Master of Science in Computer science
February 2017

Classifying Environmental Sounds with Image Networks

Venkatesh Boddapati

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author(s):
Venkatesh Boddapati
E-mail: vebo15@student.bth.se

External advisors:
Andrej Petef
E-mail: andrej.petef@sonymobile.com

Jim Rasmusson
Email: jim.rasmusson@sonymobile.com

University advisor:
Lars Lundberg
Department of Computer Science and Engineering.

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

ABSTRACT

Context. Environmental Sound Recognition, unlike Speech Recognition, is an area that is still in the developing stages with respect to using Deep Learning methods. Sound can be converted into images by extracting spectrograms and the like. Object Recognition from images using deep Convolutional Neural Networks is a currently developing area holding high promise. The same technique has been studied and applied, but on image representations of sound.

Objectives. In this study, investigation is done to determine the best possible accuracy of performing a sound classification task using existing deep Convolutional Neural Networks by comparing the data pre-processing parameters. Also, a novel method of combining different features into a single image is proposed and its effect tested. Lastly, the performance of an existing network that fuses Convolutional and Recurrent Neural architectures is tested on the selected datasets.

Methods. In this, experiments were conducted to analyze the effects of data pre-processing parameters on the best possible accuracy with two CNNs. Also, experiment was also conducted to determine whether the proposed method of feature combination is beneficial or not. Finally, an experiment to test the performance of a combined network was conducted.

Results. GoogLeNet had the highest classification accuracy of 73% on 50-class dataset and 90-93% on 10-class datasets. The sampling rate and frame length values of the respective datasets which contributed to the high scores are 16kHz, 40ms and 8kHz, 50ms respectively. The proposed combination of features does not improve the classification accuracy. The fused CRNN network could not achieve high accuracy on the selected datasets.

Conclusions. It is concluded that deep networks designed for object recognition can be successfully used to classify environmental sounds and the pre-processing parameters' values determined for achieving best accuracy. The novel method of feature combination does not significantly improve the accuracy when compared to spectrograms alone. The fused network which learns the special and temporal features from spectral images performs poorly in the classification task when compared to the convolutional network alone.

Keywords: Machine Learning, Environmental Sound Classification, Image Classification.

CONTENTS

ABSTRACT

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION 1

1.1 DEEP LEARNING 1

1.1.1 Convolutional Neural Network 2

1.1.2 Recurrent Neural Network 2

2 RELATED WORK 4

2.1 CAFFE 4

2.2 DIGITS 5

2.3 TENSORFLOW 5

2.4 KERAS 5

3 OBJECTIVES 6

3.1 RESEARCH QUESTIONS 6

4 METHODOLOGY 7

4.1 DATA GATHERING: 7

4.2 EXPERIMENT SETUP: 7

4.3 DATA PRE-PROCESSING: 7

4.4 FEATURE SELECTION: 8

4.5 DEEP NEURAL NETWORK SELECTION 9

4.6 EXPERIMENT 10

4.6.1 Preliminary stage: 10

4.6.2 Stage 1: Varying sampling rate 12

4.6.3 Stage 2: Varying frame duration 12

4.6.4 Stage 3: Selecting optimal frame overlap percentage 12

4.6.5 Stage 4: CRP Feature 13

4.6.6 Stage 5: Combining Spectrogram, MFCC and CRP 13

4.6.7 Stage 6: Using CRNN network 14

5 RESULTS 15

5.1 EXPERIMENT PRELIMINARY STAGE: BASELINE 15

5.2 EXPERIMENT STAGE 1: VARYING THE SAMPLING RATE 15

5.3 EXPERIMENT STAGE 2: VARYING THE FRAME LENGTH 18

5.4 EXPERIMENT STAGE 3: SELECTION OF OPTIMAL FRAME OVERLAPS 21

5.5 EXPERIMENT STAGE 4: CLASSIFICATION BASED ON CROSS REFERENCE PLOTS 22

5.6 EXPERIMENT STAGE 5: COMBINING SPECTROGRAM, MFCC AND CRP INTO A COLOR IMAGE FOR CLASSIFICATION 22

5.7 EXPERIMENT STAGE 6: USING CRNN 22

6 ANALYSIS AND DISCUSSION 23

6.1 EXPERIMENT PRELIMINARY STAGE: BASELINE 23

6.2 EXPERIMENT STAGE 1: VARYING THE SAMPLING RATE 23

6.3 EXPERIMENT STAGE 2: VARYING THE FRAME LENGTH 23

6.4 EXPERIMENT STAGE 3: SELECTION OF OPTIMAL FRAME OVERLAPS 24

6.5 EXPERIMENT STAGE 4: CLASSIFICATION BASED ON CROSS REFERENCE PLOTS 25

6.6 EXPERIMENT STAGE 5: COMBINING SPECTROGRAM, MFCC AND CRP INTO A COLOR IMAGE FOR CLASSIFICATION 25
6.7 EXPERIMENT STAGE 6: USING CRNN 25
6.8 THREATS TO VALIDITY 25
6.9 LIMITATIONS 25

7 CONCLUSION AND FUTURE WORK 26

REFERENCES 27

APPENDIX

LIST OF FIGURES

Figure 1: The three types of extracted features (a) Spectrogram, (b) MFCC and (c) CRP of a single sound sample 8

Figure 2: Partitioning of original dataset into 5 folds for cross-validation 11

Figure 3: A separation of color channels into black and white components. 13

Figure 4: The combination of three features in red, green and blue channels respectively to obtain a single color image. 14

Figure 5: A visual comparison of different overlap and frame length settings 21

Figure 6: Graphs showing accuracy obtained by Spectrograms of ESC-10 (left) and ESC-50 (right), on variation of sampling rate. 23

Figure 7: Graphs showing accuracy obtained by MFCCs of ESC-10 (left) and ESC-50 (right), on variation of sampling rate. 23

Figure 8: Graphs showing accuracy obtained by Spectrograms of ESC-10 (left) and ESC-50 (right), on variation of frame length. 24

Figure 9: Graphs showing accuracy obtained by MFCCs of ESC-10 (left) and ESC-50 (right), on variation of frame length 24

LIST OF TABLES

Table 1: Accuracy obtained with baseline settings 15
Table 2: The comparison of prediction accuracy of the networks against different sampling rates on the Spectrogram feature of ESC-10 dataset 15
Table 3: The comparison of prediction accuracy of the networks against different sampling rates on the Spectrogram feature of ESC-50 dataset. 16
Table 4: The comparison of prediction accuracy of the networks against different sampling rates on the MFCC feature of ESC-10 dataset. 17
Table 5: The comparison of prediction accuracy of the networks against different sampling rates on the MFCC feature of ESC-50 dataset 17
Table 6: The comparison of prediction accuracy of the networks against different frame lengths on the Spectrogram feature of ESC-10 dataset. 18
Table 7: The comparison of prediction accuracy of the networks against different frame lengths on the Spectrogram feature of ESC-50 dataset. 19
Table 8: The comparison of prediction accuracy of the networks against different frame lengths on the MFCC feature of ESC-10 dataset 20
Table 9: The comparison of prediction accuracy of the networks against different frame lengths on the MFCC feature of ESC-50 dataset 20
Table 10: Best possible accuracy on the datasets. 22
Table 11: Accuracy with the CRP feature 22
Table 12: Accuracy for the combined feature set 22
Table 13: Accuracy in preliminary trials with different configurations of hyper-parameters 22

Appendix Section

Table 14: The class wise testing accuracies on Spectrogram of ESC-50 dataset 2
Table 15: The class wise testing accuracies on Spectrogram of ESC-10 dataset 2
Table 16: The class wise testing accuracies on Spectrogram of UrbanSound8k dataset 2

1 INTRODUCTION

Sound classification is the process which takes a sound sample as input and gives the respective class label as output. There are many types of classifiers in machine learning paradigm that can perform such a task. Machine learning can be used to effectively classify sound based on the source. Feature extraction algorithms have been the most used concept to classify sound [13].

Automated sound classification has many uses such as remote surveillance, home automation, hands-free communication, etc. There are two broad types of sound in general based on the source— human made and non-human made. Human made sounds consist of speech and other non-vocal sounds. Non-human made sounds include everything else like the sounds made by animals and things [1].

