Sound Event Classification

Undergraduate Thesis

Submitted in partial fulfillment of the requirements of
BITS F421T Thesis

By

Sahil Jain
ID No. 2017A8TS0641H

Under the supervision of:

Dr. Chng Eng Siong
&
Dr. Rajesh Kumar Tripathy

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, HYDERABAD CAMPUS
July 2021

Certificate

This is to certify that the thesis entitled, "*Sound Event Classification*" and submitted by Sahil Jain ID No. 2017A8TS0641H in partial fulfillment of the requirements of BITS F421T Thesis embodies the work done by him under my supervision.

Supervisor	*Co-Supervisor*
Dr. Chng Eng Siong	Dr. Rajesh Kumar Tripathy
Associate Professor,	Asst. Professor,
NTU, Singapore	BITS-Pilani Hyderabad Campus
Date:	Date:

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, HYDERABAD CAMPUS

Abstract

Bachelor of Engineering (Hons.)

Sound Event Classification

by Sahil Jain

Sound event classification refers to the problem of detecting the presence of certain events in a given audio clip. It is a vital part of various systems such as noise pollution identification, surveillance, urban soundscape analysis etc. and has gained a lot of traction in recent years. The aim of this thesis is to understand and explore the components that comprise a typical sound event classification system pipeline, target areas with potential for improvement and propose improvements to the current systems. In particular, the thesis aims to reproduce several state-of-the-art systems, verify their results with the published values, suggest improvements to the systems and carry out experiments to validate the proposed changes.

Acknowledgements

I would like to express my sincere gratitude to Dr. Chng Eng Siong, Dr. Pham Van Tung and Andrew Koh of NTU, Singapore for their supervision throughout the project. Their patience and guidance has been invaluable to me and has taught me countless lessons about research. I would like to thank Dr Rajesh Kumar Tripathy at BITS Pilani, Hyderabad for his co-supervision and readiness to help. His cooperation made the remote thesis a smooth experience. I would also like to thank my fellow teammate Lee Yan Zhen for helping me out with her technical know-how and for sharing the difficulties of this project. I am grateful to BITS Pilani for providing me with this opportunity to pursue an off campus research project. Lastly, I would like to thank my family and friends for their continued support throughout this project.

Contents

Chapter 1

Introduction

1.1 Introduction

The sound modality contains important information about the environment and is vital to humans to perceive their surroundings and the changes that happen to it. The environment can be thought of as consisting of two parts - the scene (the inside of a bus, a public park, a coffee shop etc.) and the events (a siren, speech, a dog barking etc). Sound event classification refers to detecting sound events and identifying the categories they belong to in a given audio recording. Sound event classification has potential applications in surveillance [6], monitoring of noise pollution in cities [4], multimedia information retrieval [26] and health monitoring systems [20]. With the success of deep learning in various fields, sound event classification has gained interest in recent years from the signal processing and machine learning communities. Sound event detection is one of many audio classification scenarios such as acoustic scene classification in which a recording needs to be classified into a set a scenes such as a bus, public park etc., sound event localization in which the angle of arrival of sound events is to be calculated and finally, sound event detection with event time-boundary prediction. In this thesis however, only the task of detecting the presence of sound events has been considered. Furthermore, the use case where the input audio is of low quality has been given special consideration.

1.2 Datasets

1.2.1 DCASE 2019 Task 5

This dataset was published for the DCASE 2019 Task 5 urban sound tagging competition. It was provided by SONYC (Sounds of New York City) as a development dataset to build innovative machine listening systems [4]). The dataset contains 8 coarse grained sources of noise which

TABLE 1.1: Distribution of sound events in the DCASE 2019 Task 5 dataset

Class	Training instances	Validation instances
Engine	1269	223
Machinery impact	848	88
Non machinery impact	276	73
Powered saw	244	23
Alert signal	549	86
Music	125	25
Human voice	778	199
Dog	151	6

TABLE 1.2: Distribution of sound events in the Audioset subset

Class	Number of Samples
Breaking	307
Crying/Sobbing	1600
Emergency Vehicle	1011
Explosion	4993
Gunshot/Gunfire	1913
Motor Vehicle	3356
Screaming	7821
Siren	943

are further divided into a total of 23 fine grained classes. Each recording lasts 10 seconds. The aim of the challenge is to develop a system which can detect the presence of each of these noise sources. However, since detecting the presence of 23 classes with many of them being closely related (eg, bike engine vs car engine) deserves special attention, we focus on the simpler problem of coarse grained classification. The distribution of the coarse grained classes in the dataset is given in Table 1.1. Since each audio can contain multiple sound events, the problem can be formulated as a multi-label binary classification problem. The primary evaluation metric for this dataset is the area under precision-recall curve (AUPRC).

1.2.2 Audioset

Audioset [8] is an audio dataset provided by Google consisting of more than 500 audio classes and 2 million human annotated audio recordings. For the project, a subset of Audioset containing 10 different classes is used. The distribution of the audio classes in this subset is given in Table 1.2. The 'others' class contains audio clips from multiple classes from the Audioset ontology other than the other 9 classes. This dataset is split into five folds and all systems are evaluated using five fold cross validation on this dataset. Note that although the original Audioset ontology contains multiple labels for each audio clip, only one label has been considered for each audio file in this subset. This may introduce some label noise in the system. The primary evaluation metric for this dataset is F1-score.

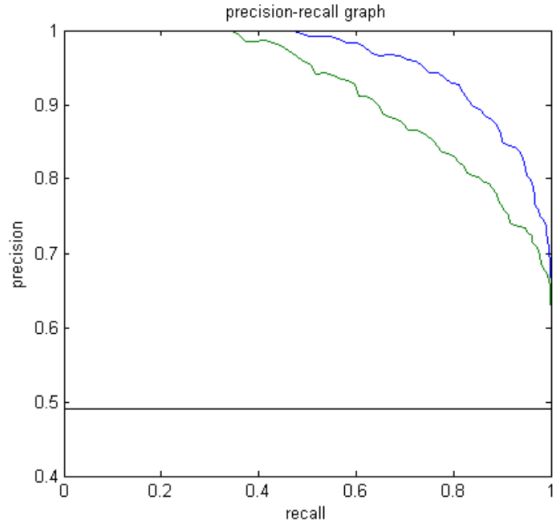
1.3 Evaluation Metrics

The evaluation metrics used throughout this thesis include area under precision-recall curve (AUPRC) and F1 score. These are described below.

