Universiteit Leiden

Master Computer Science

Data Augmentation for training Audio Classifiers

Name:	Priyansh Jain
Student ID:	s3240630
Date:	[27/08/2023]
Specialisation:	Master's in Computer Science: Data Science
1st supervisor:	Dr. E.M. Bakker
2nd supervisor:	Prof. Dr. M.S.K. Leew

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Acknowledgements

I would like to take a moment to extend my gratitude to my thesis supervisor Dr. E. M. Bakker for his guidance in assisting the writing and research process of my master's thesis. I would also like to thank my parents for their constant support and encouragement. Last but not the least, I would like to certainly thank my friends for sticking with me throughout this effort. It would not have been possible without your support and encouragement.

Abstract

Data augmentation plays a pivotal role in improving the performance of machine learning models, especially in domains where data scarcity is a prevalent issue. In recent years, audio classification has gained significant attention due to its wide range of applications, such as speech analysis, music genre classification, and environmental sound analysis. However, the limited availability of annotated audio data often poses a significant challenge in training accurate and robust audio classifiers.

This paper focuses on the utilization of data augmentation techniques to address the data scarcity problem in audio classification tasks. The objective is to enhance the generalization capability of audio classifiers by artificially expanding the training dataset through various augmentation strategies. The study explores techniques for applying the various available audio augmentation methods such as time stretching, pitch shifting, noise injection, and spectrogram manipulations.

This paper proposes an effective modification called Itermixup of well known data augmentation method Mixmatch. IterMixup uses the multiple iterations of the Mixup Functions using different mixing ratios.The proposed augmentation technique are evaluated on the UrbanSound8K benchmark audio dataset using WideResNet, MobileNetv1 and VGG Neural Networks for classification. The experimental results demonstrate the effectiveness of the proposed data augmentation technique in improving the performance of audio classifiers across the Environmental Sound Classification task.

Table of Contents

Chapter 1

Introduction

In recent years, the realm of audio processing and sound analysis has seen remarkable progress. The advent and evolution of Machine Learning (ML) and Deep Learning (DL) techniques have opened up numerous possibilities and unprecedented avenues in this sphere. The complexity and variability of audio data, though, make it one of the most challenging data types to analyze. A crucial factor that has proven to be highly effective in improving the performance of these audio classification models is data augmentation.

Data augmentation is a strategy that allows us to significantly increase the diversity of data available for training models, without actually collecting new data. In audio classification, it involves the creation of synthetic audio data by adding modifications to the existing data, such as time stretching, pitch shifting, adding noise, or changing volume levels[21]. The primary goal of such techniques is to replicate the kind of variations that could potentially occur in real-world data, thereby ensuring that the model is not only well-trained but is also robust and versatile.

Data augmentation is widely used in machine learning and computer vision[13] to increase the size and diversity of a training dataset by applying various transformations to the existing data samples. The goal of data augmentation is to improve the generalization and robustness of machine learning models by exposing them to a larger variety of data instances. Data augmentation techniques manipulate the training data in ways that preserve the underlying information while introducing realistic variations. This approach helps the model to learn more invariant and robust features, enabling it to perform better on unseen examples during the testing or deployment phase.

Despite the complexity of the task, audio data augmentation has emerged as a potential game-changer in audio-related machine learning applications, contributing to advancements in various sectors including healthcare[5], entertainment[23], surveillance[2], and more. However, understanding and effectively implementing these techniques requires a deep and broad comprehension of both the theory and practical implications involved.By incorporating data augmentation techniques into the training pipeline for audio classification, models can learn more robust representations, generalize better, and achieve improved performance on unseen audio samples, resulting in more accurate and reliable audio classification systems. In this paper, the various technique of applying various well known data augmentation methods for audio, such as Time Stretching, Pitch Shift, Occlusion, Speed Perturbation and CutOut. Here we propose a new data augmentation technique inspired by Mixmatch and show that our proposal increase the performance of the state-of-the-art Neural Network method for the audio classification tasks.

MixMatch[3] has proven to be a highly effective data augmentation approach in semi-supervised learning for the audio classification task, specifically in image classification. MixMatch augments both labeled and unlabeled data by combining consistency regularization and mixup, resulting in improved model performance with limited labeled data. Despite its success in image tasks[3], there is a limited amount of research on MixMatch's potential in the domain of audio data augmentation.
By addressing the dearth of research in audio data augmentation, this thesis aims to contribute to the advancement of audio classification models and foster wider adoption of semi-supervised learning techniques in audio processing applications. IterMixup has the potential to empower audio analysis across diverse domains and lay the foundation for more efficient and accurate audio-based systems. We have also tried to improve the results by adding Time Stretching[19] and Pitch Shifting[16] data augmentation methods.