Extensive research has been conducted in the recognition of human speech and implemented in mobile devices with high success [4]. The same cannot be said of the other types of sounds. An interesting application is the use of home monitoring equipment which identifies the different sounds produced in a domestic/interior environment and alerts the user accordingly. The sounds with such a source comprise of people talking, baby crying, dog barking, glass breaking, television or radio, etc. to name a few. The recognition of such domestic sounds, if implemented by using a mobile device, can lead to new venues of ubiquitous computing.

Environmental sounds consist of the various non-human sounds (excluding music) in the course of day-to-day life. Over the past few years, many attempts have been made in recognition of environmental sounds. Presently, there is increasing focus on classifying environmental sounds using deep learning techniques [9][11][13][16][18][21][23]. The improvements in the field of visual scene classification in recent years are leading researchers to start perfecting the task of environmental sound classification. This combination leads to immense leaps in the area of domestic automation for example, an input of both visual and audio cues to a self-navigating robot greatly improves its efficiency [4]. There is also great potential for automated surveillance with both video and audio classification applied to the target scene.

The important difference between speech/music and environmental sound is that the former are strongly structured and clearly demarcated whereas the latter have no common structure [1]. This causes it to be a whole new problem. There is possibly an elegant solution in deep learning since deep neural networks have been proven to be able to handle vast amounts of data and model complex features due to advances in computing power.

Based on the literature study, classification of spectrogram images of sound using deep networks yields the best accuracy rates. The identified research gap of using deep neural networks specially designed for object recognition from images to classify spectrogram images of sound is addressed in this thesis.

1.1 Deep Learning

Deep Learning is about learning multiple levels of representations and abstractions that help to make sense of data [24]. Many layers are used that compute non-linear functions and model highly complex data. Each layer gets its input from a layer before, computes and transforms the data and sends it to the next layer. Every layer consists of neurons that are the fundamental basic units of the network and have various modes of connections to other neurons in the same layer as well as to those of other layers depending on the type of network. Essentially, deep neural networks have been inspired from the human brain architecture.

1.1.1 Convolutional Neural Network

A regular neural network consists of a few layers of neurons where each neuron in one layer is connected to all neurons of the previous and next layers. These fully connected layers form a network. The first layer and last layer are named as “input layer” and “output layer” respectively. The layers in between are known as “hidden layers”. Now, the neural network takes in an input vector and transforms it by passing it through the hidden layers. The output layer calculates the class probabilities for classification.

Such regular neural networks perform heavy computation if the size of data is increased or if the number of layers are increased. Too many parameters are calculated and will result in overfitting without meaningful accuracy. Convolutional neural networks on the other hand don’t have fully connected layers at all levels. Instead the neurons in a layer are connected only to a small region of the layer before it. 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 other regions. This automatically forms a hierarchy of features which increase in abstraction from low-level to high-level when multiple such layers are stacked. This means that the first layer can see only a small portion of the input data and the last layer can see the whole of the input data and draw conclusions from it.

There are three types of layers popularly used in a convolutional network: 1) Convolutional layer, 2) Pooling layer and 3) Fully-connected layer which are detailed below.

1.1.1.1 Convolutional layer

The Convolutional layer takes parameters like the number of filters, size of filters, stride, etc. A filter is a small window that is slid along dimensions of the input data and performs dot products between the values stored in the filter and the input data points. This results in an activation map. During the various iterations the network learns the filters that produce activations when desirable features like an edge are encountered. Moreover, the strong spatially local connections result in the learning of spatial features from images [24].

1.1.1.2 Pooling layer

The main purpose of a pooling layer is to reduce the dimensionality of the input data which reduces the computations and the number of parameters learned, thereby reducing overfitting. Typically, a pooling layer is inserted between Convolutional layers. A pooling layer heavily discards the activations of the previous layers and hence forcing the next convolutional layers to learn from limited variety of data [24].

1.1.1.3 Fully-connected layer

Fully connected layer has neurons that are connected to all neurons of the previous layer as explained before.

1.1.2 Recurrent Neural Network

In this type of neural network, the neurons are connected in such a way that they form directed cycles. By using internal memories RNN can process sequences of information and handle temporal data unlike feed-forward networks [24].

1.1.2.1 Long Short-Term Memory (LSTM)

It is a type of network introduced in 1997 to avoid the vanishing gradient problem through the use of forget gates. Essentially, some of the information is discarded in the process of learning to establish independence from too specific information that does not significantly contribute to the learning process.

1.1.2.2 Gated Recurrent Unit (GRU)

This is a recent contribution that performs similar to LSTM but with far lesser parameters.

The rest of the document is structured in the following way- related work and description of technologies used are detailed in Section 2, aim and objectives of the thesis along with the research questions and their motivations are mentioned in Section 3, the methodology used and the various steps taken to perform the experimentation along with the explanations of each stage of the experiment are present in Section 4, the results corresponding to each of the stages in experiment design are presented in Section 5, the analysis of the results at each stage and their implications are discussed in Section 6 and finally followed by conclusions and exploration of future work in Section 7.

2 RELATED WORK

The aim of this study is to analyze the performance of deep convolutional networks designed for image classification in classifying environmental sounds. To establish the present standards and to have a baseline for comparison, a literature study was conducted as follows. *Wang et al* [10] discuss the efficiency of 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. *Zhang et al* [7] compared the performance of different classifiers and conclude that SVM were the most accurate implementations. *Silva* [5] determines that Sequential Minimal Optimization does better than k-Nearest Neighbor and even SVM.

Deep learning methods have been implemented and tested in relatively few cases, with mobile platforms being almost non-existent use-cases. The most common implementation in this type is to convert the audio signals to image format, spectrum or cepstrum, and then using a neural network to process the image. *Mostafa et al* [2] perform classification of music samples using Probabilistic Neural Network with satisfactory results. *Zhang et al* [4] aptly voices the problem that manual labelling of datasets is very costly and recommends semi-supervised learning as a better solution. *McLoughlin et al* [1] states that classification of sound in realistically noisy environments is challenging and proposes a Deep Neural Network as a viable solution. *Piczak* [6] and *McLoughlin et al* [9] both convey the same idea that Convolutional Neural Network has the best accuracy rates on spectrogram analysis and is best for the case where the training data is limited.

Since sound in the form of an image differs greatly from an actual image, there is still a gap left in the characteristics of different classifiers implemented on a mobile computing platform to classify sound data. The usage of the mobile device's graphics processing unit to speed up the calculations is still a stone left unturned.

As summarized by *Chachada, et al* [18] there are three broad types of processing audio for classification purposes: 1) Framing-based here the audio signals are separated into frames using a Hamming window. Then the features are extracted from each frame and classified separately. The only drawback with this technique is that there is no optimal window size that suits all classes of sounds. Too short window can chop a sound event into multiple frames or a too long window can have two different kinds of events in the same frame. 2) Sub-framing based processing where the frames are further subdivided and each frame is classified based on the majority voting of the sub-frames. This overcomes the limitation of hamming window length. 3) Sequential processing where the audio signals are divided into segments of 30ms with 50% overlap. The classifier then classifies the features extracted from these segments

Piczak et al [6] augmented the training sound samples by adding random delays and class dependent time stretching to the original recordings. The number of augmented variations are 10 and 4 for ESC-10 and ESC-50 [19] sound sets respectively. Framing based processing was followed with spectrogram features having 50% overlaps and which discarded the silent frames. Along with spectrograms, the deltas were computed and fed into the network in two channels.

2.1 Caffe

Reference [22] is the white paper of the tool called Caffe. It is essentially a framework for implementing Deep Neural Networks efficiently and easily with very less programming. The most commonly used types of layers are built-in and take the required parameters specifically or automatically. The framework is written using C++ library with Python and Matlab interfaces. The layers are fully editable and new ones can be created if needed. The best part of this framework is the network design phase. The various layers can be specified in their order and with specific parameters in JSON-like

Protobuf format which just looks like text description with some structure to it. Implementations of training, testing and fine-tuning are in-built. Back-propagation, solver policy, learning rate policy are also implemented with ease by just mentioning them in the deploy document. In terms of speed Caffe provides fastest implementations of the layers and algorithms in present use. Also, there is direct support for using multiple GPUs for faster computing through the integration with CUDA. Furthermore, the python and Matlab bindings can be used to directly modify or create new layers on the fly without changing the framework source code.

2.2 DIGITS

This is a software system developed by NVIDIA specifically for image classification on deep networks that runs in a web interface. It integrates with Caffe and other frameworks and essentially functions as a front end to them. It has functionality to create databases from image folders. Any deep network that can be created with the mentioned frameworks can be implemented and trained in this software. It has tools to tune the training hyper-parameters as well as to visualize the training phase using indicators from the network. Also, it supports GPU based learning with CUDA and cuDNN integration.