1.3.1 Area Under Precision-Recall Curve (AUPRC)

Precision and recall are standard evaluation metrics for classification tasks. Ideally, both metrics should have high values but in practice, a trade-off exists. Furthermore, for binary classification tasks, a classification threshold τ needs to be set and precision and recall values can only be computed for a certain threshold. A better metric for binary classification tasks is AUPRC. To evaluate this metric, precision and recall are plotted for $\tau \in [0, 1]$. The area under this curve is evaluated and the system with the higher area is deemed to be better. To get a better understanding, consider the Fig 1.1. It can be observed that the system denoted by the blue curve can have a higher value of both precision and recall than the green curve for a certain threshold. Correspondingly, it has as higher area under the curve. More details regarding the calculation of this metric can be found on the official website of the DCASE 2019 Challenge Task 5.

FIGURE 1.1: Two example precision-recall curves



1.3.2 F1 Score

F1 score is a standard evaluation metric for classification tasks and is defined as the harmonic mean of precision and recall. Mathematically, it is defined as

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{1.1}$$

A classification threshold τ has to be set before the F1 score can be evaluated. The F1 score depends on the ratio of positive to negative test cases thus care needs to be taken when comparing F1 score across datasets.

1.3.3 Micro Average vs Macro Average

The metrics described above can be calculated using micro average and macro average. Macro averaging treats all the classes equally. It is done by computing the metric individually for each class and then averaging. On the other hand, the true positives, true negative, false positives and false negatives for all the classes are considered together to evaluate the metric. In this thesis, "micro" and "macro" have been mentioned wherever necessary. Wherever not mentioned, the metric has been evaluated using micro averaging.

1.4 Organization

This chapter introduced the problem statement and described the datasets used along with their evaluation metrics. The next chapter gives an overview of a sound event classification system which is followed by a chapter dedicated to the baseline system used for this project. The next few chapters study various aspects of the system with suggestions for improvement.

Chapter 2

Overview of a Sound Event Classification System

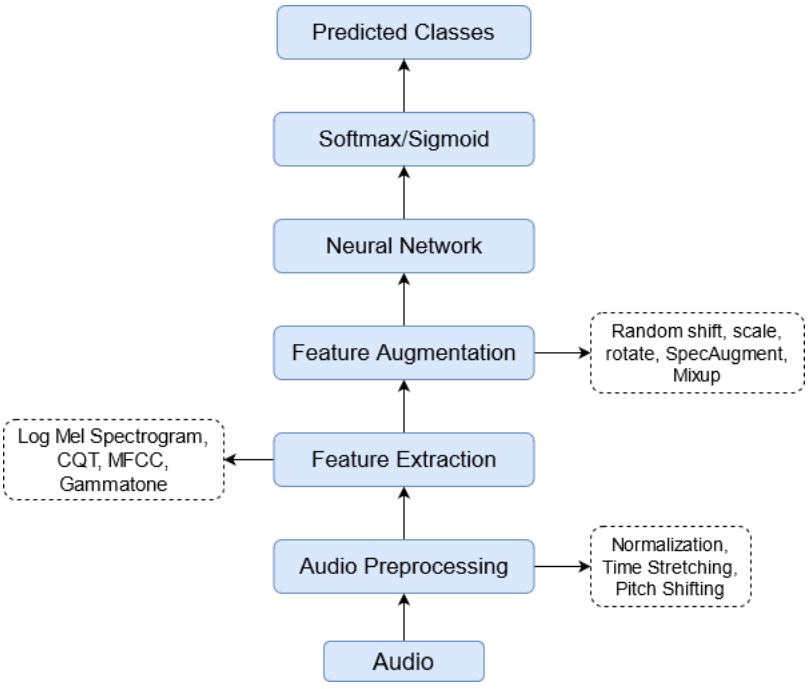


FIGURE 2.1: General pipeline of a sound event classification system

This chapter introduces the various components of a general sound event classification system. These components include audio prepossessing, feature extraction, data augmentation and classification. An overview of the general pipeline is given in Fig. 2.1

2.1 Audio Preprocessing

This stage involves normalization of the audio signal along with data augmentation. The audio signals are generally normalized into a range from $[-1, 1]$ [28]. To prevent overfitting and to circumvent the problem of limited data, several data augmentation schemes are applied at this stage. The two most commonly applied data augmentations used in audio classification are time stretching and pitch shifting [16]. Time stretching refers to changing the speed of a signal by a small amount without affecting the pitch and pitch shifting refers to changing the pitch of the signal by a small amount without affecting the speed. These two transformations are used to generate new audio samples which can be used for training purposes.

2.2 Feature Extraction

Several stationary and non-stationary feature extraction techniques have been used for sound classification. The most popular stationary features include fast Fourier transform (FFT) and Mel frequency cepstral coefficients (MFCC). Non-stationary features include several short-time Fourier transform (STFT) based spectrograms such as log Mel spectrogram, constant-Q transform and gammatone spectrogram. The overall process of generating a STFT based feature is given in the figure below with the difference lying in the FFT analysis portion. For example, log Mel spectrogram applies Mel filters, gammatone spectrograms apply gammatone filters etc.

2.2.1 FFT

For FFT [3], the output is windowed into a fixed number of bands of equal width. Signal power from each band is averaged to generate a single feature. These values are used as the input for classification systems.

2.2.2 MFCC

To generate MFCCs [14], the signal is first divided into a user controlled number of sections and a Hamming window is applied. Next, a mel frequency filter bank is applied to each section. The output of the filter bank produced a series of values which are fed into a cepstral coefficient formula to generate the MFCC feature vector.

2.2.3 Constant-Q Transform

The constant-Q transform (CQT) [15] is also a STFT based spectrogram transform. In CQT, the frequency bins are geometrically spaced and the Q-factors (ratios of the center frequencies to band-widths) of all bins are equal. The generated spectrum has a frequency axis of log scale instead of linear scale. The frequency resolution is better for low frequencies and the time resolution is better for high frequencies. It is more suitable for analysis of music signals.

2.2.4 Log Mel Spectrogram

To generate log Mel spectrograms [14] , the audio signal is first separated into segments. The FFT is calculated for each segment and the frequency scale is transformed to the Mel scale. The frequency spectrum is grouped into several frequency bins and log transformation is applied to the magnitude. It has been found across various studies that generally, log Mel features tend to outperform CQT and MFCC features, although CQT features perform well for some models [11].

2.2.5 Gammatone Filter

The gammatone filter [7] is a linear filter described by an impulse response. It is the product of a gamma distribution and sinusoidal tone and provides a closer approximation to the bandwidths of filters in the human auditory system. Recently, gammatone spectrograms have had success in audio classification tasks [17].