In this paper the following research question are addressed,

1. What are the current state of the art methods for data augmentation for audio classification task?

2. Does the performance of the current state-of-the-art methods improve using a new data augmentation technique?

Contribution
In this paper we have researched the various data augmentation methods that are already available and in common use for audio classification tasks. We have tried a novel experiment using the state-of-the-art method to improve the results using one of the datasets used in the same method.
Structure
The rest of the paper is structured as follows: Section 2 discusses the previous work on various data augmentation methods. Section 3 discusses the various fundamentals used in the paper like some relevant machine learning algorithms and the basic terminology is explained. Section 4 discusses the methods, setup, dataset and other aspects of the paper. Section 5 discusses the contributions of this paper to the current methods.

Chapter 2

Related Work

The main path for processing in Audio classification includes: data collection and pre-processing, feature extraction , data augmentation, model selection, model architecture,and then training the model using the training data obtained from data augmentation, validation and then evaluation. This is introduced in the paper by J. Solomon et al. [21] and Karol J. Picszak et al.[19].
Although the signal classification for audio has its uniqueness, once it is converted to a 2-d spectrum , we use image processing methods but with constraints by the fact that images are mel-spectrograms instead of just images.

There are various data augmentation methods for images like random cropping, flipping rotations etc. Some new methods like Cutout[7] and Random Erasing [26] have been introduced to solve problems with computer vision. The network is made to focus on the whole image by cutting out a part of the input image, so the accuracy of the method increases. There are also some other methods which use linear mixing like Sample Pairing, Mixup and Between-class learning. Random Erasing[26] was introduced recently where a random part of the image is changed to random values or greyed out. This helps in creating a larger dataset for training and testing while also improving the accuracy of the neural net. In [15], they have tried to implement the Random erasing method on audio where they also introduce a new method called Intra-Class Random erasing where they do now want good features to be lost while performing Random Erasing. They Achieve this by exchanging a randomly selected region of sample with a randomly selected sample of a different sample. This method has also shown some improvement over the already improved Random Erasing. Another way of using Random Erasing was proposed in [14] where they focus on learning robust features of the same class by randomly exchanging randomly selected region from the images. Cutout[7] involves masking out (cutting out) random patches of pixels from an image during training. This process encourages the model to focus on the remaining parts of the image and learn robust features, improving its ability to generalize to new and unseen data.

Mixup[25] can also be used for Audio Classification[6]. But, the temporal nature of the speech or audio signal need special augmentation methods for the audio signals like adding Gaussian noise, time stretch and pitch shift. The latest addition to the augmentation methods for audio signals is SpecAugment[18]. Google proposed SpecAugment[18] which uses time warping , frequency masking and time masking augment audio. The 2-dimensional spectrum diagram of the audio signal is treated as an image with time on the x-axis and frequency on the y-axis.

In [21] published in 2016, J.Salamon et al. proposed a neural network architecture for environmental sound classification and data augmentation method for sound classification. The deep convolutional neural network consists of 3 convolutional layers interleaved with 2 pooling operations, followed by 2 fully connected (dense) layers. The input for the network are Time Frequency patches (TF-pathces) which are extracted using Essentia [4]. For training, cross-entropy loss is determined using stochastic gradient descent.
They tried 4 different deformations for audio augmentation resulting in 5 augmentation sets. The deformations used were, Time Stretching, Pitch Shifting, Dynamic Range compression and Background Noise. The improvement of not only due to the augmented dataset but the combination of the augmented dataset and the deep convolutional model.
Another research paper by [1] et al. discussed the use of laboratory generated dataset. These materials were selected to mimic the generation of squeaks and rattles from automobiles. Now the dataset size was increased using data augmentation techniques like Pitch Shifting, adding Background Noise and Time stretching. Then dataset was passed through a CNN architecture containing 2 convolutional layers, max-pooling and 2 fully connected layers. Then the two fully connected hidden layers have 5000 ReLU each for better feature processing and the output layer has 8 classes. For both of the datasets the best accuracy was obtained by combining the various data augmentation methods. The best accuracy was 97.1% and %97.7 for squeak and rattle datasets respectively for the combined augmentation,