2.3 TensorFlow

Reference [29] details the aspects of TensorFlow which is another deep learning framework. The facilities provided are similar to that of Caffe but the execution is very different. It is also very versatile and can enable the creation of very complex architectures and computations with relative ease.

2.4 Keras

This is a front-end package that works with both TensorFlow and Theano as back-ends. This package library contains parameterized implementations of many popular layers used in DNNs. In this thesis Keras is used with TensorFlow as back-end and the tensor representation style of Theano for conducting the CRNN experiment.

3 OBJECTIVES

This thesis was conducted with the aim to analyze the performance of deep machine learning architectures in the classification of environmental sounds. State-of-the-art Convolutional Neural Networks architectures, generally used for image classification and object recognition tasks, were selected to be trained on image representations of sound data. The primary variables are the different parameters used to convert sound data into image data. The effect of various configurations of the variables on the classification accuracy was analyzed. A novel method of combining multiple representations of same audio data in a single image is proposed and its effect on classification accuracy is analyzed.

3.1 Research Questions

RQ1: What is the best classification accuracy achievable on the environmental sound dataset by using state-of-the-art Deep Convolutional Neural Networks?

Motivation: A few works have been done on classifying environmental sounds using new deep network architectures. No work is present that tries to make use of Convolutional Networks which are specifically designed for object recognition. Furthermore, no work was found on evaluation of pre-processing parameters for sound data in the context of spectral image classification. Hence, this thesis explores various combinations of pre-processing parameters in extracting spectral and cepstral images while trying to produce the best possible classification accuracy on two widely popular image classifying networks. A better classification accuracy than previous works in the same audio-image classification process is considered a bonus.

RQ2 Is the combination of features better than any single one of them in producing models with high accuracy?

Motivation: Previous works have tried to combine spectral features and other types of data in two parts of the same image or trained the network with different types of data of each sound successively. This thesis proposes and tests a novel method of combining two types of data in different color channels of the same image. While the combinations in previous works have been beneficial for the performance, it remains to be seen if the new method is useful or not.

RQ3 Is using Recurrent Neural Networks in conjunction with Deep Convolutional Neural Networks beneficial on the same dataset?

Motivation: Since sound is a time-series data there is scope of utilizing RNN to classify sound samples. A recent work was done by fusing CNN and RNN in the same network to classify spectral images. An attempt was made in this thesis to use the same network, with slight modifications to fit the data, and analyze its performance on the combined features dataset.

4 METHODOLOGY

4.1 Data gathering:

One of the main problems with training deep neural architectures in a supervised manner is the amount of computational effort and labeled data required for efficient learning. While the former is in some part addressed on a universal basis by hardware advances and general-purpose GPU computing, the latter is very domain-dependent. Three publicly available datasets were selected for evaluation of the models: *ESC-50* [19], *ESC-10* [19] and *UrbanSound8K* [20].

The *ESC-50* dataset is a collection of 2000 short (5 seconds) environmental recordings comprising 50 equally balanced 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) prearranged into 5 folds for comparable cross-validation.

ESC-10 is a less complex standardized subset of 10 classes (400 recordings) selected from the *ESC-50* dataset (*dog bark*, *rain*, *sea waves*, *baby cry*, *clock tick*, *person sneeze*, *helicopter*, *chainsaw*, *rooster*, *fire crackling*).

UrbanSound8K is a collection of 8732 short (less than 4 seconds) excerpts of various urban sound sources (*air conditioner*, *car horn*, *playing children*, *dog bark*, *drilling*, *engine idling*, *gun shot*, *jackhammer*, *siren*, *street music*) prearranged into 10 folds.

4.2 Experiment setup:

Hardware: Desktop PC
RAM: 12GB
Processor: Intel Core i7-960 (8 cores @3.20 GHz)
Graphic card: NVIDIA GeForce GTX 970
Software:
Operating System: Ubuntu 14.04 LTS
Anaconda python
Deep learning framework: Caffe, TensorFlow (with Keras)
NVIDIA Deep Learning GPU Training System (DIGITS)
MATLAB 2015b

4.3 Data pre-processing:

Since the size of the ESC dataset is too small to use for deep learning, augmentations were made to the original data in the form of time-stretching. Each original audio file was used to produced six additional audio samples with varying degrees of time-stretching applied.

This step was done in Matlab. The Signal Processing Toolbox provides a function named "*resample*" that samples a time-series data such as audio with a new sampling interval which is a factor of the original one. A factor of >1 speeds up the audio whereas that of <1 slows it down. Speed up of the audio also means that its pitch increases and total audio length decreases. The reverse is true for slowdown of the audio. The six values of the factors are 0.6, 0.75, 0.9, 1.1, 1.25 and 1.4. Considering that the length of original audio recordings in the datasets is 5 seconds, this augmentation produces variations in audio length between 3 and 8 seconds.

As a result, there are seven times as many audio samples in the augmented dataset. This augmentation produces variation of pitch and length of audio in the training data which is good for a deep learning network.

The procedure of K-fold cross-validation with k = 5 was followed for experiment stages 1 and 2 (detailed further ahead in Section 4.6). This means the original dataset was divided into 5 subsamples or folds with similar class distribution in each fold. Hence in every network training process four folds were considered as training set and the remaining fold was considered as validation set. Hence there are five variations of training and validation which can be averaged over to get the most generalized results.

4.4 Feature selection:

Audio can be represented in the form of visual images by converting it into Spectrogram, Mel-Frequency Cepstral Coefficients (MFCC), and Cross Recurrence Plot (CRP).

Spectrogram is a representation of the energy in the spectrum of frequencies, of a sound, that varies with time.

Mel-Frequency Cepstral Coefficients are the non-linear representation of the power spectrum of a sound adjusted to log scale.

Cross Recurrence Plot is a matrix visualization where each element represents the distance between the phase trajectories of a time series, such as an audio sample.

This was done in Matlab. The input audio clip is preformatted to convert it into a monophonic signal by adding together half the signal amplitudes of each channel in case it is stereophonic. Next, the audio clip is resampled to a sampling rate as required by the experiment with the order of filters set to 20. Then, the audio clip is trimmed to an even number of samples and normalized. Finally, one of the above features is extracted.

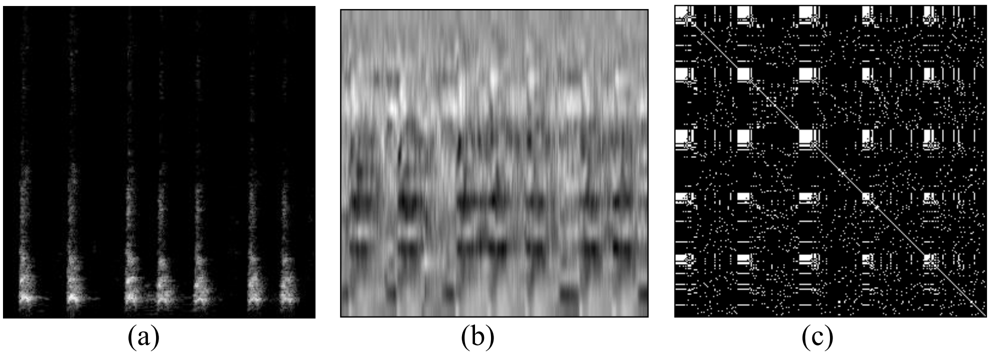

Figure 1: The three types of extracted features (a) Spectrogram, (b) MFCC and (c) CRP of a single sound sample are shown.

The extraction of spectrogram was done using in-built function of Matlab. MFCC was extracted using proprietary Matlab code. CRP was extracted using the CRP Toolbox in Matlab.

4.5 Deep Neural Network selection

AlexNet

Reference [26] is work published to participate in the ILSVRC2012 competition. The authors propose a network design to train on a large database of images belonging to 1000 classes. This design was the winner of that challenge in 2012.

The network is composed of eight layers in total. There are five convolutional layers followed by three fully-connected layers. Each of the first two convolutional layers are followed by Response Normalization layer and Max Pooling layer. ReLU activation is provided after every layer. The data input layer accepts images of size 224 x 224 pixels. Their original specification was designed to be implemented on two GPUs in parallel but DIGITS provides a single GPU implementation. Also, DIGITS provided definition allows for the input image mirroring and random crop of the image as measures of data augmentation as per the original implementation.