2.3 Data Augmentation

To deal with the problem of limited data, several data augmentation methods are used in sound event classification systems. Signal augmentation methods such as time stretching and pitch shifting are performed in the audio preprocessing stage whereas other methods such as random translate, scale, rotate, time and frequency masking, mixup etc. are applied on the extracted features. Since this is an extensive topic, it is dealt with separately in Chapter 4.

2.4 Classification

Due to the success of convolutional neural networks (CNNs) in image classification tasks, they have received widespread attention in sound event classification as well. In most systems, either the raw audio waveform [22] or a spectrogram based representation is used as an input to a CNN

system. Other than CNNs, recurrent neural networks (RNNs) have also been widely applied for this task either alone or in combination with CNNs. Several systems use a segment of large networks pre-trained on millions of samples as a feature extractor for their classifiers. Networks pre-trained on both image data [1] and audio data [12] have shown to be successful for audio classification. Other strategies such as using multiple features simultaneously have also proven to be successful [2]. Ensemble models using multiple networks with either the same or different architectures for their branches have also had success. In each of these architectures, the outputs of the CNN or RNN layers are flattened into one-dimensional vectors. These vectors are passed through fully connected layers with the last fully connected layer containing as many neurons as output classes. Softmax or sigmoid activation is used depending on if the problem is multi-class or multi-label.

Chapter 3

Baseline System

As a baseline system for this thesis, the DCASE 2019 Task 5 first placed submission has been used [1]. The system uses log Mel spectrograms as the features and a modified version of MobileNetV2 as the classifier. Several data augmentation techniques such as mixup, random erasing and random shift were also used.

3.1 Model Architecture

This system uses a modified version of the MobileNetV2 architecture [21] for classification. The MobileNetV2 architecture consists of a convolutional layer which is followed by several bottleneck residual blocks. The output from the residual blocks is then fed through fully connected layers after being spatially averaged. Finally, the output of the fully connected layer is used for classification. As this architecture was designed for images, it expects a three channel input. However, the log mel features for audio classification are only single channeled therefore, a modification must be made to this network. For this task, two convolutional layers are added to the beginning of the network with a kernel size of 1x1. These layers take the single channel input spectrogram and output a 3 channel feature map. This feature map is then fed to the MobileNetV2 model.

3.2 Feature Extraction

This system uses log Mel spectrograms of the audio clips as inputs to the neural network. The number of samples for window size and hop length were 256 and 694 respectively. For computation of the frequency bins, the lowest and highest frequencies were 20 Hz and 20500 Hz. The number of Mel bins was 128. No additional preprocessing steps were performed.

3.3 Data Augmentation

As the provided dataset is small, data augmentation techniques were required to alleviate the issue of insufficient data. The data augmentation techniques applied to the system included, random shift/scale/rotate, grid distortion and random erasing. For random erasing, a random area of the input image was erased with a probability of 0.5. The minimum and maximum percentage of area erased was 0.02 and 0.4 respectively. Mixup, in which new samples and labels are generated by linearly combining two existing samples, was also used. The beta distribution for mixup used 1.0 as its input parameter.

3.4 Training

The optimization objective used was binary cross entropy. The model was trained using the training set and the validation set was used to determine the stopping point. The AMSGrad variant of the Adam optimizer was used to train the network. The learning rate was initialized with a value of 0.001 and was decayed by a factor of 10 if the validation loss did not improve for more than 5 epochs.

3.5 Results

The system was implemented using PyTorch and trained on Google Colab. The performance of the system is given in the Table 3.1. These results are used as a baseline for this dataset.

Metric	Value
Micro AUPRC	0.843
Micro F1 (@ 0.5)	0.587
Macro AUPRC	0.657

TABLE 3.1: Results for the best performing DCASE 2019 Task 5 system

Chapter 4

Data Augmentation

In this chapter, various kinds of data augmentation techniques for audio classification are examined and their performance is evaluated on the DCASE 2019 Task 5 dataset. Broadly, three types of data augmentation can be applied to audio data. These include signal augmentation, spectrogram augmentation and image augmentation. Signal augmentation involves changing the audio signal by techniques such as time stretching and pitch shifting. Spectrogram augmentation includes methods like time masking, frequency masking and time warping. Image augmentation refers to the augmentation techniques typically used in image classification tasks. A spectrogram can be treated as an image and these techniques can then be used.

4.1 Signal augmentation

The most commonly used signal augmentation techniques are time stretching, pitch shifting, noise injection and dynamic range compression. Time stretching is changing the speed of a signal without affecting the pitch. Pitch shifting refers to changing the pitch of a signal without affecting the speed. Noise injection may involve either adding additive white Gaussian noise (AWGN) or some other background noise to the audio signal. Dynamic range compression is used to reduce the volume of loud sounds and amplify quiet sounds.

To determine the effect of pitch shifting, Table 4.1 compares the results obtained when only pitch shifting is used compared to when no data augmentation is used. The detailed class-wise performance is given in Table A1 in the appendix. A random number of semitones in the range [-4, 4] were shifted for each signal. From Table 4.1, it can be observed that pitch shifting leads to an increase in the micro and macro AUPRC of the system. The micro F1 score is lesser when pitch shifting is applied. However, this may simply be due to the classification threshold of 0.5 being sub-optimal. Experiments related to time stretching and noise injection will be carried out in the future.

Metric	No augmentation	Pitch Shifting
Micro AUPRC	0.813	**0.827**
Micro F1 (@ 0.5)	**0.578**	0.529
Macro AUPRC	0.635	**0.645**

TABLE 4.1: Effect of pitch shifting on performance

4.2 Spectrogram Augmentation

Spectrogram augmentation for speech processing was introduced in the SpecAugment paper [19]. SpecAugment includes time masking, frequency masking and time warping. Time and frequency masking refer to masking out continuous time or frequency blocks from the spectrogram and replacing them with either zeros or the average value of the spectrogram. Time warping involves selecting a random point along the horizontal line and warping it a few time steps either to the left or right. Fig. demonstrates the various SpecAugment transformations.

Time shifting of the spectrogram is also a widely used data augmentation technique. Here, the spectrogram is shifted on the time axis by a random number of time steps. The part of the spectrogram which exceeds the boundary of the time axis is wrapped around to the beginning to preserve information.

Table 4.4 compares the performance of the baseline system without any augmentation to a system with spectrogram augmentations. It can be observed that using spectrogram augmentations provides a significant improvement in performance. For the experiments, one time mask and one frequency mask were used. The maximum length of the time and frequency masks was 40 and 30 samples respectively.

