed for image classification in classifying environmental sounds.

3.1. Overview

To establish the current standards and have a baseline for comparison, a literature study was conducted. Wang et al. [19] discuss the efficiency of a Gabor-based non-uniform scale frequency map that combines Principle Component Analysis and Linear Discriminate Analysis to extract features from the sound samples followed by classification using Support Vector Machines (SVMs). A high accuracy rate is reported.

Deep learning methods have been implemented and tested in relatively few cases, with mobile platforms being almost non-existent use-cases. Mostafa et al. [20] perform music sample classification using Probabilistic Neural Network with satisfactory results. Zhang et al. [21] a voice problem that man labeling datasets are very costly and recommend semi-supervised learning to be a better solution. McLoughlin et al. propose a Deep Neural Network as a viable solution. Piczak et al. [23] and McLoughlin et al. both propose similar methods that spectrum analysis preferably giving the best accuracies with Convolutional Neural Networks and is best for the case where the training data is limited. Piczak et al. [23] applied the augmentation technique to increase data sizes by adding random delays and class-dependent time stretching to the original recordings. Based on Dataset sizes ESC-50, ESC-10, different variations are applied. Discarding the silence and extracting spectrogram from an audio sample with a 50% overlap.

Along with spectrograms, the deltas were computed and fed into the network in two channels. Sharma *et al*. [24]. Proposed a model with Augmentation with DCNN to get an accuracy of 98.60% for Us8k, 97.25% for ESC-10, 95.50% for ESC-50. Aytar *et al*. [25] proposed using unlabeled video for transferring the specific visual information from well-trained image recognition to audio and called this approach as SoundNet. It achieved an accuracy of 74.2% for ESC-50 and 92.2% for ESC-10.

3.2 Related Approaches

Historically, many research works on Environmental Sound Classification (ESC) have relied on spectrogram analysis. This master thesis focuses on Deep Convolutional Neural Networks applied with spectrogram approaches. Boddapati et al. [26] proposed a model in which audio represented in the form of visual images by converting into a spectrogram (representation of the energy in the spectrum of frequencies varies with time), Mel-Frequency Cepstral Coefficients (MFCC) is a nonlinear representation of the power spectrum of a sound adjusted to log scale. Cross recurrence Plot (CRP) is a matrix visualization of each element representing a time series's phase trajectories. In this approach, three extracted images combined into a single image, and then for evaluation, ESC-50 and ESC-10 are considered and applied two pre-trained models AlexNet and GoogLeNet.

The best accuracy achieved 93% on the UrbanSound8k dataset with pre-trained model GoogLeNet using Spectrogram feature, and the best accuracy is 91% for ESC-10 and 73% for ESC-50.

Zhang et al. [27] proposed two combined features to give a more extensive representation of environmental sounds, and then a four-layered convolutional neural network is presented. Finally, the CNN features are fused with Dempster-Shafer evidence theory to compose the TSCNN-DS model. As log-spectrogram and MFCC

features extracted at first. Then, chroma, spectral contrast, and Tonnetz are aggregated with log-spectrogram and MFCC to form LMC and MC feature sets. The extracted feature sets passed through two same convolutional neural networks of the same dimensions, and then evidence theory is applied at the softmax layer of two models to get meaningful combinations from the two modes. The best accuracy achieved on the UrbanSound8k dataset with two feature sets, LMCNet and MCNet, is 95.2% and 95.3%. The proposed TSCNN-DS model achieves 97.2% on the UrbanSound8k dataset.

Mushtaq et al. [28] focus on the task of environmental sound classification with three audio feature extraction techniques like Mel-Spectrogram (Mel), Mel-Frequency Cepstral Coefficients (MFCC), and Log-Mel by using Deep Convolutional Neural Networks (DCNN) and also applied significant data augmentation. In this study, the author proposes two convolutional neural networks, one with max-pooling function represented as Model-1 and the second without max-pooling function represented as Model-2. The audio extracted features are combined into a single image and passed through both Convolutional networks and introduced some offline data augmentation techniques to enhance the datasets using L2 regularization. Besides, various well-known deep networks with a transfer learning approach and pre-trained with freezing the initial layers and unfreezing the layers with optimal learning and discriminative learning proposed in the following paper. This study exhibits the state of art-results with a combination of transfer learning and discriminative learning and the highest accuracy achieved for ESC-10 (99.04%), ESC-50 (97.57%), and Us8k (99.49%).

CHAPTER 4

METHODOLOGY

4.1 Overview

This chapter gives more detail regarding the architecture and the components used in identifying the individual environmental sounds. As discussed in chapter 1, the proposed solution consists of several components. Figure 11 shows the overall architecture of the proposed solution with few steps. In the following sessions, each component illustrated in detail regarding the architecture and outline implementation.

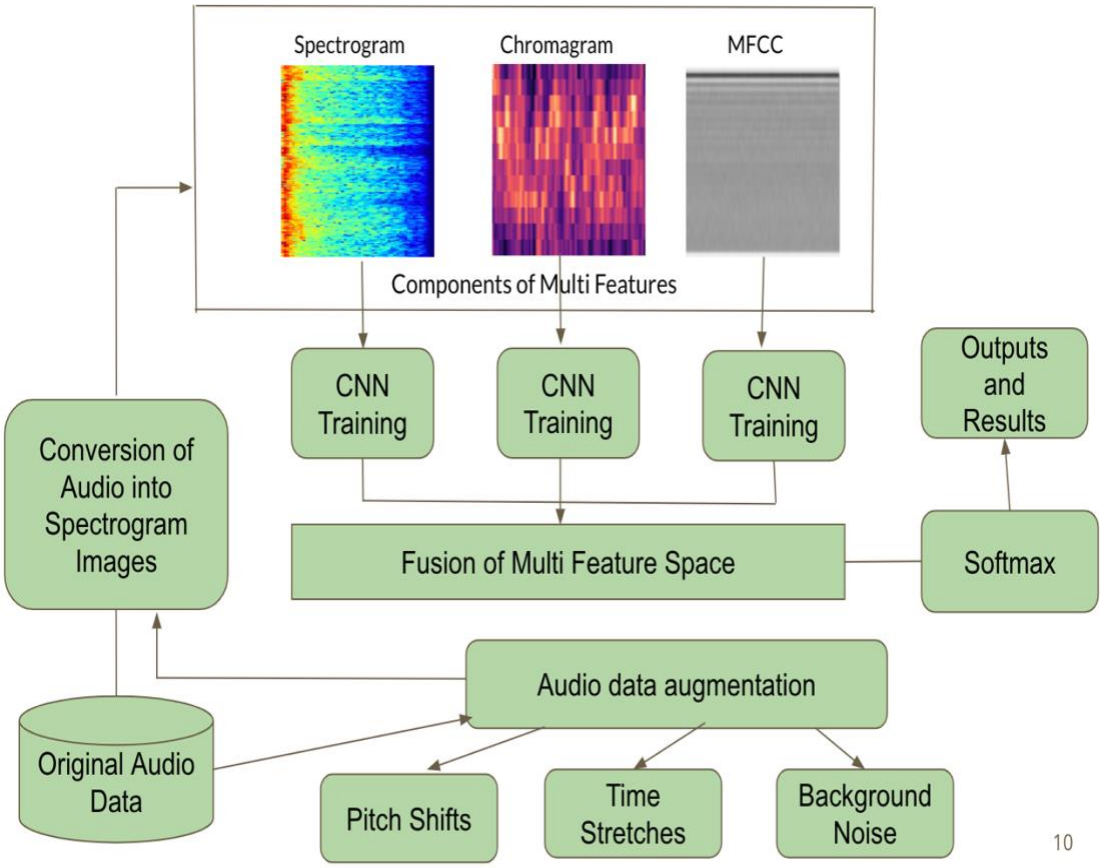


Figure 11: Overall Proposed Framework

4.2 Data Collection and pre-processing

One of the primary issues with training deep convolutional neural architectures requires large amounts of computational effort and labeled data required for efficient learning. While hardware advances and general-purpose GPU computing address the former, the latter is very domain-dependent. Three publicly available datasets were selected to evaluate the models: ESC-50, ESC-10, and UrbanSound8K.