However, in this use case the data consists of image representation of sound which is strictly ordered due to its temporal nature. Hence the random mirroring aspect of training data augmentation is disabled. Instead, other dataset specific augmentation is done to the sound dataset itself before its images are fed into this network. The network design used in this thesis can be found at *https://github.com/bkasvenkatesh/Classifying-Environmental-Sounds-with-Image-Networks/blob/master/AlexNet.prototxt*

GoogLeNet

Reference [27] is the work done to participate in ILSVRC2014 competition and is the winner. It is a very deep network with 100 layers, with a depth of 22 layers that employs two new things called Inception layers and embeddings. The inception layers perform local sparse abstractions of the input and the embeddings function in a way similar to pooling layer by reducing the representation size of the input. This architecture has been optimized to perform well on smaller devices with memory constraints. Even though a lot of computation is done, the memory requirement is less.

This network is provided in the DIGITS package too and similar to AlexNet, the mirroring has been disabled for the purpose of this thesis. The network definition used in this thesis can be found at *https://github.com/bkasvenkatesh/Classifying-Environmental-Sounds-with-Image-Networks/blob/master/GoogLeNet.prototxt*

Convolutional Recurrent Neural Network

Reference [28] is the work done on fusing CNN and RNN in the same network. The network comprises of four convolutional layers and two GRU layers. The justification behind this fusion is that the spatially local features can be modelled by the CNN part of the network and the temporal features can be ascertained by the RNN part. This network was difficult to implement at the time in Caffe and hence another framework called “TensorFlow” was used. The original network is designed to accommodate data image size of 96 x 1366 whereas the size used in this thesis is 256 x 256. Also, there was no ready-made implementation of that network and had to be specified in python using “Keras” front-end.

The input images used in the above paper are highly rectangular and hence can be reduced to a strip with square shaped pooling windows. Our input images are square and of different dimensions from those used in the paper and hence the pooling operations are modified in the window shape and size to obtain a similar sized strip at the end of convolution operations which is Nx1x14. This is the only modification done to the network architecture in the interest of fitting our data. The python file used to design and train this network can be found at *https://github.com/bkasvenkatesh/Classifying-Environmental-Sounds-with-Image-Networks/blob/master/CRNN.py*

4.6 Experiment

In a broad sense, the independent variables of this experiment are the input feature dataset and the deep networks while the dependent variable is the classification accuracy obtained on the test set. In a finer sense, the input feature datasets are changed by altering the feature extraction parameters in some stages and so on. While the dependent variable is the same in every stage (testing accuracy), the independent variables vary among the stages and are detailed below.

The independent variables of the experiment stages 1-3 are:

- Sampling rate
- Frame Length
- Overlap percentage
- Input feature
- Deep network

The independent variables of experiment stages 4 and 5 are:

- Input feature
- Deep network

The independent variables of experiment stage 6 are:

- Base learning rate
- Number of learning parameters
- Number of dropout layers
- Input feature
- Deep network

The classification accuracy on the test set is chosen as the dependent variable or the evaluation metric. The deep network is trained on the training subset of the dataset. The trained model is then used to predict the samples in the test subset of the dataset which is as close as it can get to predicting unknown data in the real-world application. The choice of overall classification accuracy as the evaluation metric is justified by its simplicity. The AUC-ROC metric (Area under Curve of Receiver Operating Characteristic) is significantly useful in the case of binary classification, or in the case of skewed distribution of population among the multiple classes in multi-class classification setting and also in the case of variable cut off for the prediction probability. But, in this experiment the class population is evenly distributed in both the training and testing subsets and also, there is no concept of varying cut offs in assigning the classes as the class with highest probability is assigned as the single predicted class for each sample. Hence the overall prediction accuracy has been selected as the evaluation metric for the purpose of this experiment.

4.6.1 Preliminary stage:

Since there was no existing work on the chosen datasets using the selected neural networks, this stage of experimentation was carried out to form a baseline reference to judge the performance of further stages.

In this stage the features, spectrogram and MFCC, were extracted from the original un-augmented sound datasets ESC-10 and ESC-50. Four image datasets were created resulting from the combination of a feature and audio datasets, i.e. two image datasets were created by extracting the spectrogram feature from each of the two sound sets and two more image datasets were created with the MFCC feature in a similar fashion. Each of the 4 datasets were applied to train each of the two networks separately.

As a baseline approach the feature extraction parameters and the network hyper-parameters were all initialized to the most common values which are: -

Extraction parameters:

Sampling rate: 32 kHz
Frame length: 30ms
Overlap percentage: 50%

Hyper-parameters:

Training period: 30 epochs
Base learning rate: 0.01
Solver type: Stochastic Gradient Descent
Learning rate change policy: Exponential decay
Gamma: 0.95

The accuracy measures of the trained models are not so good. This is owing to the fact that the size of the dataset is too small for image deep learning (2000 samples in the bigger ESC-50 dataset which are further split into training and test sets of 80% and 20% samples respectively). After this stage the original sound datasets were prepared for 5-fold cross validation. To achieve this, the sample space was divided into five sub-sample spaces or folds with similar class distribution. Since the datasets are already pre-arranged into 5 folds there is guaranteed uniform class distribution and absolutely no correlation between training and testing data. Hence, one iteration comprised of four folds as training data and the remaining fold as validation data. In each iteration the training data was augmented as explained in section 4.3 and such a combination of augmented training data and original test data in the datasets was used for all further experimentation. This order of this procedure is very crucial as augmenting the dataset before splitting into folds may cause the presence of augmented versions of training samples in the testing fold, resulting in a correlation between training and testing data which produces incorrect (higher) prediction accuracy.

Other factors which may affect the performance accuracy are the feature parameters themselves. It is obvious that when a static image represents a dynamic audio signal there is a loss of information. It then implies that more the information that is captured in an image, better the performance. The parameters of the extracted features can be tweaked so that they contain more information from the audio samples.

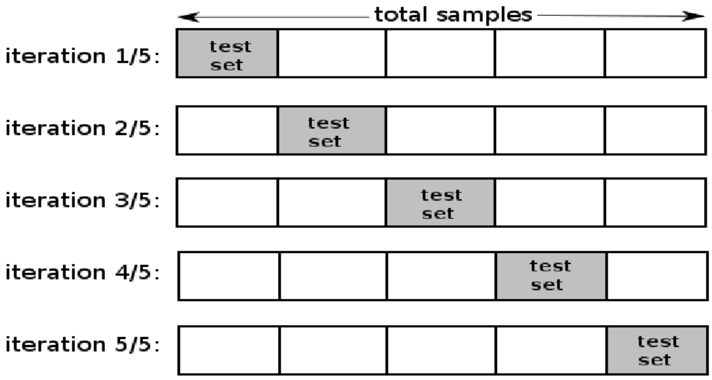

Figure 2: Partitioning of original dataset into 5 folds for cross-validation is shown. *(image taken from www.researchgate.net)*

But more information in the features does not necessarily mean that there will be an increase in the performance of the selected networks. More information also means more processing time for extracting the feature from a sample. Also, the size of the resulting feature image is bigger. Since the input image size for the network is limited to 256 x 256, any feature image bigger than that must be either cropped (again, information loss!) or resized, which is the same as using the parameters for lesser information. Hence determination of the parameter values which result in the optimal trade-off is important. This is worked on in two stages that follow.

4.6.2 Stage 1: Varying sampling rate

Analysis sampling rate determines the range of sound frequencies that are analyzed. The maximum frequency is equal to half the sampling rate. Increasing the sampling rate increases the maximum pitch of the audio that can be represented in the feature. To analyze sound samples of frequency 20kHz (human hearing maxima) a sampling rate of 40kHz is required. But most adults cannot differentiate between sounds of frequencies higher than 16kHz. Hence the sampling rate of 32kHz is reasonable choice for the higher extreme.

This stage compares the performance of the networks on the four augmented datasets by varying just the Sampling Rate of analysis with 3 values: 8kHz, 16kHz and 32kHz. This stage gives an insight into whether the audio content in higher frequency ranges is an important distinguishing factor among different sounds.

5-fold cross-validation is done and the results are averaged over the 5 folds of training and testing.

4.6.3 Stage 2: Varying frame duration

The frame length of an FFT (Fast Fourier Transform) is directly proportional to frame duration and determines the temporal and frequency resolution of the spectrogram and MFCC. Higher frame duration gives higher frequency resolution but lower temporal resolution and vice versa with lower frame duration.