Loris Nanni et al. [17] have used various Data Augmentation approaches . 2The transformations done on the signals were speed scaling, pitch shift,volume change, random noise addition and time shift. This generated abut 10 new signals for each training signal. They also did applied some special augmentation techniques on spectrograms namely Random Shifts, SameClassSum, Vocal Tract Length Normalization (VTLN), TimeShift, ImageWarp, RandomEMDA AUgmenter (Equalized mixture Data Augmentation). Next they tried Signal Augmentation was dome on the raw audio Signals they fallowing methods were applied Wow resampling, adding noise, clipping, speedup, Harmonic Distortion, Gain,TimeShift, soundmix, DynamicRangeCompression and Pitchshift. This resulted in 11 transformed versions of the input signal. The Used the BIRDZ dataset and CAT sound dataset. They used CNNs which are already pretrained like GoogleNEt and VGGNet. The results demonstrated that a combination of various data augmentation methods maximize the performance of the model.

Tom Ko et al. have reported experiments with audio speed perturbation[12] which emulates a combination of tempo perturbation and vocal tract length perturbation. The combination of the above two mentioned methods performed better than them individually. For tempo perturbation the given audio signal $x(t)$ is multiplied by a factor α which gives us $x(\alpha t)$, by using fourier transform on the resulting signal we can see that warping factor produces shifts in the frequency components by an amount proportional to the frequency. The dataset used was the gayle mandarin dataset.

Yidong Wang et al.have publisher a paper in 2022 which proposes a unified semi-supervised learning benchmark called USB[24] for classification tasks in computer vision, natural language processing, and audio domains. The benchmark aims to enable consistent evaluation over multiple datasets from multiple domains and reduce the training cost to make the evaluation of SSL more affordable. The authors evaluate 14 SSL algorithms on 15 tasks across domains and find that pre-training techniques can be helpful in the SSL scenario because it can not only accelerate the training but also improve the generalization performance. The paper also provides an environmentally friendly and low-cost evaluation protocol with pre-training and fine-tuning paradigm, reducing the cost of SSL experiments.

Yuan Gong proposed a novel self-supervised learning framework called Masked Spectrogram Patch Modeling (MSPM)[8] for audio and speech classification. MSPM is a joint discriminative and generative framework that predicts a specific frequency band in a specific time range given the neighboring band and time information. The proposed framework is evaluated on various speech and audio tasks, including audio event classification, keyword spotting, speaker identification, and speech emotion recognition. The experiments demonstrate that MSPM can significantly outperform from-scratch models for all six benchmarks evaluated with an average improvement of 60.9%. The proposed framework can reduce the need for large amounts of labeled data for audio and speech classification.

Rongjie Huang et. al proposed a method called Make-An-Audio[11] for text-to-audio generation using prompt-enhanced diffusion models. The main challenges in audio generation are the lack of high-quality text-audio datasets and the complexity of modeling long continuous audio data. To address these challenges, the authors introduce a pseudo prompt enhancement approach to construct natural languages that align well with audio, allowing the use of unsupervised language-free data. They also use a spectrogram autoencoder to predict self-supervised representations instead of waveforms, ensuring efficient compression and high-level semantic understanding. The paper presents several key contributions, including the Make-An-Audio method, which leverages latent diffusion with a spectrogram autoencoder for modeling long continuous waveforms. They also investigate textual representation and highlight the advantages of contrastive language-audio pretraining. The paper evaluates Make-An-Audio and demonstrates state-of-the-art results through quantitative and qualitative evaluations. Additionally, the authors generalize the model to X-to-Audio generation, enabling the generation of high-definition, high-fidelity audios based on user-defined inputs.

David Bertholet et al.[3] introduced a new data augmentation method which is MixMatch. The MixMatch algorithm has demonstrated state-of-the-art performance in semi-supervised learning tasks and has proven to be effective in leveraging large amounts of unlabeled data to enhance model performance with limited labeled data. It provides a holistic approach that combines data augmentation, mixup, and consistency regularization to create more robust and accurate pseudo-labels for the unlabeled data. Mixmatch gives us an accuracy of 82% which is the state-of-the-art accuracy achieved on the UrbanSound8K dataset. In this paper, we have tried to improve on Mixmatch by running the mixup function with multiple values of lambda.

Chapter 3

Fundamentals

In this section we will talk about the background information and the measures that we have used to evaluate the performance of the methods.

3.1 Evaluation Metrics

In most of the research about audio classification discussed in the 2 uses accuracy, error rate and f-score as metrics to evaluate the performance of the model. To compare our results with the models and methods discussed before, we are also using the same metrics for evaluation,
Accuracy
Accuracy for classification tasks is the ratio of correctly predicted classes to the the total number of predictions. This can be shown by the formula below.