The ESC-50 dataset consists of 2000 short (5 seconds) environmental recordings with equally balanced among 50 classes of sound events in 5 major groups (animals, natural soundscapes and water sounds, human non-speech sounds, interior/domestic sounds, and exterior/urban noises) defined ten classes per category. The field recordings are far from clean and noiseless. Thus there exhibits some overlaps in the background. The extraction process's primary goal is to keep little background when possible and sound events exposed in the foreground. The dataset provides various sound events, some widespread (laughter, dog barking, cat meowing) and some quite distinct(glass breaking, helicopter, and airplane noise).

ESC-10 is a subset of 10 classes (400 recordings) selected from the ESC-50 dataset, represents three general groups of sounds as follows:

- Transient with meaningful temporal patterns(dog bark, clock tick, person sneeze),
- Strong Harmonic content(baby crying, rowing rooster),
- More/less structured noise(rain, sea waves, fire crackling, helicopter, chainsaw).

This subset presents a slightly different problem than the original ESC-50 dataset, where the different classes are much easily distinguished and pronounced with little ambiguity.

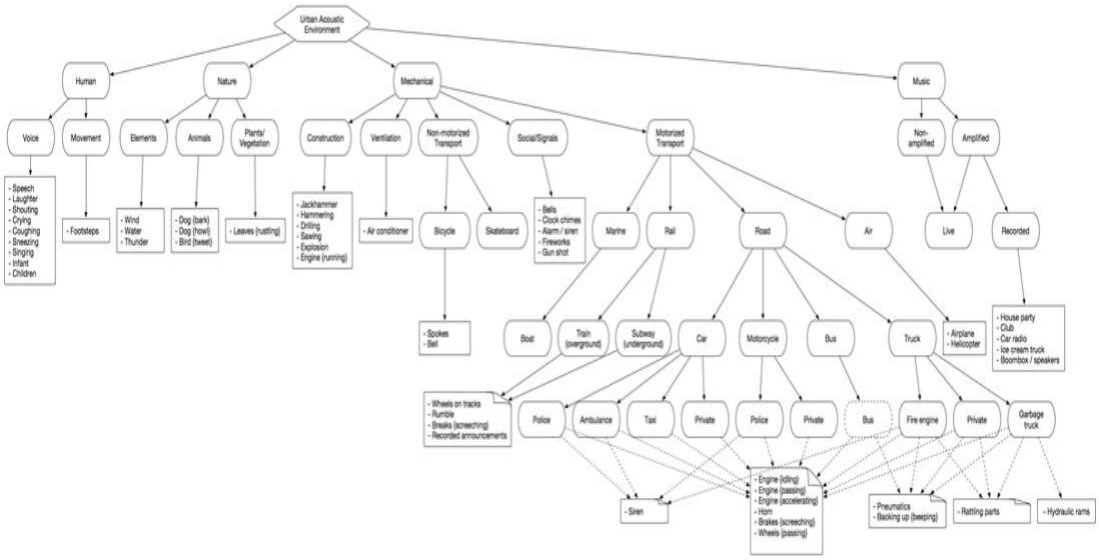


Figure 12: Urban Sound Taxonomy of UrbanSound8k Dataset

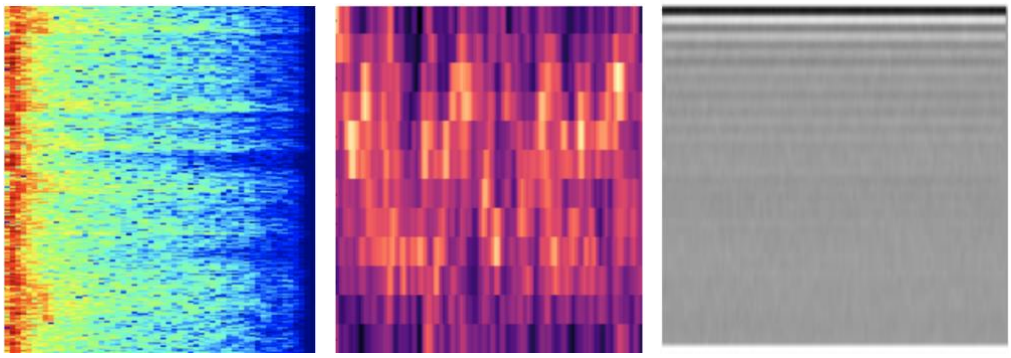
UrbanSound8K is a collection of 8732 short (less than 4 seconds) excerpts of various urban sound sources. Salamon et al. [8] describe a taxonomy of environmental sounds, and the dataset they release includes ten sub-classes from that taxonomy, which are :(air conditioner, car horn, playing children, dog bark, drilling, engine idling, gunshot, jackhammer, siren, street music) prearranged into ten folds for cross-validation. There are up to 1000 examples per class, and each of them has a maximum duration of 4 seconds. Figure 12 shows the sound classes in the UrbanSound8K dataset from urban sound taxonomy. Table 1 shows all the information of datasets.

Table 1. Information of Datasets.

Datasets	Classes	No of samples	Duration
UrbanSound8K	10	8732	9.7 hours
ESC-50	50	2000	2.8 hours
ESC-10	10	400	33 min

4.3 Feature Selection

When it comes to audio features, Spectrogram (SG), Chromagram (CG), and Mel-Frequency Cepstral Coefficients (MFCC) are generated according to the characteristics of the human ear and have different hearing sensitivities to sound waves of different frequencies. They follow a nonlinear method to process the original audio signal, highlighting low frequencies and suppressing high frequencies. The audio features SG, CG, MFCC representations are quite visible in the image representations. Therefore, it is widely used in Environmental sound classification (ESC) tasks. Figure 13 shows the visual representations of different features.



(a) (b) (c)

Figure 13: The Three Types of Extracted Features (a) Spectrogram, (b) Chromagram and (c) MFCC

The spectrogram is a spectrum of frequencies of a signal as it varies with time. The overall extraction of spectrogram features described as follows:

Signal:

An audio signal comprises diverse single-frequency sound waves that travel together as a disturbance or pressure change in the medium. When sound is is recorded, we only capture the resultant amplitudes of those multiple waves is a digital representation of

an audio signal known as a waveform. Figure 14 shows an overall illustration of a signal with overlapping windows.