This stage is designed to determine the optimal trade-off between temporal resolution and frequency resolution. Frame duration beyond 50ms result in too low temporal resolution and below 20ms result in too low frequency resolution. Hence this stage compares the performance of the networks on the 4 augmented datasets by varying just the frame duration (in milliseconds) with values 20ms, 30ms, 40ms and 50ms. 5-fold cross-validation is done and the results are averaged over the 5 folds of training and testing.

4.6.4 Stage 3: Selecting optimal frame overlap percentage

Frame overlapping improves the temporal resolution of spectrogram and MFCC. This is a convenient way to preserve to temporal resolution while using higher frame durations. As explained in Stage 2, favoring higher frame duration for higher frequency resolution results in lower resolution on the time axis. Hence, using higher frame overlap with high frame duration should result in a well-balanced spectrogram resolution.

But, as mentioned earlier, the temporal resolution should not be so high as to produce too big images because the downscaling to required size negates any benefit of the higher resolution. Hence, in this stage the sound samples from different classes were used to produce un-scaled spectrogram images with varying frame overlap percentages (0%, 25%, 50% and 75%) resulting from different combinations of frame and overlap durations were visually inspected for desirable resolution.

In order to answer the research question RQ1 the best 30-epoch model will be re-trained with the best values of the parameters for a full duration of 100 epochs to get the best possible accuracy. The highest accuracy obtained before overfitting begins is noted. This time the UrbanSound8K dataset is also used to measure the performance of the networks. Its extraction parameters will be the same as of ESC-10 dataset's respective features' since the generalization is being done on the number of classes in a dataset and both of them have 10 classes.

4.6.5 Stage 4: CRP Feature

Unlike spectrogram and MFCC features which have many similarities, the CRP feature is a totally different concept. It is used in the classification of musical sound as explained in [21]. The CRP toolbox for MATLAB was used to generate the CRP images from the two augmented sound datasets. Computing the recurrence plot is a more complex and hence more time-taking process when compared to spectrogram and MFCC generation. Hence it is suitable for shorter time series rather than the 5s audio clips in the sound dataset. In the initial trials of producing the recurrence plots for single audio clips it was observed that a 5s clip took more than a day to get processed. Whereas a down-sampled (22.05 kHz) clip of less than 0.5s length took around 3s to get processed. On further observation the short clip had 15000 samples and the resulting recurrence plot image had a resolution of 15000 x 15000.

If an even shorter clip having 256 samples were to be used to produce a plot image of size 256 x 256 then there would have been a major loss in the audio signal information. Hence it was decided that a short down-sampled audio clip containing 15000 samples which took 3s to get processed was a reasonable compromise between audio information and computation time.

In order to reduce the 5s sound clips from the dataset to the size required to generate CRP, an audio event extraction technique similar to the one in [9], where 3 high energy frames in the spectrogram are recognized and joined together to create an event-only clip, is used. In this case, however, five highest energy sound sample points are recognized and a window of 3000 samples around each of the 5 high points is extracted from the clip and joined together effectively creating an events-only clip containing 15000 samples which is then used to produce the recurrence plot. The resulting images are then scaled down to the appropriate size of 256 x 256. In this way the CRP image dataset is created and used to train the networks.

4.6.6 Stage 5: Combining Spectrogram, MFCC and CRP

A digital color image has 3 color channels- red, green and blue. A signal processing software, like MATLAB, represents an image as a 3-dimensional matrix where the height and width correspond to the respective image dimensions and the depth represents the 3 color channels. Every pixel of the image is converted into an RGB value which is the intensity of each of the primary colors. In effect a color image can be thought of consisting 3 slices corresponding to the 3 color channels.

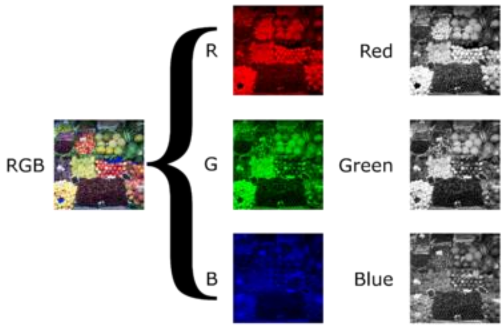

Figure 3: A separation of color channels into black and white components is shown. *(image taken from www.wikipedia.org)*

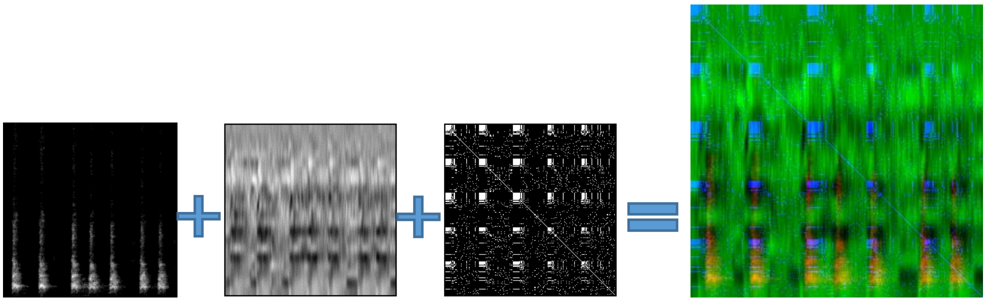

Figure 4: The combination of three features in red, green and blue channels respectively to obtain a single color image is shown.

The spectrogram, MFCC and CRP are all black and white images and hence consist of only one slice instead of three. The representations were combined, each forming a slice, into a color image. The reason behind this step is that fact that the deep networks were designed to analyze color images (3 slices), whereas using just one form of representation forced the networks to process only one slice thereby not making full use of their potential. This combination would likely improve the learning due to more data available in the form of two extra slices to work with.

4.6.7 Stage 6: Using CRNN network

A recent work on music tagging was discovered during the progress of this thesis. The idea of combining a CNN and an RNN in the same network to use the advantages of the respective architectures as explained in Section 3.5 might be beneficial to the scope of this thesis too and thus the motivation in conducting this stage of the experiment. The dataset of combined features as explained above is used with varying network hyper-parameters.

5 RESULTS

The results obtained, the accuracy metric, of the various experiment stages are presented stage-wise in this section. Experiment stages 1 and 2 are conducted with 5-fold cross-validation and hence, the average and standard deviation across the folds are shown too.

5.1 Experiment preliminary stage: Baseline

Dataset	AlexNet	GoogLeNet
ESC-10 Spectrogram	77.1	78.7
ESC-10 MFCC	71.9	74.2
ESC-50 Spectrogram	62.6	65.1
ESC-50 MFCC	44.3	49.8

Table 1: Accuracy obtained with baseline settings

5.2 Experiment stage 1: Varying the sampling rate

Sample rate	Fold	AlexNet	GoogLeNet
8 kHz	1	82.3	85.6
	2	81.9	86.4
	3	82.4	85.1
	4	83.2	84.3
	5	82.6	85
Average		*82.48*	*85.28*
Std. Dev.		0.43	0.69
16 kHz	1	78.5	86.2
	2	76.4	87.5
	3	78.8	86.7
	4	77.1	85.9
	5	78.1	84.6
Average		*77.78*	*86.18*
Std. Dev.		0.89	0.96
32 kHz	1	78.7	79.1
	2	78	79.6
	3	77.5	77.2
	4	78.4	78.5
	5	79.4	79
Average		*78.4*	*78.68*
Std. Dev.		0.64	0.82

Table 2: The comparison of prediction accuracy of the networks against different sampling rates on the Spectrogram feature of ESC-10 dataset is shown

Sample rate	Fold	AlexNet	GoogLeNet
8 kHz	1	67.4	69.3
	2	66.2	64.9
	3	68.8	68.2
	4	67.5	70.7
	5	65.7	68.9
Average		*67.12*	*68.4*
Std. Dev.		1.09	1.93
16 kHz	1	68.6	71.8
	2	68.1	70.1
	3	66.8	71.4
	4	70.3	73.6
	5	69.5	71.5
Average		*68.66*	*71.68*
Std. Dev.		1.19	1.12
32 kHz	1	63	68.1
	2	63.3	68.8
	3	62.7	69.6
	4	64.8	67.4
	5	62.1	65
Average		*63.18*	*67.78*
Std. Dev.		0.90	1.57

Table 3: The comparison of prediction accuracy of the networks against different sampling rates on the Spectrogram feature of ESC-50 dataset is shown.