	No augmentation	SpecAugment
Micro AUPRC	0.813	**0.833**
Micro F1 (@ 0.5)	**0.578**	**0.601**
Macro AUPRC	0.635	**0.655**

TABLE 4.2: Effect of spectrogram augmentation on performance

4.3 Image Augmentation

Image augmentation techniques consist of transformations such as random shift/scale/rotate, random erasing, grid distortion and mixup. Random shift/scale/rotate are straightforward transformations in which the input spectrogram is either translated, scaled or rotated by a small amount (determined randomly). Random erasing [29] selects a rectangular portion from the spectrogram and replaces it a constant value. Mixup [27] refers to linearly combining the inputs and outputs of two random samples to impose a linear regularization on the model. Grid

distortion, random erasing and mixup are depicted in the images below. Table 4.3 illustrates the improvement in performance by using image augmentation methods on spectrograms.

	No augmentation	Image Augmentation
Micro AUPRC	0.813	**0.843**
Micro F1 (@ 0.5)	**0.578**	**0.587**
Macro AUPRC	0.635	**0.657**

TABLE 4.3: Effect of image augmentation on performance

4.4 Combining Spectrogram and Image Augmentation

It can be observed that time and frequency masking perform a similar operation to random erasing, i.e., both replace a random section of the spectrogram with a single constant value. Also, grid distortion and time warping both work to distort the input spectrogram. Therefore using both pairs of augmentation may be redundant and may instead lead to a loss in information. While replacing grid distortion with time warping doesn't seem to noticeably affect performance, replace random erasing with time and frequency masking seems to be beneficial. This may be due to that fact that time and frequency masking pay special attention to the time-frequency structure of a spectrogram and are able to augment the data more suitably compared to randomly erasing a block. Table 4.4 compares the performance of the baseline system which uses random erasing to when time and frequency masks replace random erasing. It can be observed that there is an improvement in the micro and macro AUPRC values. The decrease in micro F1 may be due to 0.5 being a suboptimal threshold.

	Random Erasing	Time/Frequency Mask
Micro AUPRC	0.843	**0.850**
Micro F1 (@ 0.5)	**0.587**	0.585
Macro AUPRC	0.657	**0.664**

TABLE 4.4: Performance comparison after replacing random erasing with time and frequency masking

4.5 Summary

In this chapter, various data augmentation methods were studied including signal augmentation (time stretching and pitch shifting), spectrogram augmentation (time/frequency masking and time warping) and image augmentation (random shift/scale/rotate, grid distortion, random erasing and mixup). The effects of each of these kinds of augmentation on performance are evaluated. Finally, time and frequency masking are used to replace random erasing in the baseline system leading to an improvement in performance.

Chapter 5

Mixup Variants for Data Augmentation

Mixup is a data augmentation technique in which new samples are created by linearly combining two existing samples. In recent years, several modifications and variations to mixup have been proposed to improve data augmentation. In this chapter, we explore some of these mixup and some of its variants and observe their effect of classification performance.

5.1 Mixup

Mixup was introduced by [27] and proposed the generation of new samples by linearly combining the inputs and outputs of two samples. Let X_1 and X_2 be two random training spectrograms and let Y_1 and Y_2 be their corresponding labels. The generation of a new sample X and new label Y is done by:

$$X = \lambda X_1 + (1 - \lambda) X_2 \tag{5.1}$$

$$Y = \lambda Y_1 + (1 - \lambda) Y_2 \tag{5.2}$$

Here, λ is a number from 0 to 1 generated by a Beta distribution. The probability density function for the Beta distribution is given by:

$$P = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)} \tag{5.3}$$

Both α and β are hyperparameters. In practice, both are generally set to 1. Fig. 5.1 illustrates mixup for spectrograms.

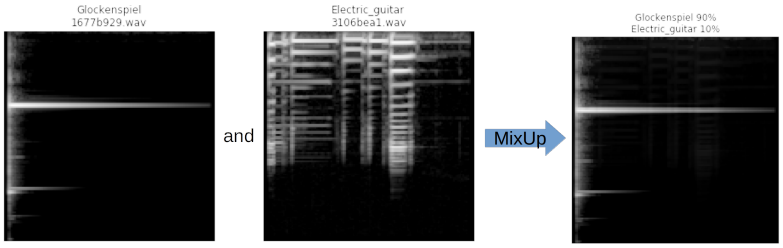


FIGURE 5.1: Mixup

5.2 Specmix

SpecMix [5] can be thought of as a combination of Mixup and SpecAugment. Instead of linearly combining two samples, continuous time blocks and frequency blocks from one spectrogram are replaced by corresponding blocks from another spectrogram. The new labels are generated by linearly combining the two labels, each weighted by the number of pixels of the respective spectrogram in the new spectrogram. In Fig. 5.2, the new label would be 65% harmonica plus 35% electric guitar.

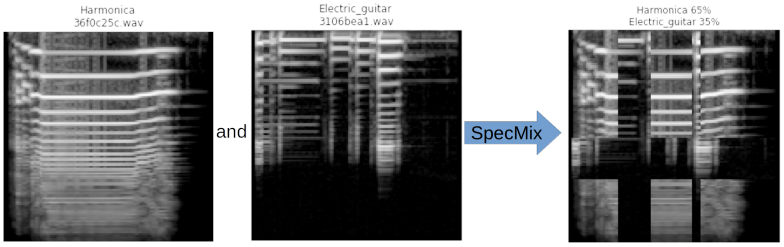


FIGURE 5.2: Specmix augmentation

5.3 Hidden Feature Mixup

In hidden feature mixup [18], instead of applying mixup on the input samples, mixup is instead applied to the feature maps after the first convolutional layer. The new labels are generated as they were for normal mixup, by linearly combining the original labels. Hidden feature mixup is depicted in Fig 5.3.