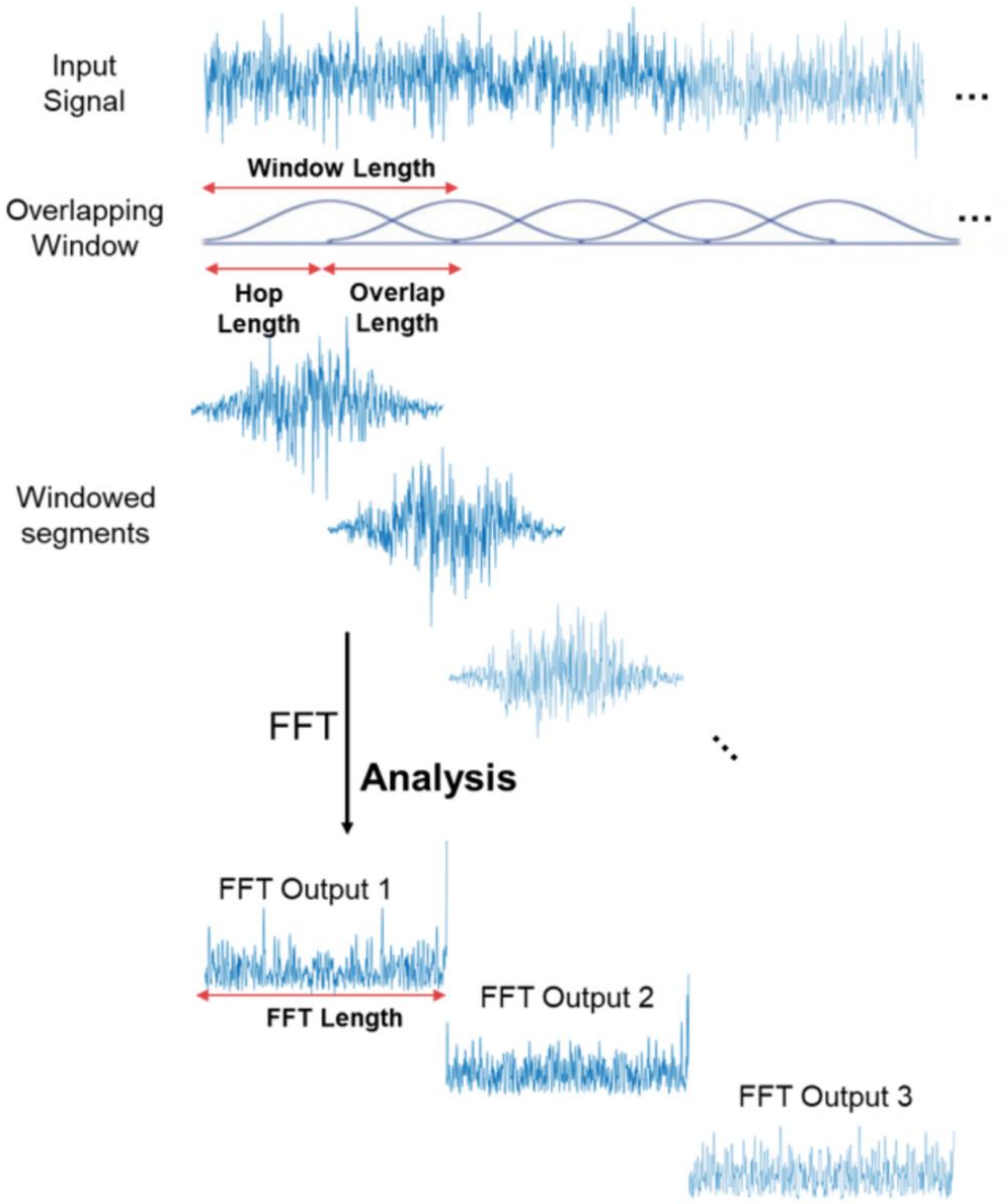


Figure 14: An Illustration of a Signal, after Multiplying with Overlapped Windows.

Fourier Transform:

A Fourier Transform is a mathematical concept that can breakdown a signal into its constituent frequencies, where an audio signal comprised of single-frequency sound waves can convert the time domain into the frequency domain. The result is known as

a spectrum. An algorithm **Fast Fourier Transform (FFT)** used to compute Fourier transforms in signal processing. This single-frequency content varies over time, such as the case with most audio signals like music and speech. These signals are known as non-periodic signals. The idea is to break the spectrum of signals into smaller windows and compute the FFT for each frame, which will get frequencies of each window and window number will represent the time, and it is called a **Short-Time Fourier Transform (STFT)**. Figure 14 illustrates a random signal after multiplying with the overlapping windows and applying FFT on the multiplied windows.

Humans can only perceive a minute and full range of frequencies and amplitudes, where the overlapped windows are transformed both y-axes(frequency) into log-scale and color dimension(amplitude) into decibels, as the log scale of amplitudes to form a **Spectrogram**. Finally, mapped y-axis (frequency) onto the logarithmic-scale to form **Log-Spectrogram**. Log-scaled spectrogram features extracted from all recordings with sample rate 44100 Hz with overlap calculated by multiplying step size and sample rate and segments calculated by multiplying window size and sample rate using Librosa implementation.

Chromagram:

The Chroma feature is a descriptor representing a musical audio signal's tonal content in a consolidated form. Consequently, chroma features as an essential prerequisite for high-level Features for classification. Human perception of pitch is periodic because two pitches differ by an octave if they are similar in color. Separation of the pitch into two components: tone height (octave number) and chroma. Twelve traditional pitch classes on an equal-tempered scale. Figure 15 shows a twelve-dimensional feature vector is formed by adding up all pitches belonging to the same classes.

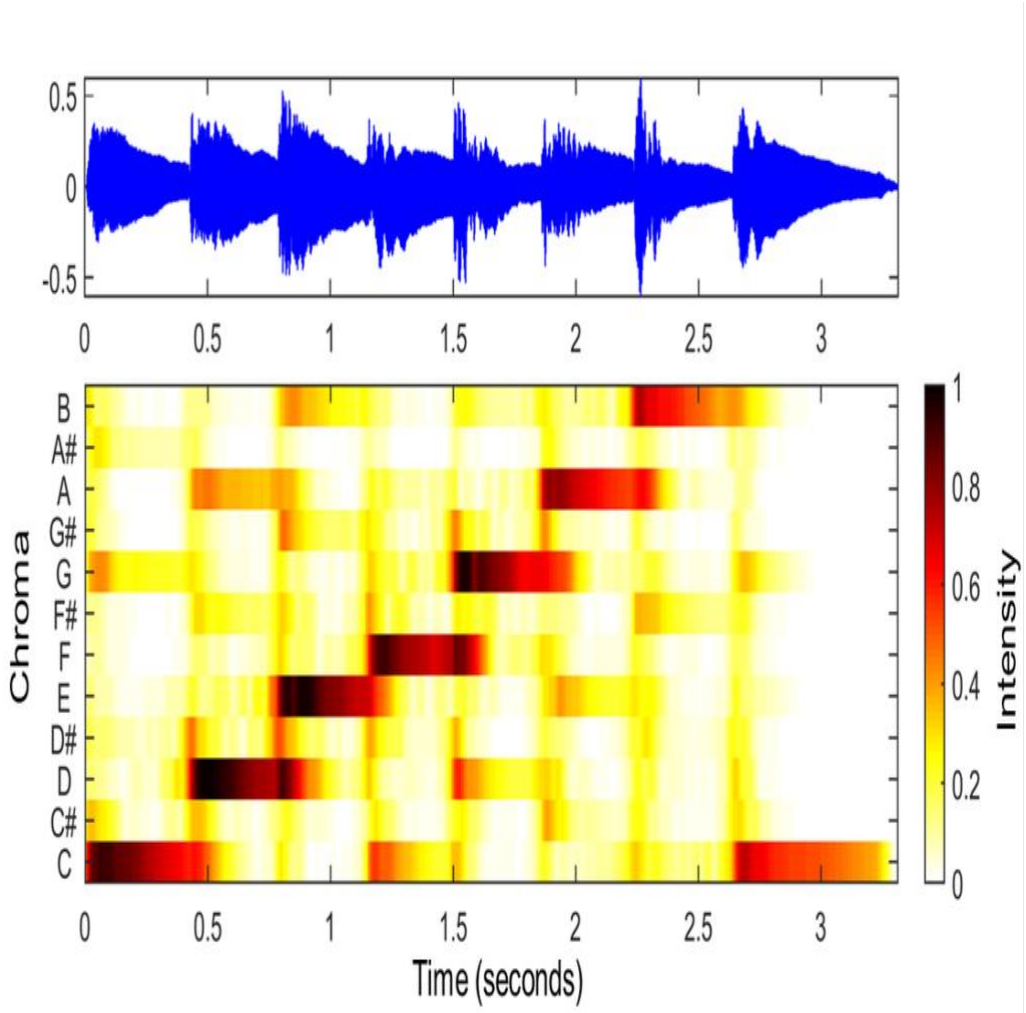


Figure 15: A 12-Dimensional Feature Vector Representation of An Audio Signal.