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

Error rate
Error rate is the degree of prediction error of a model made with respect to the true mode. It can also be framed as ratio of incorrectly classified objects to the total number of classifications.

$$ErrorRate = \frac{Number\ of\ incorrect\ classifications}{Total\ number\ of\ classifications}$$

From the above two equations, it can be deduced that:

$$Accuracy = 1 - ErrorRate$$

Precision
Precision is a metric used to evaluate the performance of a machine learning or classification model. It measures the accuracy of the positive predictions made by the model, i.e., the proportion of true positive predictions among all positive predictions made by the model. Where, True Positives (TP) are the instances that are correctly predicted as positive by the model. False Positives (FP) are the instances that are incorrectly predicted as positive by the model when they are actually negative.

$$Precision = \frac{truePosities}{truepositives + falsepositives}$$

Recall

Recall, also known as sensitivity or true positive rate, is a metric used to evaluate the performance of a machine learning or classification model. It measures the proportion of actual positive instances that are correctly identified by the model. Where, True Positives (TP) are the instances that are correctly predicted as positive by the model. False Negatives (FN) are the instances that are incorrectly predicted as negative by the model when they are actually positive.

$$Recall = \frac{truepositives}{truepositives + falsenegatives}$$

F-Score

3.2 Audio Data Representation

In this section we discuss the various audio representation techniques used in the experiments.

Waveform Representation

A waveform is a continuous representation of an audio signal over time. It is obtained by plotting the amplitude of the audio signal against time. Waveform representation retains the original time-domain information but can be challenging to work with directly due to its high dimensionality. An example can be seen in 3.1

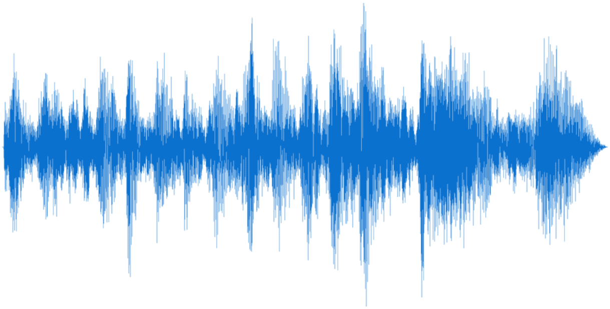


Figure 3.1: Audio Sample as a Waveform

Spectrograms

A spectrogram is a visual representation of the frequencies present in an audio signal over time. It is obtained by applying the Fourier Transform to short segments of the audio signal. Spectrograms provide a 2D representation where time is on the x-axis, frequency is on the y-axis, and color represents the amplitude or energy of the frequency components. 3.2 is an example of a spectrogram.

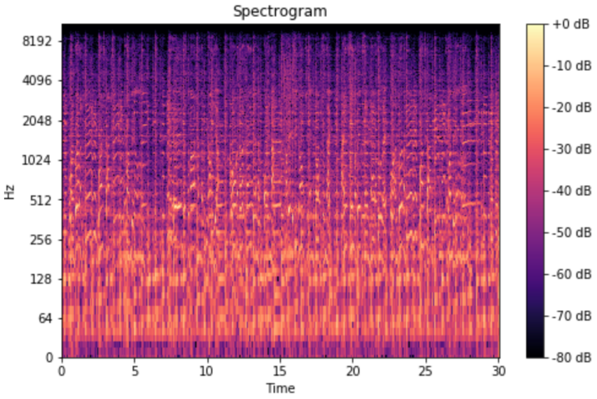


Figure 3.2: Audio Sample as a Spectrogram

Mel-Spectrogram

A mel-spectrogram represents the frequencies over time. However, the frequency scale is transformed to a mel scale, which is more aligned with human auditory perception. A mel-spectrogram can be seen in 3.3

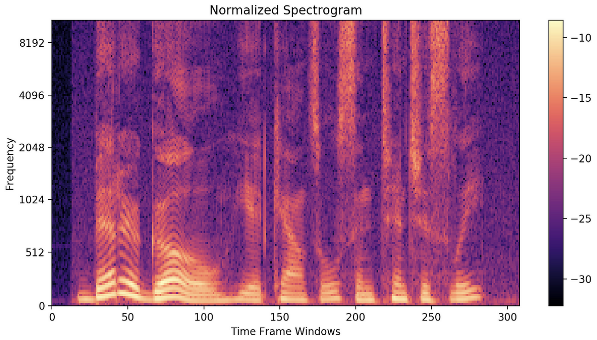


Figure 3.3: Spectrogram Adjusted to a *mel* scale

FFT

The Fast Fourier Transform (FFT) is a computational technique that converts an audio signal from its time-domain representation into a frequency-domain representation. It achieves this by breaking down the signal into different frequency components, revealing the distribution of frequencies and their corresponding amplitudes. The result, known as the FFT spectrum, showcases the strength of various frequencies within the signal. This representation is crucial for tasks such as spectral analysis, music analysis, sound event detection, and audio classification, where it allows for the identification of dominant frequencies, harmonics, and patterns in the audio data.