5.4 Manifold Mixup

Manifold mixup [24] is a generalization of the hidden feature mixup. In this technique, instead of mixing up the inputs of a particular hidden layer, the mixup layer is selected randomly. More formally, the algorithm is described as:

- For every minibatch, select a random layer 'k' from the neural network
- Process the inputs normally till the layer 'k'
- Perform mixup on the outputs of 'k' and continue processing normally

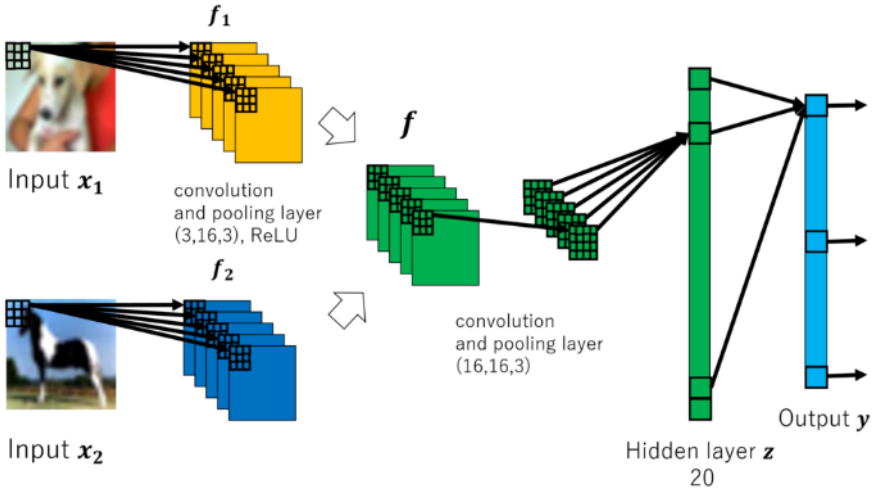


FIGURE 5.3: Hidden feature mixup

5.5 Experiment and Results

All the mixup based augmentation techniques were evaluated for the audio tagging task on the Audioset dataset described in Chapter 1. The F1 scores for the experiments are given in Table 5.1. As a baseline, the baseline system described in Chapter 3 is used with the data augmentation modifications described in Chapter 4. Other than the mixup variants described in this chapter, a method termed 'random selection' was also evaluated in which for every minibatch, a random method from SpecAugment, Mixup, Manifold Mixup and SpecMix was used.

Augmentation	F1 Score
No Mixup	0.8225
Manifold Mixup	0.8462
Mixup	0.8483
SpecMix	0.8475
SpecAugment	0.8537
Random Selection	0.8512
SpecAugment + Mixup	**0.8611**

TABLE 5.1: Effect of mixup variants on performance

From Table 5.1, we can see that every method helps to improve the performance compared to the baseline of "no mixup" by a certain amount. However, the highest increase is observed when

mixup is combined with SpecAugment. With more careful parameter tuning, the performance of the other methods may be improved.

5.6 Summary

In this chapter, different variants of the mixup method were studied and evaluated. These methods include SpecMix, mixup of hidden features and manifold mixup. The performance of each of these methods is evaluated on a subset of Audioset. Although each of these methods helps to improve performance, the combination of SpecAugment and mixup provides the best results however, with more careful hyperparameter tuning, the performance of other methods may be improved.

Chapter 6

Feature Combination

In this chapter, we compare the classification performance of various spectrogram features including log Mel spectrogram, CQT and gammatone spectrograms. Further, a feature fusion based system is proposed to use multiple features for better performance.

6.1 Performance comparison of various features

To evaluate the performance of the different features, the Audioset subset as described in Chapter 1 is used along with the baseline system described in Chapter 3 and data augmentation in Chapter 4. Table 6.1 shows the performance of various features on the audio tagging task using several different metrics. All the features show a similar overall performance. However, from the class-wise performance in Table 6.2 we can see that each feature is better than the others for particular classes. This suggests that combining models of different features may yield better performance.

6.2 Feature combination

To combine the predictions of the different feature models, a weighted average scheme is used. The weights for the different models are based on the AUPRC score of that model on a particular class. The weight for each class for a particular feature is given by:

$$Weight_{feature,class} = \frac{AUPRC_{feature,class}}{\sum_{features} AUPRC_{feature,class}} \tag{6.1}$$

Here, $AUPRC_{feature,class}$ denotes the AUPRC score of that class for that particular feature. The aggregate output for a particular class is given by:

$$Output_{class} = \sum_{feature}^{n} \frac{AUPRC_{feature,class}}{\sum_{features} AUPRC_{feature,class}} \cdot P(class|feature) \tag{6.2}$$

Where $P(class|feature)$ is the softmax output for a particular feature and class. The results for this feature combination system are given in Table 6.1 and 6.2. It can be observed that the feature combination model outperforms the individual feature models.

Model	Accuracy	Precision	Recall	F1-Score
Log-mel	0.854	0.852	0.854	0.852
CQT	0.852	0.851	0.852	0.850
Gammatone	0.859	0.858	0.859	0.858
Combined	0.869	0.868	0.869	0.867

TABLE 6.1: Performance comparison of various features

Class	AUPRC Score			
	Log-mel	CQT	Gammatone	Combined
breaking	0.546	0.451	0.574	0.611
chatter	0.972	0.968	0.974	0.981
crying_sobbing	0.905	0.901	0.916	0.930
emergency_vehicle	0.881	0.886	0.869	0.890
explosion	0.788	0.775	0.797	0.820
gunshot_gunfire	0.926	0.922	0.937	0.947
motor_vehicle_road	0.962	0.965	0.962	0.971
screaming	0.782	0.770	0.787	0.824
siren	0.910	0.915	0.907	0.924

TABLE 6.2: Class-wise performance comparison

6.3 Summary

In this chapter, the performance of different time-frequency representations including log Mel, gammatone and CQT is evaluated. It was observed that each representation is better at predicting the presence of certain classes. To this effect, a feature combination system was proposed to leverage the information provided by each representation and improve performance. For improvement, a joint finteuning stage can be added to further finetune the feature combination model. Other than combining the predictions, the activations of the hidden fully connected or convolutional layers may also be fused to form a classifier.

Chapter 7

Temporal Spectral Attention

Temporal-spectral attention [25] was introduced for environmental sound classification task to learn to pay attention to time frames and frequency bands which are more important for classification. In this chapter, we explore the temporal-spectral attention architecture, evaluate its performance on the ESC-50 dataset and the Audioset subset and suggest improvements.

7.1 Architecture

7.1.1 Attention block

Both temporal and spectral attention have the same architecture with the difference being only in the dimension along which pooling is performed. For each attention block, the input feature map of dimension $T \times F \times C$ is passed through a 1×1 convolution for dimension reduction. The output feature map is of dimensions $T \times F$. For temporal attention, average pooling is performed along the frequency dimension and for spectral attention, pooling is performed along the time dimension. The resulting vector is passed through sigmoid activation and then multiplied to the original tensor. Fig. 7.1 and Fig. 7.2 illustrate the temporal and spectral attention blocks.