Mel-frequency Cepstral Coefficients(MFCC):

.Mel Frequency Cepstrum (MFC) represents a linear cosine transform of a short-term log power spectrum of a speech signal on a nonlinear Mel scale of frequency. Mel-frequency cepstral coefficients (MFCCs) are together make up an MFC. The formula as follows:

$$x(f,\tau) = \int_{-\infty}^{+\infty} w(t-\tau)y(t)e^{-j2\pi f\tau}dt$$

y(t) is the time domain signal. x(f, τ) is the frequency domain signal. w(t- τ) is the hamming window. Zero is padded to maximum length in all the datasets. $e^{-j2\pi f\tau}$

represents a Fourier change factor. $x(f, \tau)$ is passed through Mel scale filter banks, Mel spectrogram x(f_{mel}, τ) procured.

The formula of its filter f_{mel} is as follows:

$$f_{\text{mel}} = 2595\log_{10}(1+f/700)$$

f_{mel} is the calculated Mel scale frequency; f is the standard frequency. The number of filter banks to 40. $f_{\text{mel}} \in [0,40)$. The magnitude value converted to a logarithmic formula is as follows:

$$S' = 10\ log_{10}(\frac{S}{ref})$$

After the above pre-processing, discrete cosine transform (DCT) is performed on Log Mel spectrogram to obtain MFCC features. The formula of DCT transformation is as follows.

$$C_t(n)\sum_{m=0}^{N-1} S_t(m)\cos\ (\frac{\pi n(m-0.5)}{m})$$

In the Equation, n is the number of MFCCs; Ct(n) is the n-th MFCC coefficient of the t-th frame; St(m) is the logarithmic power spectrum of the audio signal; M is the number of triangular filters.

4.4 Model Architectures

This study proposed a novel approach to classifying environmental sounds by extracting three visual representations from audio clips. Multi-channel inputs with Convolutional Neural Networks models obtained from scratch and transfer learning models are employed, followed by aggregation of features. A single hidden layer is adopted to convert high level aggregated features into the low-level feature vector. Softmax layer to predict the individual sounds.

As mentioned earlier, training a CNN requires many decisions with architecture(Input size, layer size, filter dimensions, spatial pooling) and hyperparameters(learning rate, batch size, dropout probability, amount of regularization, and different activations applied). This selection process and time required for training a complete model evaluated in detail through the following architecture.

Convolutional Neural Network Architecture:

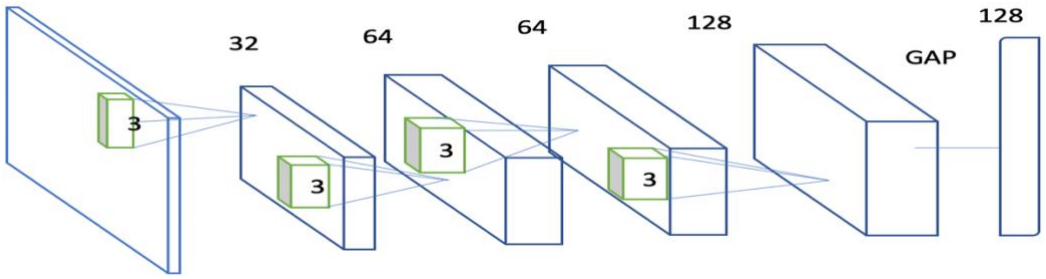


Figure 16: The Architecture of the Proposed CNN Model.

The framework of the proposed CNN model is shown in Figure 16. The model architecture is as follows:

- L1: The first layer uses 32 kernels with a 3x3 receptive field. The Leaky Rectified Linear Unit (ReLU) as the activation function. It is followed by a 2*2 stride step max-pooling function. The dropout rate of 0.25 is also used.
- L2: The second layer uses 64 kernels with a 3x3 receptive field. The Leaky Rectified Linear Unit (ReLU) is used as the activation function. The padding involved in the layer is "same." A 2*2 max-pooling function follows it. The dropout rate of 0.25 is also used.
- L3: The third layer consists of the same 64 kernels with a 3x3 receptive field. The Leaky Rectified Linear Unit (ReLU) is used as the activation function. The padding involved in the layer is "same". It is followed by a 2*2 stride step max-pooling function. The dropout rate of 0.25 is also used.

- L4: The fourth layer consists of 128 kernels with a 3x3 receptive field. The Leaky Rectified Linear Unit (ReLU) is used as the activation function. It is followed by a 2*2 stride step max-pooling function. The dropout rate of 0.25 is also used; the Global Max-Pooling function is utilized.

At the training stage, we use dropout probability to avoid overfitting in every layer. The batch size is set to 64, 128 for each dataset for experimentation. The training lasts for 150 epochs, the Adam optimizer and categorical cross-entropy are used as a loss function.

4.5 Feature Fusion

Generally, for image classification and recognition tasks, single modality features do not capture higher-order representations. Thus it is a viable solution to fuse multiple ConvNets, which can discriminate the classes of different features from multiple inputs. A function f^* : Y^1, Y^2, Y^3->Y^* fuses three different input feature maps $Y^1 \in R^{(1xD1)}$, $Y^2 \in R^{(1xD2)}$, $Y^3 \in R^{(1xD3)}$ and generates a generalized feature map $Y^* \rightarrow R^{(1xD^*)}$. Here, Where D1, D2, D3 are no of feature maps from different ConvNets.

Sum Fusion:

$Y^{sum} = f^{sum}(Y1, Y2, Y3)$, computes the sum of three different feature maps.

$$Y^{sum} = \sum_{i=1}^{3} Y_i$$

This fusion's output dimension is a straightforward stacking of three feature maps, but the output will give competitive results.

The outputs of ConvNets are fused with the concatenation method, then the first dense layer that consists of 500 hidden units with Leaky Rectified Linear Unit (ReLU) as an activation function. The dropout rate of 0.5 is also used to avoid overfitting problems.

Finally, the last layer is the second layer and also known as an output layer, which consists of a total number of classes in the dataset. The activation function used in this layer is softmax to get the target class probabilities. Figure 17 shows an overall framework with a Multi-Channel fusion network.

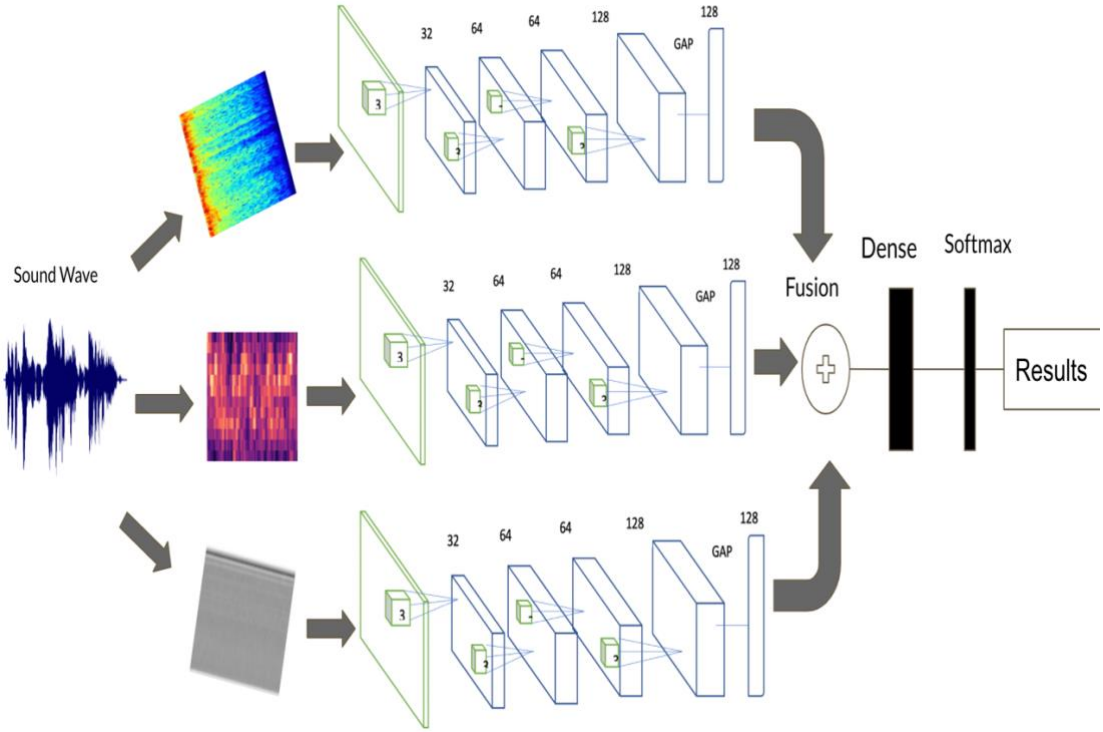


Figure 17: The overall framework of the Multi-Channel Fusion Network.