Sample rate	Fold	AlexNet	GoogLeNet
8 kHz	1	13.5	10
	2	12.4	9.8
	3	13.9	10.5
	4	14.1	11.2
	5	14.6	10.3
Average		*13.7*	*10.36*
Std. Dev.		0.74	0.48
16 kHz	1	77.8	79.3
	2	77.5	77.1
	3	74.6	79.4
	4	76.3	79.3
	5	75.9	76.5
Average		*76.42*	*78.32*
Std. Dev.		1.15	1.26

32 kHz	1 2 3 4 5	72.2 74.4 72.9 72.5 73.1	75 76.1 75.9 77.2 75.4
Average		*73.02*	*75.92*
Std. Dev.		0.76	0.75

Table 4: The comparison of prediction accuracy of the networks against different sampling rates on the MFCC feature of ESC-10 dataset is shown.

Sample rate	**Fold**	**AlexNet**	**GoogLeNet**
8 kHz	1 2 3 4 5	2.2 2.6 3.1 2.9 2.5	1.8 2.3 2.6 2 1.9
Average		*2.66*	*2.12*
Std. Dev.		0.31	0.29
16 kHz	1 2 3 4 5	46.7 47.2 45.8 46.1 46.5	52.7 54.6 53.1 52.5 54.8
Average		*46.46*	*53.54*
Std. Dev.		0.48	0.97
32 kHz	1 2 3 4 5	47.1 43.2 44.7 44.3 45.4	50.6 48.5 49.1 48.6 48.9
Average		*44.94*	*49.14*
Std. Dev.		1.29	0.76

Table 5: The comparison of prediction accuracy of the networks against different sampling rates on the MFCC feature of ESC-50 dataset is shown

5.3 Experiment stage 2: Varying the frame length

Frame length	Fold	AlexNet	GoogLeNet
20ms	1 2 3 4 5	72.5 73.1 72.8 71.9 72.6	68.2 68.4 68.1 68.5 69.3
Average		*72.58*	*68.5*
Std. Dev.		0.39	0.42
30ms	1 2 3 4 5	82.6 82.4 83.6 81.7 81.9	85.1 84.6 84.3 85.4 85.7
Average		*82.44*	*85.02*
Std. Dev.		0.71	0.51
40ms	1 2 3 4 5	83.4 83.8 82.6 82.1 83.5	89.6 88.1 89.4 88.9 87.4
Average		*83.08*	*88.68*
Std. Dev.		0.63	0.82
50ms	1 2 3 4 5	85.3 86.1 84.6 86.4 85.7	90.9 91.3 89.8 90.5 90.2
Average		*85.62*	*90.54*
Std. Dev.		0.63	0.52

Table 6: The comparison of prediction accuracy of the networks against different frame lengths on the Spectrogram feature of ESC-10 dataset is shown.

Frame length	Fold	AlexNet	GoogLeNet
20ms	1 2 3 4 5	67.1 65.4 66.8 67.3 66.9	69.4 70.2 69.6 68.7 68.3
Average		*66.7*	*69.24*
Std. Dev.		0.67	0.67

30ms	1	68.5	71.2
	2	67.9	71.6
	3	68.3	70.1
	4	66.4	68.9
	5	68.6	70.3
Average		*67.94*	*70.42*
Std. Dev.		0.80	0.94
40ms	1	68.1	73.8
	2	66.9	73.2
	3	66.7	72.9
	4	68.5	73.6
	5	67.8	72.5
Average		*67.6*	*73.2*
Std. Dev.		0.69	0.47
50ms	1	65.7	72.3
	2	66.1	70.8
	3	63.9	70.6
	4	66.4	71.3
	5	64.8	72.7
Average		*65.38*	*71.54*
Std. Dev.		0.92	0.83

Table 7: The comparison of prediction accuracy of the networks against different frame lengths on the Spectrogram feature of ESC-50 dataset is shown.

Frame length	Fold	AlexNet	GoogLeNet
20ms	1	73.2	76.3
	2	72.5	75.7
	3	73.6	77.2
	4	73.9	75.1
	5	72.3	76.4
Average		*73.1*	*76.14*
Std. Dev.		0.62	0.71
30ms	1	77.5	79.8
	2	78.2	81.3
	3	77.9	79.5
	4	76.4	80.6
	5	77.1	80
Average		*77.42*	*80.24*
Std. Dev.		0.63	0.64
40ms	1	68.2	76.5
	2	67.4	75.2
	3	69.3	77
	4	68.9	76.3
	5	67.1	75.8

Average		*68.18*	*76.16*
Std. Dev.		0.84	0.62
50ms	1	72.1	74.6
	2	72.3	76.2
	3	71.9	74.8
	4	73.6	75.1
	5	71.4	76.2
Average		*72.26*	*75.38*
Std. Dev.		0.73	0.69

Table 8: The comparison of prediction accuracy of the networks against different frame lengths on the MFCC feature of ESC-10 dataset is shown.

Frame length	Fold	AlexNet	GoogLeNet
20ms	1	45.8	47
	2	44.1	48.3
	3	43.4	46.1
	4	47.6	47.4
	5	46.7	48.1
Average		*45.52*	*47.38*
Std. Dev.		1.57	0.79
30ms	1	46.3	52.8
	2	46.1	51.9
	3	44.8	54.3
	4	45.9	53.6
	5	41.2	53.1
Average		*44.86*	*53.14*
Std. Dev.		1.90	0.80
40ms	1	45.7	50.2
	2	46.4	49.8
	3	45.3	49.6
	4	46.2	51.1
	5	44.9	50.7
Average		*45.7*	*50.28*
Std. Dev.		0.55	0.56
50ms	1	45.5	46.4
	2	46.8	47.1
	3	43.1	45.9
	4	44.3	46.7
	5	43.9	47.5
Average		*44.72*	*46.72*
Std. Dev.		1.29	0.55

Table 9: The comparison of prediction accuracy of the networks against different frame lengths on the MFCC feature of ESC-50 dataset is shown

5.4 Experiment stage 3: Selection of optimal frame overlaps

Samples of different overlap settings from 0% to 75% were visually inspected. 50% overlap with 30ms frame length produced sufficiently good energy distribution in the spectrograms with good enough resolution. While overlap settings less than 50% produced lower resolution images and non-continuous energy distribution, those over 50% produced unnecessarily high resolution spectrogram images which would have be down-scaled anyway.

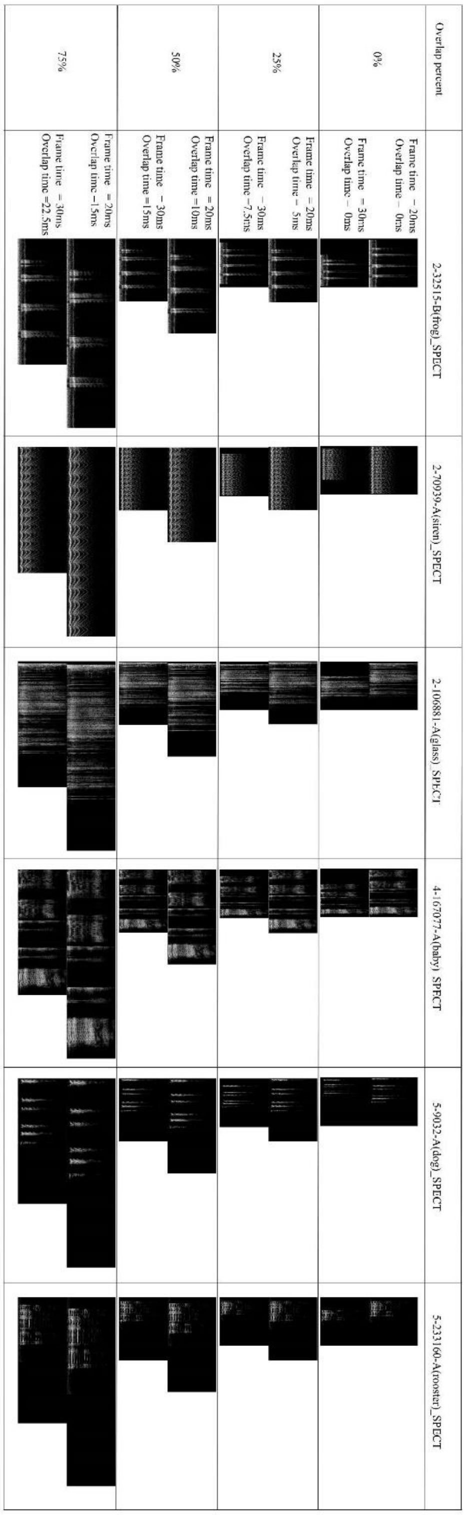

Figure 5: A visual comparison of different overlap and frame length settings

Dataset	AlexNet	GoogLeNet
ESC-50 Spectrogram	68	73
ESC-10 Spectrogram	85	90
UrbanSound8K Spectrogram	90	93

Table 10: Best possible accuracy on the datasets is shown.
(The class-wise accuracies can be found in the Appendix section)

5.5 Experiment stage 4: Classification based on Cross Reference Plots

Dataset	**AlexNet**	**GoogLeNet**
Esc-50 CRP	12.71	10.13
Esc-10 CRP	28.57	27.7

Table 11: Accuracy with the CRP feature is shown

5.6 Experiment stage 5: Combining Spectrogram, MFCC and CRP into a color image for classification

Dataset	**AlexNet**	**GoogLeNet**
Esc-50 Combined	65	73
Esc-10 Combined	86.45	86.25
UrbanSound8K Combined	92	93

Table 12: Accuracy for the combined feature set is shown.