7.1.2 Combining temporal and spectral attention

The temporal and spectral attention blocks can be combined in series or in parallel. It has been found that the parallel combination seems to provide better performance compared to the series combinations. Parallel temporal-spectral attention is illustrated in Fig 7.3. Three more parameters α, β, γ are introduced. These parameters are trainable and determine which the contributions of each block.

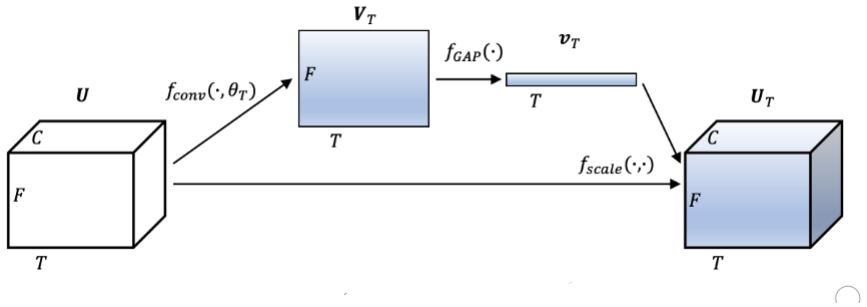


FIGURE 7.1: Temporal attention

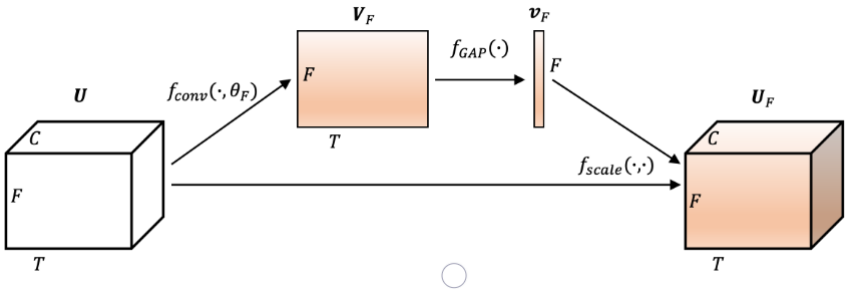


FIGURE 7.2: Spectral attention

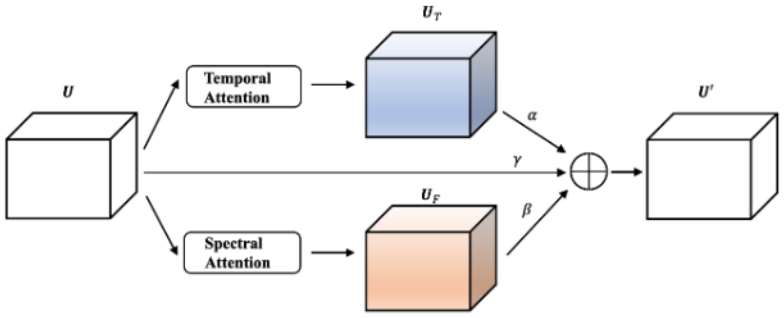


FIGURE 7.3: Parallel temporal-spectral attention

7.2 Experiment and Results

Two datasets were used to test the temporal-spectral attention mechanism. These were the ESC-50 dataset and the subset of Audioset described in Chapter 1. The system used was the same one as described in Chapter 3 with one notable difference - the MobileNetV2 model was replaced by a Cnn10 model [13]. The temporal-spectral attention block was added after every convolutional block in the neural network. The performance of softmax activation in the attention block (replacing sigmoid activation) was also evaluated. The results are given in Table 7.1. From the table, it can be observed that the sigmoid activation based attention block described in [25] led actually led to a drop in performance for both datasets. Instead, using softmax activation helps improve performance.

	ESC-50 (Accuracy)	Audioset subset (F1-score)
No attention	77.25	85.87
Attention (sigmoid activation)	76.25	82.81
Attention (softmax activation)	78.5	86.01

TABLE 7.1: Temporal-spectral attention performance

7.3 Scope for improvement

Several modifications to the temporal-spectral attention architecture can be explored. Two potential modifications to the attention architecture are described below. Other than these two, it may also be worthwhile to explore temporal and spectral attention as two separate mechanisms as they provide different types of information.

7.3.1 Channel-wise attention

The 1×1 convolution flattens the channel dimension to create a global feature map. However, this may lead to a loss in local information. Thus, an attention mechanism can be designed to preserve local channel-wise information by excluding the 1×1 convolution layer. This block can be used in conjunction with the original attention block.

7.3.2 Adding learnable parameters

The temporal-spectral attention block only uses five learnable parameters (two for the 1×1 convolutions, three for the merging parameters). However, as described in [10], fully connected layers with a bottleneck mechanism can be added to enable the attention block to learn more information.

7.4 Summary

In this chapter, the temporal-spectral attention mechanism was studied and its performance was evaluated. The sigmoid activation in the proposed approach was replaced by a softmax activation to improve performance. Furthermore, two potential improvements to the attention architecture were proposed. Further research is required to test the validity of these approaches.

Chapter 8

Improving Performance on Telephony Audio Using Knowledge Distillation

Sound event classification on low quality audio or telephony audio is an important task for surveillance and monitoring systems. As there are no publicly available telephony audio datasets, telephony audio is simulated from high quality audio using several codecs. In this chapter, classification performance on simulated telephony data is compared to the performance on high quality data. Furthermore, a knowledge distillation framework is proposed to improve the performance on telephony data.

8.1 Performance comparison

Telephony audio is simulated for the Audioset subset described in Chapter 1 using several codecs. The resultant audio is of sampling rate 8 kHz compared to the original 22.5 kHz. The performance of the baseline system (described in Chapter 4) on both, the normal and telephony data, is described in Table 8.1. From the table, it can be observed that there is a decrease in precision, recall and F1-score for the telephony data as compared to the normal data. This is expected as simulating telephony data introduces noise and leads to loss in information due to a lower sampling rate.