As shown in Figure 17, we can see that the audio waveform converted into Spectrogram, Chromagram, and MFCC features. Then, input images passed through CNN of the same architecture, and the fusion method is applied. Finally, the softmax is adopted to predict the sound classification results.

CHAPTER 5
RESULTS AND EVALUATIONS

5.1 Introduction

This chapter describes what metrics we have used to evaluate our model predictions and the implementation where the multi-channel input characteristics we compare them and present the results with them.

5.2 Evaluation Setup

All the programming is performed in Python. Besides, all the general libraries used for data processing and analysis in Python, such as Numpy or Matplotlib, four specific libraries were used in this project: The library SoundFile was used to read and write audio files. The audio analysis library Librosa [29] was used to resampling the audio files and generating the Mel-frequency spectrograms fed as training data to the network. Google's library Tensorflow [30] was used for the neural network programming part, version 2.0.0. The library Scikit-learn was used for calculating the confusion matrixes shown in the below sections.

This experiment used Google Colaboratory notebook, which provides all resources to execute and train the model with datasets. A runtime with GPU is Tesla K80 having 2496 CUDA cores with 26GB GDDR5 VRAM and CPU core hyperthreaded Xeon Processors @2.3Ghz two cores and two threads. The hard drive of the runtime machine is 1 TB HDD.

5.4 Evaluation Metrics

The evaluation metrics used to evaluate our models are accuracy, precision, recall, f1 score, and confusion matrix. Accuracy is defined as the correctly classified images' ratio to the total number of images present in the test dataset. Precision is a number of correctly predicted positive images to the classifier's number of positive

results. The recall is defined as the ratio of the number of correctly predicted images to the total number of relevant samples. The confusion matrix shows the evaluated dataset's overall performance, which illustrates how well a model can classify the samples correctly or falsely as either positive or negative, as shown in Figure 18.

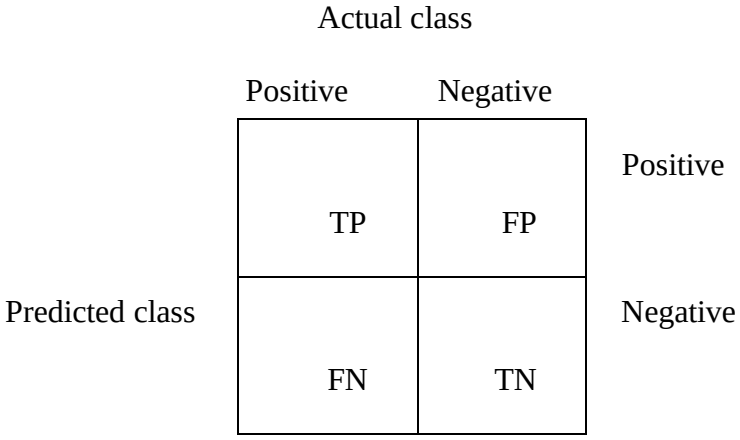


Figure 18: Confusion Matrix

Accuracy = TP+TN/ TP+TN+FP+FN ; Precision = TP/TP+FP Recall = TP/TP+FN

Recall = TP/TP+FN ; F1 Score = (2*precision*recall) / (precision + recall)

5.5 Results and Discussions

This section describes the evaluation performance of the methods and techniques discussed above on environmental sound classification with different inputs and CNN models with data augmentation. In this experimentation, both CNN models trained from scratch and pre-trained models use transfer learning techniques.

Results of using proposed CNN architecture

This section shows the performance evaluation of the proposed CNN architecture for augmented data. Figure 18 shows the training and validation accuracy for both original and augmented datasets, as it can be seen that the proposed CNN architecture with the augmentation approach gave better performance results. Accuracy on the ESC-10 dataset is 87.5% and for the augmented ESC-10 dataset is 99.9%. The ESC-50 dataset is 73.30%, and the augmented ESC-50 dataset is 96.89%.

In the last dataset, UrbanSound8k, the accuracy is 95.5% and for the augmented Us8k is 97.46%. Figure 19 shows the results of the proposed CNN model.

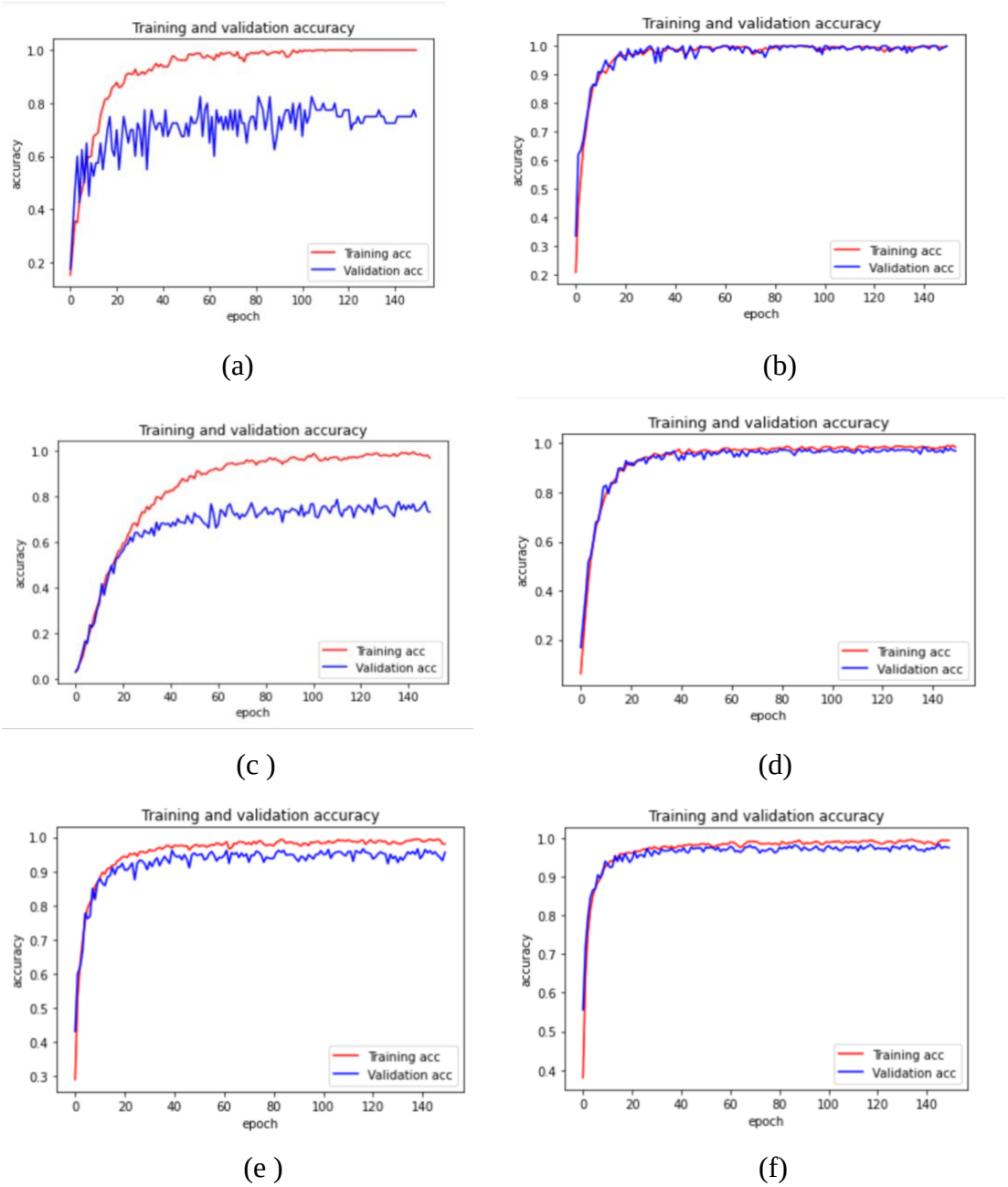


Figure 19: Training Accuracy Vs. Validation Accuracy of proposed CNN model Architecture for Both Original and Augmented Datasets. (a), (b) Related to ESC-10, (c), (d) Related to ESC-50, (e), (f) Associated with Us8k Dataset. The right part of Figure (b), (d), (f) Shows the Results of Augmented Datasets.