5.7 Experiment stage 6: Using CRNN

Feature set	**No. of parameters**	**Learning rate**	**Accuracy %**
Spectrogram	0.1 million	0.01	60.28
Spectrogram	0.1 million	0.001	40.61
Spectrogram	0.5 million	0.01	60.28
Spectrogram	0.5 million	0.001	40.28
Combined	1 million	0.01	58.22
Combined	0.5 million	0.1	25.2
Combined	0.5 million	0.01 with higher dropout	55.11
Combined	0.5 million	0.01	60

Table 13: Accuracy in preliminary trials with different configurations of hyper-parameters is shown.

6 ANALYSIS AND DISCUSSION

6.1 Experiment preliminary stage: Baseline

The accuracy measures of the trained models are not so good. This is owing to the fact that the size of the dataset is too small for image deep learning (2000 samples in the bigger ESC-50 dataset which are further split into training and test sets of 80% and 20% samples respectively).

6.2 Experiment stage 1: Varying the sampling rate

On the ESC-10 dataset (10 classes) a sampling rate of 8kHz in Spectrogram and 16kHz in MFCC produced models with best accuracy on both the networks trained.

On the ESC-50 dataset (50 classes) a sampling rate of 16kHz proved to be the best in both MFCC and Spectrogram on both networks.

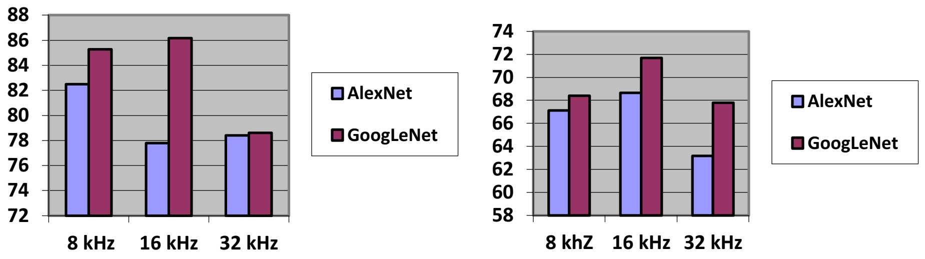

Figure 6: Graphs showing accuracy obtained by Spectrograms of ESC-10 (left) and ESC-50 (right), on variation of sampling rate.

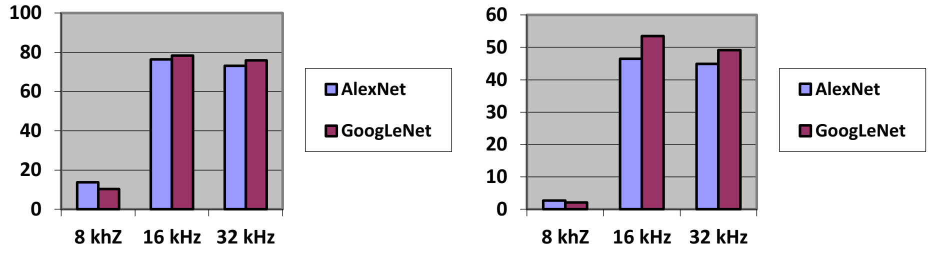

Figure 7: Graphs showing accuracy obtained by MFCCs of ESC-10 (left) and ESC-50 (right), on variation of sampling rate.

6.3 Experiment stage 2: Varying the frame length

The frame length was varied while retaining the best respective sampling rates from the previous stage.

On the ESC-10 dataset (10 classes) a frame length of 50ms on Spectrogram and 30ms on MFCC produced best accuracy on both networks.

On the ESC-50 dataset (50 classes) a frame length of 40ms in Spectrogram and 30ms in MFCC produced best accuracy on both networks.

It is quite evident that the prediction accuracy surpassed that of Piczak's model on both the ESC-50 and UrbanSound8K datasets

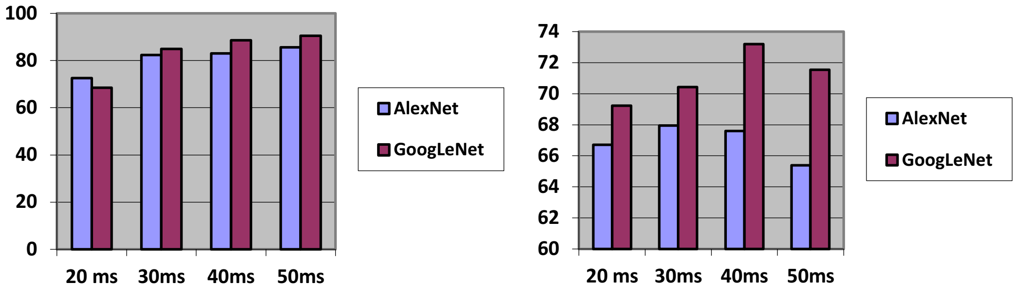

Figure 8: Graphs showing accuracy obtained by Spectrograms of ESC-10 (left) and ESC-50 (right), on variation of frame length.

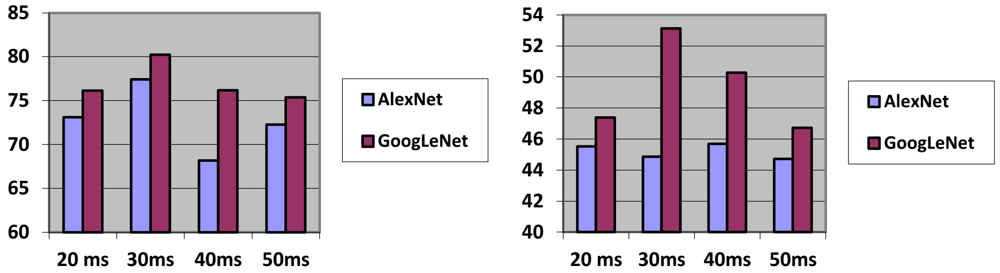

Figure 9: Graphs showing accuracy obtained by MFCCs of ESC-10 (left) and ESC-50 (right), on variation of frame length

6.4 Experiment stage 3: Selection of optimal frame overlaps

An overlap of 50% on 30ms frame length produced an image of dimensions 256 x 332 pixels, which is closer to the target size of 256 x 256. On visual inspection the resolution of the image on the time axis is satisfactory, i.e. not too distorted as is the case with 75% overlap and not too choppy as with 0% overlap. Hence this setting produced images with just the right amount of detailing which does not go to waste during a consequent resize. RQ1 is answered as follows.

The best accuracy on ESC-10 dataset is 90% when trained on GoogLeNet using Spectrogram feature with a frame length of 50ms and sampling rate of 8kHz. The same on AlexNet gives 85% accuracy.

The best accuracy on ESC-50 dataset is 73% when trained on GoogLeNet using Spectrogram feature with a frame length of 40ms and sampling rate of 16kHz. The same on AlexNet gives 68% accuracy.

The best accuracy on UrbanSound8K dataset is 93% when trained on GoogLeNet using Spectrogram feature with a frame length of 50ms and sampling rate of 8kHz. The same on AlexNet gives 90% accuracy.

All the three best accuracies are significantly better than those achieved by Piczak [6] with an improvement of 17.5%, 13.2%, and 26.2% on the three datasets respectively.