	Normal data	Telephony data
Precision	**82.02**	80.62
Recall	**82.32**	80.99
F1-Score	**82.03**	80.67

TABLE 8.1: Performance comparison between normal audio and telephony audio

8.2 Improving Performance Using Knowledge Distillation

Knowledge distillation is a model compression technique introduced by [9] to improve the performance of a smaller model by learning from a larger model. In this section, a framework is proposed to improve the performance of a model trained on telephony data by learning from a model trained on higher quality data. The proposed framework is illustrated in Fig. 8.1. The framework consists of 2 models - the teacher model (trained on high quality data), and the student model (trained on telephony data). Initially, the teacher model is trained independently on the high quality data. Thereafter, its weights are frozen and it is used only to generate predictions. The student model learns from two signals - the soft predictions of the teacher model, and the ground truth labels. While training, for every telephony spectrogram received by the student, the corresponding high quality spectrogram is sent to the teacher. The student then learns to reduce the cross-entropy loss between its predictions and the ground truth labels along with reducing the KL divergence between its soft predictions and the soft predictions of the teacher model. The loss function is described below.

$$L_{KD} = \alpha \cdot CE(sm(P_s), L) + (1 - \alpha) \cdot T^2 \cdot KL(sm(\frac{P_s}{T}), sm(\frac{P_t}{T})) \tag{8.1}$$

Here, CE refers to cross-entropy loss, KL refers to KL-divergence, P_s and P_t refer to the student and teacher output before applying softmax respectively. The function sm refers to softmax and T is the softmax temperature to generate soft labels. The parameter α is a hyperparameter to weight the cross-entropy and KL-divergence loss.

8.3 Experiment and Results

The experiments for knowledge distillation were carried out on the Audioset subset. Low quality telephony data was generated using several codecs. The model architecture used was the same model as described in Chapter 3. For knowledge distillation, the values of T and α used were 2 and 0.5 respectively. The model trained on high quality data is referred to as the '22 kHz model' and the model trained on low quality data is referred to as the '8 kHz model'.

From Table 8.2, it can be observed that using knowledge distillation improves the performance of the student model. Surprisingly, it also outperforms the teacher model. This may be due

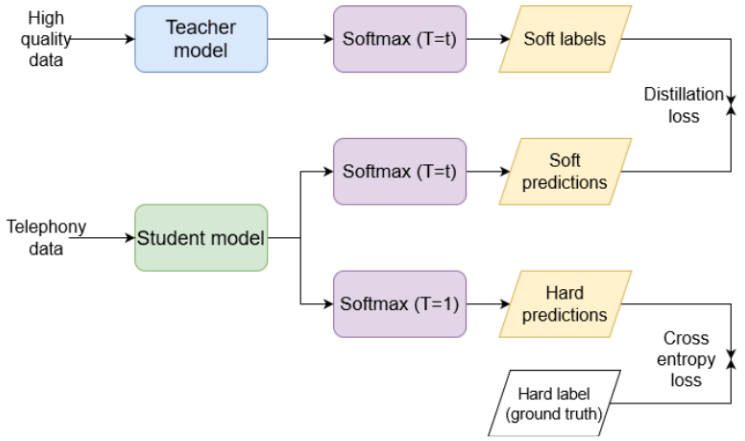


FIGURE 8.1: Knowledge distillation framework

	22 kHz model	8 kHz model	8 kHz model (KD)
Precision	**82.02**	80.62	**82.22**
Recall	**82.32**	80.99	**82.54**
F1-Score	**82.03**	80.67	**82.22**

TABLE 8.2: Performance of knowledge distillation on telephony audio

to the fact that the spectrograms generated for high and low quality data use different cut-off frequencies and thus may contain complementary information. As a next step, an iterative cyclic distillation approach may be tried out where the models acting as the student and teacher are interchanged with every iteration.

8.4 Cyclic Knowledge Distillation

As the performance of the 8 kHz student model after knowledge distillation is better than the 22 kHz teacher model, we can now try using the 8 kHz model as the teacher and the 22 kHz model as the student. More generally, a cyclic distillation framework can be used with the 22 kHz and 8 kHz models alternating as teacher and student. This framework is depicted in Table 8.3 and can be described as follows:

- Initialize both (22 kHz and 8 kHz) models randomly.
- Train the 22 kHz model individually.
- Use 22 kHz model as teacher and train the 8 kHz model as student.
- Use the newly trained 8 kHz model as teacher and the 22 kHz model as student (don't randomly initialize again, use the same model as in the previous step).
- Continue till performance keeps improving.

Iteration	Teacher model	Student model
1	22.050 kHz	8 kHz
2	8 kHz	22.050 kHz
3	22.050 kHz	8 kHz
	⋮	⋮
n-1	22.050 kHz	8 kHz
n	8 kHz	22.050 kHz

TABLE 8.3: Cyclic distillation

The results for cyclic distillation are given in Table 8.4. The parameters α and T were set to 0.5 and 2 respectively. From the table, it is observed that performances improve only till iteration 2, with iteration 3 having performances very similar to iteration 1.

	22 kHz baseline	8 kHz baseline	Iteration 1	Iteration 2	Iteration 3
Precision	82.02	80.94	82.16	82.88	82.16
Recall	82.32	81.23	82.48	83.23	82.51
F1-Score	82.03	80.99	82.17	82.89	82.18

TABLE 8.4: Cyclic distillation results

At the end of this process, the F1-score for the 22 kHz model improved from 82.02 to 82.88 and the 8 kHz model improved from 80.94 to 82.16. Tweaking the knowledge distillation hyperparameters α and T for further iterations may help improve the performance.

8.5 Summary

In this chapter, the performance of the baseline model on telephony data was evaluated. Then, a knowledge distillation based framework was proposed to improve the performance on telephony data using a model trained on higher quality data. The results showed that the framework was successful in improving the performance on telephony data. Surprisingly, the student model trained using knowledge distillation also outperformed the teacher model. To this effect, an iterative cyclic distillation approach was proposed to further improve the effect of knowledge distillation in improving performance for lower quality data.

Chapter 9

Conclusion and Future Work

9.1 Conclusion

This thesis has presented an overview of the Sound Event Classification systems, analysed, verified and reproduced recent research directions and proposed minor improvements.

Chapter 2 laid down the general pipeline of a sound event classification system and discussed the various components. In Chapter 3, the baseline system for this thesis was discussed and implemented and its results verified with the published values.

In Chapter 4, various data augmentation methods were studied and implemented including signal augmentation, spectrogram augmentation and image augmentation. The SpecAugment data augmentation technique was incorporated in the baseline method to yield better results.

In Chapter 5, different variations of the mixup method were studied and implemented including SpecMix, hidden feature mixup and manifold mixup. However, these methods were unable to improve the performance with regular mixup performing the best.

In Chapter 6, the performance of different time-frequency representations (log Mel, CQT and gammatone) was evaluated and a feature combination system using weighted averages was proposed which further yielded an increase in accuracy.

In Chapter 7, temporal-spectral attention was investigated to improve the performance of a convolutional neural network on spectrogram inputs. Further, suggestions to improve the effectiveness of the temporal-spectral attention mechanism were proposed.