Table 2 Comparison of Different Input Feature Combinations on the Us8k Dataset.

Model(Input Feature)	Val Accuracy(%)	Test Accuracy(%)
Spectrogram(SG)	92.39	91.81
Chromagram(CG)	85.33	86.47
MFCC	96.87	97.68
SG + CG	95.41	95.58
CG + MFCC	96.92	96.71
SG + MFCC	97.14	97.52
SG + CG + MFCC	**98.27**	**97.79**

Table 2 shows the experimentation results with a comparison of different input feature combinations on the Us8k dataset. Different input feature combinations are implemented in the proposed CNN model. The input dimension for spectrogram(SG) and Chromagram(CG) is 128x128x3 representation. The 40x174 dimensional MFCC feature has been adopted for this implementation.

Comparing feature combinations, MFCC's feature inputs provide more performance improvement to the proposed CNN model. Input feature combination spectrogram(SG), Chromagram(CG) and, MFCC achieved higher validation and test accuracy among all feature combinations 98.27% and 97.79%, which outperforms the existing state-of-art approaches on the UrbanSound8k dataset. In earlier baseline models, it is described how three pairs of classes are most confused: air conditioner with idling engines, jackhammers with drills and, children playing with street music. As the confusion matrix shown in Figure 20 shows, most of the confused pairs are classified correctly with the proposed Multi-Channel Fusion network.

	precision	recall	f1-score	support
air_conditioner	0.98	1.00	0.99	197
car_horn	0.99	0.98	0.98	100
children_playing	0.94	0.98	0.96	203
dog_bark	0.97	0.97	0.97	202
drilling	0.97	0.99	0.98	203
engine_idling	0.99	0.98	0.98	202
gun_shot	0.97	0.99	0.98	89
jackhammer	0.99	0.99	0.99	251
siren	0.99	1.00	0.99	210
street_music	0.99	0.91	0.95	199
micro avg	0.98	0.98	0.98	1856
macro avg	0.98	0.98	0.98	1856
weighted avg	0.98	0.98	0.98	1856
samples avg	0.98	0.98	0.98	1856

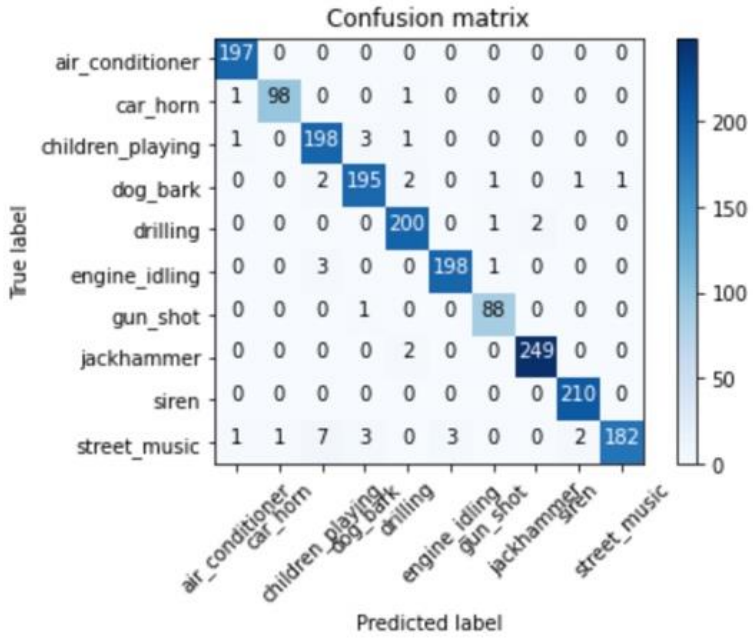


Figure 20: Confusion Matrix Evaluated on the UrbanSound8K Dataset.

Table 3 shows the experimentation results with a comparison of different input feature combinations on the ESC-50 dataset. Input feature combination spectrogram(SG), Chromagram(CG) and, MFCC achieved higher validation and test accuracy among all feature combinations 97.57% and 97.89%, which outperforms the existing state-of-art approaches on the ESC-10 dataset

Table 3 Comparison of Different Input Feature Combinations on the ESC-50 Dataset.

Model(Input Feature)	Val Accuracy(%)	Test Accuracy(%)
Spectrogram(SG)	93.19	93.30
Chromagram(CG)	86.79	85.79
MFCC	97.69	97.79
SG + CG	93.90	93.50
CG + MFCC	97.50	97.60
SG + MFCC	97.56	97.79
SG + CG + MFCC	**97.57**	**97.89**

Similarly, In Table 4, experimentation results are shown to compare different input feature combinations on the ESC-10 dataset. Input feature combination spectrogram (SG), Chromagram (CG) and, MFCC achieved higher validation and test accuracy among all feature combinations 100.00% and 98.00%, which outperforms the existing state-of-art approaches on the ESC-10 dataset

Table 4 Comparison of Different Input Feature Combinations on ESC-10 Dataset.

Model(Input Feature)	Val Accuracy(%)	Test Accuracy(%)
Spectrogram (SG)	97.50	97.50
Chromagram (CG)	94.99	94.49
MFCC	98.50	99.50
SG + CG	98.00	98.00
CG + MFCC	98.50	99.50
SG + MFCC	99.00	99.50
SG + CG + MFCC	**100.00**	**98.00**

Emotion Dataset:

In this study, the emotion dataset under the speech recognition task is also considered an evaluation dataset for our proposed Multi-Channel Fusion network model. This dataset [30] consists of around 1400 samples labeled as eight classes 0=neutral, 1=calm, 2=happy, 3=sad, 4=angry, 5=fearful, 6=disgust, 7=surprised. The proposed augmentation approach is performed on this dataset resulting in around 12,000 audio samples. Figure 21 shows the emotion dataset's confusion matrix, which

clearly shows that the proposed model can identify different emotions through audio clips and achieve better performance results.

	precision	recall	f1-score	support
neutral	0.86	0.88	0.87	49
calm	0.89	0.96	0.92	95
happy	0.95	0.98	0.96	94
sad	0.93	0.89	0.91	95
angry	0.99	0.93	0.96	107
fearful	0.98	0.92	0.95	98
disgust	0.95	0.96	0.95	91
surprised	0.99	0.93	0.96	91
micro avg	0.95	0.93	0.94	720
macro avg	0.94	0.93	0.94	720
weighted avg	0.95	0.93	0.94	720
samples avg	0.93	0.93	0.93	720

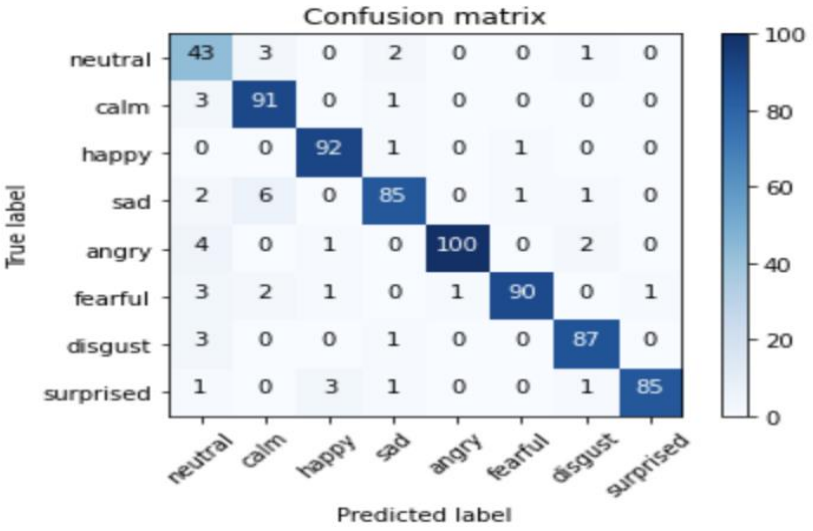


Figure 21: Confusion Matrix Evaluated on the Emotion Dataset.