6.5 Experiment stage 4: Classification based on Cross Reference Plots

The classification accuracy using just the CRP feature was very low on both the networks. This may be due to the fact that different classes of sounds produced similar-looking CRP images. Hence, using just the CRP feature is not a feasible option.

6.6 Experiment stage 5: Combining Spectrogram, MFCC and CRP into a color image for classification

A combination of the three features did not improve the classification accuracy of both the networks. It can be argued that this combination provided more information for the network to learn than any single feature could and by doing do only managed to increase the dimensionality of data.

RQ2 can be answered with a negative. The assumption that there would be improvement with combination of features is proved wrong.

6.7 Experiment stage 6: Using CRNN

Since there was no improvement in the accuracy of models beyond 60 % with different configurations, this was taken as the final result. The RQ3 is answered with a strong negative.

6.8 Threats to validity

1. Only two pre-processing parameters were explored in experiment stages 1 and 2. Also, all exhaustive combinations of the two parameters were not tested due to time constraint and the long training time of the deep networks. Hence stage 1 focuses on only the first parameter and stage 2 focuses on the second parameter while using the best resulting value of the first parameter from stage 1.
2. Just a few discrete values of the two parameters are tested in stages 1 and 2.
3. A visual inspection approach is chosen to select an optimal overlap percentage parameter in stage 3 instead of statistical analysis due to the availability of expert opinion in the field of signal processing from the external advisors.
4. Some of the existing works chosen during the literature study are not peer reviewed but have interesting ideas.

6.9 Limitations

1. The training time of the deep networks is too high to perform statistical analysis of all possible combinations of the experiment variables.

7 CONCLUSION AND FUTURE WORK

It has been established in this thesis that deep Convolutional Neural Networks, which are designed specifically for object recognition in images, can be successfully trained to classify spectral images of environmental sounds. The best possible classification accuracies on ESC-50, ESC-10 and UrbanSound8K datasets were 73%, 90% and 93% respectively with GoogLeNet architecture. The proposed method of combining different sound features as different color channels of the same image did not improve the classification accuracy. The fusion of Convolutional Neural Network and Recurrent Neural Network in the same deep architecture also could not yield high accuracy.

Future work can be done on testing the feasibility of the implementation of this classification task on a mobile platform with GP-GPU. The target device could essentially be a small portable unit which can be used to recognize environmental sounds in its immediate vicinity. This has immense potential in the field of home security/surveillance, ubiquitous computing, Internet of Things to name a few.

References

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

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

[3] P. Khunarsal, C. Lursinsap, and T. Raicharoen, "Very short time environmental sound classification based on spectrogram pattern matching," *Information Sciences*, vol. 243, pp. 57–74, Sep. 2013.

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

[5] P. Silva, "Classification, Segmentation and Chronological Prediction of Cinematic Sound," 2012, pp. 369–374.

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

[7] L. Lu, H.-J. Zhang, and S. Z. Li, "Content-based audio classification and segmentation by using support vector machines," *Multimedia Systems*, vol. 8, no. 6, pp. 482–492, Apr. 2003.

[8] J. Chen, H. Li, S. Tang, and J. Sun, "A SOM-based probabilistic neural network for classification of ship noises," in *Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on*, 2002, vol. 2, pp. 1209–1212.

[9] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 2015, pp. 559–563.

[10] J.-C. Wang, C.-H. Lin, B.-W. Chen, and M.-K. Tsai, "Gabor-Based Nonuniform Scale-Frequency Map for Environmental Sound Classification in Home Automation," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 607–613, Apr. 2014.

[11] J.-C. Wang, H.-P. Lee, J.-F. Wang, and C.-B. Lin, "Robust Environmental Sound Recognition for Home Automation," *IEEE Transactions on Automation Science and Engineering*, vol. 5, no. 1, pp. 25–31, Jan. 2008.

[12] V. Bountourakis, L. Vrysis, and G. Papanikolaou, "Machine Learning Algorithms for Environmental Sound Recognition: Towards Soundscape Semantics," 2015, pp. 1–7.

[13] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic events using deep neural networks," in *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, 2014, pp. 506–510.

[14] C.-H. Lee, C.-C. Han, and C.-C. Chuang, "Automatic Classification of Bird Species From Their Sounds Using Two-Dimensional Cepstral Coefficients," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1541–1550, Nov. 2008.

[15] S. Scholler and H. Purwins, "Sparse Approximations for Drum Sound Classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 933–940, Sep. 2011.

[16] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic Scene Classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, May 2015.

[17] L. Xue and F. Su, "Auditory scene classification with deep belief network," in *MultiMedia Modeling*, 2015, pp. 348–359.

[18] S. Chachada and C.-C. Jay Kuo, "Environmental Sound Recognition: A Survey," *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2013.

[19] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2015, in press.

[20] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 1041–1044.

[21] T. Park and T. Lee, "Musical instrument sound classification with deep convolutional neural network using feature fusion approach," *arXiv preprint arXiv:1512.07370*, 2015.

[22] Y. Jia et al., "Caffe: Convolutional architecture for fast feature embedding," in Proceedings of the 22nd ACM international conference on Multimedia, 2014, pp. 675–678.

[23] S. Paul, L. Singh, and others, "A review on advances in deep learning," in *Computational Intelligence: Theories, Applications and Future Directions (WCI), 2015 IEEE Workshop on*, 2015, pp. 1–6.

[24] L. Deng and D. Yu, "Deep Learning: Methods and Applications," in *Foundations and trends in Signal Processing,* 2014, pp 192-387.

[25] S. Yadav and S. Shukla, "Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification," 2016, pp. 78–83.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[27] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[28] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional Recurrent Neural Networks for Music Classification," *arXiv preprint arXiv:1609.04243*, 2016.

[29] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

APPENDIX

The class wise accuracies of the best implementations are shown below.

Class name	Per-class accuracy
101 - Dog	100.0%
102 - Rooster	100.0%
103 - Pig	25.0%
104 - Cow	50.0%
105 - Frog	62.5%
106 - Cat	50.0%
107 - Hen	62.5%
108 - Insects	62.5%
109 - Sheep	100.0%
110 - Crow	100.0%
201 - Rain	50.0%
202 - Sea waves	87.5%
203 - Crackling fire	50.0%
204 - Crickets	87.5%
205 - Chirping birds	62.5%
206 - Water drops	62.5%
207 - Wind	25.0%
208 - Pouring water	87.5%
209 - Toilet flush	100.0%
210 - Thunderstorm	62.5%
301 - Crying baby	62.5%
302 - Sneezing	75.0%
303 - Clapping	100.0%
304 - Breathing	37.5%
305 - Coughing	75.0%
306 - Footsteps	75.0%
307 - Laughing	25.0%
308 - Brushing teeth	75.0%
309 - Snoring	50.0%
310 - Drinking - sipping	50.0%
401 - Door knock	87.5%
402 - Mouse click	75.0%
403 - Keyboard typing	37.5%
404 - Door - wood creaks	62.5%
405 - Can opening	100.0%
406 - Washing machine	37.5%
407 - Vacuum cleaner	87.5%
408 - Clock alarm	100.0%
409 - Clock tick	100.0%
410 - Glass breaking	87.5%
501 - Helicopter	25.0%
502 - Chainsaw	87.5%
503 - Siren	87.5%
504 - Car horn	75.0%
505 - Engine	50.0%
506 - Train	50.0%

507 - Church bells	100.0%
508 - Airplane	50.0%
509 - Fireworks	37.5%
510 - Hand saw	62.5%

Table 14: The class wise testing accuracies on Spectrogram of ESC-50 dataset

Class name	Per-class accuracy
001 - Dog bark	75.0%
002 - Rain	75.0%
003 - Sea waves	75.0%
004 - Baby cry	87.5%
005 - Clock tick	87.5%
006 - Person sneeze	87.5%
007 - Helicopter	87.5%
008 - Chainsaw	100.0%
009 - Rooster	100.0%
010 - Fire crackling	100.0%

Table 15: The class wise testing accuracies on Spectrogram of ESC-10 dataset

Class name	Per-class accuracy
air conditioner	90.4%
car horn	90.65%
children playing	78.0%
dog bark	93.2%
drilling	94.0%
engine idling	78.0%
gun shot	95.7%
jackhammer	81.2%
siren	93.1%
street music	90.0%

Table 16: The class wise testing accuracies on Spectrogram of UrbanSound8k dataset