In Chapter 8, the performance of the baseline model on telephony audio was evaluated. A knowledge distillation based framework was proposed to improve the performance on telephony audio using a model trained on high quality audio.

9.2 Future Work

The baseline system uses a MobileNetV2 architecture pre-trained on Imagenet. In recent years, various large scale neural network models pre-trained on audio data have been released and fine-tuning of these models on the task data may give better results. Various other architectures such as transformers or other attention mechanism based architectures can also be investigated to further improve the performance.

A new data augmentation method may be developed by combining SpecMix and hidden feature mixup. The challenge here would be making the intermediate feature mixing operations differentiable.

The feature combination system proposed in Chapter 6 trains each feature model separately and combines the results. A joint finetuning stage can be included to further improve the accuracy of the system. Furthermore, combining the soft output vectors or the intermediate features of each feature model can be investigated.

Several suggestions were made to improve the temporal-spectral attention architecture explored in Chapter 7 including channel-wise attention and adding more trainable parameters.

A basic knowledge distillation approach was described in Chapter 8 to improve performance on telephony audio. More advanced feature learning based knowledge distillation approaches can be experimented with (such as [23]).

Bibliography

[1] S. Adapa. "Urban Sound Tagging using Convolutional Neural Networks". In: *ArXiv* abs/1909.12699 (2019).

[2] Jisheng Bai, Chen Chen, and Jianfeng Chen. "A Multi-feature Fusion Based Method For Urban Sound Tagging". In: *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. 2019, pp. 1313–1317. DOI: 10.1109/APSIPAASC47483.2019.9023099.

[3] Mehdi Banitalebi-Dehkordi and Amin Banitalebi-Dehkordi. *Music Genre Classification Using Spectral Analysis and Sparse Representation of the Signals*. 2018. arXiv: 1803.04652 [cs.SD].

[4] Juan P. Bello et al. "SONYC: A System for Monitoring, Analyzing, and Mitigating Urban Noise Pollution". In: *Communications of the ACM* 62.2 (2019), pp. 68–77. DOI: 10.1145/3224204.

[5] Eric Bouteillon. "SPECMIX : A SIMPLE DATA AUGMENTATION AND WARM-UP PIPELINE TO LEVERAGE CLEAN AND NOISY SET FOR EFFICIENT AUDIO TAGGING". In: 2019.

[6] C. Clavel, T. Ehrette, and G. Richard. "Events Detection for an Audio-Based Surveillance System". In: *2005 IEEE International Conference on Multimedia and Expo*. 2005, pp. 1306–1309. DOI: 10.1109/ICME.2005.1521669.

[7] D. P. W. Ellis. *Gammatone-like spectrograms*. http://www.ee.columbia.edu/ dpwe/resources/matlab/gammatonegram/. 2009.

[8] Jort F. Gemmeke et al. "Audio Set: An ontology and human-labeled dataset for audio events". In: *Proc. IEEE ICASSP 2017*. New Orleans, LA, 2017.

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network*. 2015. arXiv: 1503.02531 [stat.ML].

[10] Jie Hu et al. *Squeeze-and-Excitation Networks*. 2019. arXiv: 1709.01507 [cs.CV].

[11] M. Huzaifah. "Comparison of Time-Frequency Representations for Environmental Sound Classification using Convolutional Neural Networks". In: *ArXiv* abs/1706.07156 (2017).

[12] Bongjun Kim. *Convolutional Neural Networks with Transfer Learning for Urban Sound Tagging*. Tech. rep. DCASE2019 Challenge, 2019.

[13] Qiuqiang Kong et al. *PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition*. 2020. arXiv: 1912.10211 [cs.SD].

[14] Juncheng Li et al. *A Comparison of deep learning methods for environmental sound*. 2017. arXiv: 1703.06902 [cs.SD].

[15] Thomas Lidy and Alexander Schindler. "CQT-BASED CONVOLUTIONAL NEURAL NETWORKS FOR AUDIO SCENE CLASSIFICATION". In: Sept. 2016.

[16] Rui Lu. "BIDIRECTIONAL GRU FOR SOUND EVENT DETECTION". In: 2017.

[17] D. Ngo et al. "Sound Context Classification Basing on Join Learning Model and Multi-Spectrogram Features". In: *ArXiv* abs/2005.12779 (2020).

[18] Hideki Oki and T. Kurita. "Mixup of Feature Maps in a Hidden Layer for Training of Convolutional Neural Network". In: *ICONIP*. 2018.

[19] Daniel S. Park et al. "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition". In: *Interspeech 2019* (2019). DOI: 10.21437/interspeech.2019-2680. URL: http://dx.doi.org/10.21437/Interspeech.2019-2680.

[20] Y. Peng et al. "Healthcare audio event classification using Hidden Markov Models and Hierarchical Hidden Markov Models". In: *2009 IEEE International Conference on Multimedia and Expo* (2009), pp. 1218–1221.

[21] Mark Sandler et al. "MobileNetV2: Inverted Residuals and Linear Bottlenecks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4510–4520. DOI: 10.1109/CVPR.2018.00474.

[22] Yuji Tokozume and Tatsuya Harada. "Learning environmental sounds with end-to-end convolutional neural network". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 2721–2725. DOI: 10.1109/ICASSP.2017.7952651.

[23] Frederick Tung and Greg Mori. *Similarity-Preserving Knowledge Distillation*. 2019. arXiv: 1907.09682 [cs.CV].

[24] Vikas Verma et al. *Manifold Mixup: Better Representations by Interpolating Hidden States*. 2019. arXiv: 1806.05236 [stat.ML].

[25] Helin Wang et al. *Environmental Sound Classification with Parallel Temporal-spectral Attention*. 2020. arXiv: 1912.06808 [cs.SD].

[26] Yun Wang, Leonardo Neves, and Florian Metze. "Audio-Based Multimedia Event Detection Using Deep Recurrent Neural Networks". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Shanghai, China: IEEE Press, 2016, 2742–2746. DOI: 10.1109/ICASSP.2016.7472176. URL: https://doi.org/10.1109/ICASSP.2016.7472176.

[27] Hongyi Zhang et al. *mixup: Beyond Empirical Risk Minimization*. 2018. arXiv: 1710.09412 [cs.LG].

[28] Zhichao Zhang et al. *Deep Convolutional Neural Network with Mixup for Environmental Sound Classification*. 2018. arXiv: 1808.08405 [cs.SD].

[29] Zhun Zhong et al. *Random Erasing Data Augmentation*. 2017. arXiv: 1708.04896 [cs.CV].