Results with transfer learning:

In this part, the transfer learning models are adopted. The whole training process involves the same implementation of the proposed framework. Instead of using the proposed CNN architecture, we replaced it with pre-trained models. Figure 22 shows the comparison of these transfer learning models with the data augmentation approach. The best accuracy achieved on the ESC-10 dataset by VGG16 is 99.90%. In

the case of the ESC-50 dataset, the highest achieved by DenseNet is 98.2%. For the Us8k dataset and emotion dataset, the highest was achieved by VGG16 with 97.84% and 94.16%. All the results experimented with high computing resources provided by Google Colab, as mentioned in earlier sections.

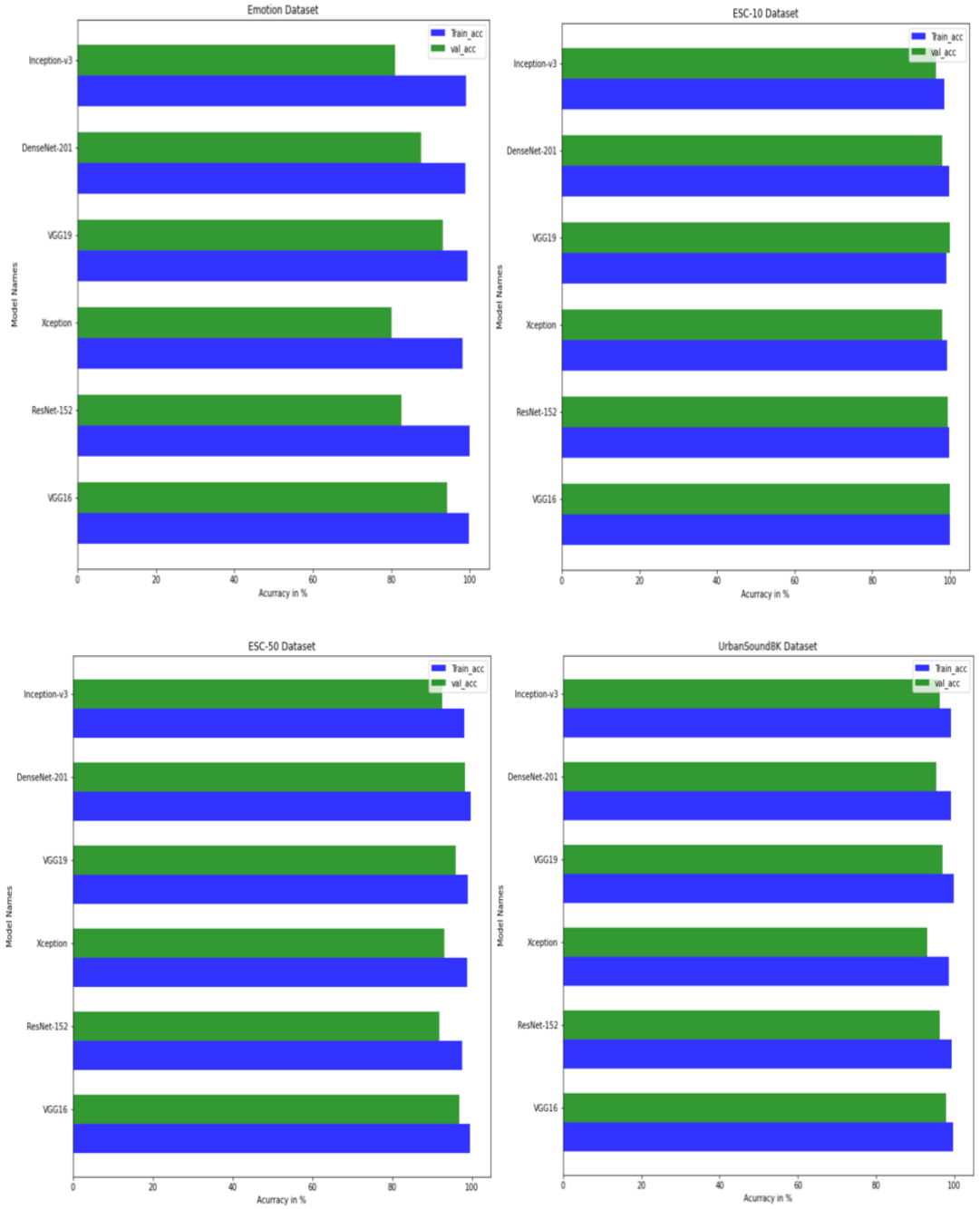


Figure 22: Training and Validation Accuracies on all Datasets with Transfer Learning.

In Table 5, the augmented dataset information is presented. For ESC-10, about 1600 audio clips were generated. For ESC-50, about 10000 audio clips were generated. For Us8k, about 23,732 audio clips were generated. Different approaches implemented by existing models include image-based feature extraction models and audio-based combination techniques. Implementing deep learning models and different augmentation techniques and transfer learning with regularization can overcome many issues while classifying urban sounds [28]. Some data enhancement techniques cannot address the issue, as mentioned in [33], [35]. The features extracted in both waveforms and spectrograms are mentioned in [36].

A detailed comparison of our outcomes with existing approaches is demonstrated in Table 6. Augmentation techniques implemented in existing approaches have an extensive marginal improvement in accuracy and performance. This study approached multiple inputs to describe how multi-model fusion network implementation can find dominant patterns in spectral images. This research exhibits the state-of-the-art and highest accuracy achieved for ESC-10 (99.50%), ESC-50 (97.89%), and Us8k (97.79%).

Table 5. Information of Augmented Datasets.

Datasets	Classes	No of samples	Duration
UrbanSound8K	10	23,732	38.8 hours
ESC-50	50	10000	8.4 hours
ESC-10	10	1600	2.2 hours

Table 6. Comparison of the Proposed Approach with Existing Models.

[References] year	Methodology	ESC-10 in %	ESC-50 in %	Us8k in %
Results of Human Accuracy				
[23] 2015	Human Accuracy	95.7	81.3	–
Results of other's data augmentation techniques and models accuracy				
[33] 2017	DCNN + augmentation	–	–	79.0
[35] 2018	CNN + augmentation + mix-up	91.7	83.9	83.7
[24] 2019	Multi-Channelinput + DCNN-8 + substantial augmentation	97.25	95.50	98.60
[32] 2018	EnvNet-v2 + Augmentation	91.4	84.9	78.3
Results of other's Image-based methodology accuracies				
[26] 2017	Images (Combined features + Google Net)	91	73	93
[31] 2019	Images (CNN + TDSN)	56	49	–
[37] 2020	Pyramidal concatenated CNN	94.8	81.4	78.1
Results of other's audio-based methodology models accuracy				
[34] 2018	Pro-CNN (Combine features)	92.1	82.8	91.9
[25] 2016	Sound Net	92.2	74.2	–
[36] 2018	WaveMsNet	93.7	79.1	–
[28] 2020	Proposed NAA (ResNet-152 + Discriminative learning)	99.04	97.30	99.49
This Study 2020	Multi-Channel Fusion Network	99.50	97.89	97.79

CHAPTER 6

CONCLUSION

This paper proposed a Multi-Channel Fusion network approach with different visual representations of urban sounds such as Spectrogram (SG), Chromagram (CG), and MFCC is introduced for environmental sound classification. Data augmentation for raw input audio files like time-stretching, background noise, and silence removal is adopted to improve accuracy further. As a result, the proposed ESC model achieved state-of-the-art performance on ESC-10 and ESC-50 datasets and competitive performance on the UrbanSound8k dataset. Furthermore, we explored pre-trained models with transfer learning to extract features for urban sound classification. The results of the transfer learning models also gave the best performances. However, the urban noise environment and some other complex challenges still exist for implementation. Future work focuses on network design, distinct input audio features, and conditional class augmentation for performance enhancements.

BIBLIOGRAPHY

[1] R. Radhakrishnan, A. Divakaran, and A. Smaragdis, "Audio Analysis for Surveillance Applications," in IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005. IEEE, 2005, pp. 158–161.

[2] S. Chu, S. Narayanan, C.-C. J. Kuo, and M. J. Mataric, "Where am I? Scene Recognition for Mobile Robots Using Audio Features," in 2006 IEEE International conference on multimedia and expo. IEEE, 2006, pp. 885– 888.

[3] R. Bardeli, D. Wolff, F. Kurth, M. Koch, K.-H. Tauchert and K.-H. From most, "Detecting bird sounds in a complex acoustic environment and application to bioacoustic monitoring," Pattern Recognition Letters, vol. 31, no.12, pp.1524–1534, 2010.

[4] M. B. Dias, "Navpal: Technology solutions for enhancing urban navigation for blind travelers," tech. report CMU-RI-TR-21 Robotics Institute Carnegie Mellon University, 2014.

[5] "Feature learning with deep scattering for urban sound analysis," 2015 European Signal Processing Conference, Nice, France, Aug. 2015.

[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998

[7] J. Piczak, Environmental sound classification with convolutional neural networks. IEEE International Workshop on Machine Learning for Signal Processing, Boston, USA (2015)

[8] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," Proceedings of the 22nd ACM international conference on Multimedia, pp. 1041-1044, 2014, [online] Available: https://doi.org/10.1145/2647868.2655045.

[9] Y. LeCun, "Backpropagation Applied to Handwritten Zip Code Recognition," Neural Computation, 1, pp. 541–551, 1989.

[10] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time-series," in The Handbook of Brain Theory and NeuralNetworks, M. A. Arbib, ed. [Online]. Available: http://www.iro.umontreal.ca/~lisa/pointeurs/handbook-convo.pdf

[11] P. Jain,"Complete Guide of Activation Functions," [Online] Available: https://towardsdatascience.com/complete-guide-of-activation-functions 34076e95d044.

[12] C. Francois. "Keras", [Online] Available:https://github.com/fchollet/keras, 2015.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenetlarge scale visual recognition challenge, 2014, [online] Available:http://www.image-net.org/.

[14] S. Liu, W. Deng, "Very Deep Convolutional Neural Network Based Image Classification Using Small Training Sample Size," 3rd IAPR Asian Conference on Pattern Recognition (ACPR), November 2015.

[15] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition," arXiv preprint arXiv:1512.03385v1, December 2015.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, "Rethinking the inception architecture for computer vision," Proc. IEEE Conference Computer Vision Pattern Recognition, pp. 2818-2826, June 2016.

[17] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," Proceedings of the IEEE CVPR, pp. 1251-1258, 2017.

[18] G. Huang, Z. Liu, L. v. d. Maarten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proc. Comput. Vision Pattern Recognit., 2017, pp., 2261–2269.

[19] J-C. Wang, C-H. Lin, B-W. Chen, M-K. Tsai. (, 2014). Gabor-Based Nonuniform Scale-Frequency Map for Environmental Sound Classification in Home Automation. Automation Science and Engineering, IEEE Transactions on. 11. 607-613. 10.1109/TASE.2013.2285131.

[20] M. M. Mostafa and N. Billor, "Recognition of Western-style musical genres using machine learning techniques," in Expert Systems with Applications, vol. 36, no. 8, pp. 11378–11389, Oct. 2009.

[21] Z. Zhang and B. Schuller, "Semi-supervised learning help in sound event classification," in Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, 2012, pp. 333–336.

[22] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, "Robust Sound Event Classification Using Deep Neural Networks," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 3, pp. 540–552, Mar. 2015.

[23] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in Proceedings of the IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), 2015, pp. 1–6.

[24] J. Sharma, O.-C. Granmo, M. Goodwin, Environment sound classification using multiple feature channels and deep convolutional neural networks
J Latex Cl Files, 14 (8) (2019), pp. 1-11.

[25] Y. Aytar, C. Vondrick, A. Torralba, SoundNet: learning sound representations from unlabeled video no. Nips Adv. Neural Inf. Process. Syst. (2016), pp. 892-900.

[26] V. Boddapati, A. Petef, J. Rasmusson, L. Lundberg, Classifying environmental sounds using image recognition networks, Procedia Comp Sci, 112 (2017), pp. 2048-2056.

[27] Y. Su, K. Zhang, J. Wang, K. Madani, Environment sound slassification using a Two-Stream CNN Based on Decision-Level Fusion. *Sensors (Basel)*. 2019;19(7):1733. Published 2019 Apr 11. doi:10.3390/s19071733

[28] Z. Mushtaq, S-F. Su, Q-V. Tran, Spectral images based environmental sound classification using CNN with meaningful data augmentation, Applied Acoustics, Volume 172, 2021,107581, ISSN 0003-682X, https://doi.org/10.1016/j.apacoust.2020.107581.

[29] B. McFee, et al, librosa: Audio and music signal analysis in Python no. Scipy Proceedings of the 14th Python in Science Conference (2015), pp. 18-24.

[30] M. Abadi et al, "TensorFlow: A System for Large-Scale Machine Learning TensorFlow: A system for large-scale machine learning," 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16), pp. 265-284, 2016.

[31] A. Khamparia, D. Gupta, N.G. Nguyen, A. Khanna, B. Pandey, P. Tiwari, Sound classification using convolutional neural network and tensor deep stacking network IEEE Access, 7 (2019), pp. 7717-7727.

[32] Y. Tokozume, Y. Ushiku, T. Harada, Learning from between-class examples for deep sound recognition ICLR (2018), pp. 1-13.

[33] J. Salamon, J.P. Bello, Deep convolutional neural networks and data augmentation for environmental sound classification IEEE Signal Process Lett, 24 (3) (2017), pp. 279-283.

[34] S. Li, Y. Yao, J. Hu, G. Liu, X. Yao, J. Hu, An Ensemble stacked convolutional neural network model for environmental event sound recognition app Sci, 8 (7) (2018).

[35] Z. Zhang, S. Xu, S. Cao, S. Zhang, Deep Convolutional Neural Network with mix-up for environmental sound classification Chinese Conference on Pattern Recognition and Computer Vision (PRCV) (2018), pp. 356-367.

[36] B. Zhu, et al. Learning environmental sounds with multi-scale convolutional neural network Proceedings of the International Joint Conference on Neural Networks (2018).

[37] F. Demir, M. Turkoglu, M. Aslan, A. Sengur, A new pyramidal concatenated CNN approach for environmental sound classification
Applied Acoustics, 170 (2020), Article 107520, 10.1016/j.apacoust.2020.107520.

VITA

Nagababu Chilukuri completed his bachelor's degree in Computer Science Engineering from JNTUK-University College of Engineering Vizianagaram. He started his master's in computer science at the University of Missouri-Kansas City (UMKC) in January 2019, emphasizing Data Sciences and graduating in December 2020. While studying at UMKC, he worked as a Graduate Assistant for Python/Deep Learning Course and Web-development course.