3.3 Audio Data Augmentation Techniques

In this section we discuss, the various audio augmentation methods used in our experiments.

Occlusion

It consists of setting a segment of the waveform to zero. The size of the segment is randomly chosen up to a user-defined maximum size. The position of the segment is also chosen randomly. Occlusion is applied on the raw audio signal.

CutOut[7]
It sets the values within a random rectangle area with the -80 dB value, which corresponds to the silence energy level in our spectrograms. The length and width of the removed sections are randomly chosen from a predefined interval and depend on the spectrogram size. This is applied to log mel spectrograms

Speed Perturbation[12]
We resample the raw audio signal up (nearest-neighbor upsampling) or down (decimation) according to a rate chosen randomly within a predefined interval. The resulting waveform is either shorter or longer. Padding or cropping is randomly applied at the start and the end of the stretched signal to keep the signal duration constant.

Pitch Shift[16]
Pitch shifting alters the perceived pitch of a sound without significantly changing its duration. It involves modifying the frequency of the audio signal while compensating for changes in playback speed to maintain the original duration. Techniques like resampling, phase vocoding, and granular synthesis are used for pitch shifting. It finds applications in music production, sound design, vocal processing, and audio manipulation, enabling tasks like creating harmonies, changing musical keys, and adding unique effects to audio signals. **Time Stretching**[19]
Time Stretching is the process of altering the duration of an audio signal without impacting its pitch. By expanding or compressing the time axis of an audio waveform, variations of the original samples are created, aiding in enhancing the dataset's diversity. This technique is especially beneficial for models, making them more resilient to variations in speech rates or musical tempos. Several algorithms, such as the phase vocoder and synchronized overlap-add (SOLA), facilitate time stretching, ensuring minimal introduction of artifacts. While it's a potent augmentation method, careful application is crucial to avoid distortions and ensure realistic audio representations.
When weak data augmentations are applied, it means that there is a 50% chance of the augmentation to be applied to the signal. Similarly, when strong data augmentation techniques are applies, there is a 100% chance that the data augmentation will be applied to a signal.

Chapter 4

Methodology

4.1 Neural Networks

4.1.1 WideResNet

For this method we have used the same Neural Network that has been used by Leo Cances et al.[6]. WideResNet-28-2 is a specific convolutional neural network architecture used for image classification tasks.The architecture is an extension of the original ResNet model by Kaiming He et al.[9], which introduced the concept of residual blocks to alleviate the vanishing gradient problem in deep networks. WideResNet further improved the performance by increasing the width (number of channels) of the layers while keeping the depth relatively shallow.
The notation "WideResNet-28-2" refers to the specific configuration of the network:

Layer	Architecture
input	Log mel spectrogram
conv1	BasicBlock(32)
	Max pool
block1	[BasicBlock(32) BasicBlock(32)] x 4
block2	[BasicBlock(32) BasicBlock(32)]x4
block3	[BasicBlock(32) BasicBlock(32)] x4
	Avg pool
	ReLU
	Linear

Table 4.1: WideResNet-28-2 Architecture

The overall architecture typically consists of several residual blocks with a set number of layers, where each block contains two or more convolutional layers and a skip connection to retain the learned features from the previous layers. The skip connections help in the efficient training of deeper networks.
WideResNet-28-2 is known for its ability to achieve competitive accuracy on various image classification datasets while being more computationally efficient than very deep networks like ResNet-152. WideResNet-28-2 often achieves state-of-the-art performance on classification

benchmarks, demonstrating its suitability for various tasks, and its transfer learning capability enhances efficiency. Its performance is optimized through data augmentation techniques and alignment with problem complexity and computational resources. The Wide-ResNet architecture gives about 84% accuracy.

4.1.2 MobileNetV1

MobileNetV1[10] is a lightweight deep learning model architecture introduced by Google in 2017, primarily aimed at mobile and embedded vision applications. Its main innovation is the use of depthwise separable convolutions, which significantly reduce the number of parameters and computations compared to standard convolutions. This factorization of convolution operations into depthwise and pointwise convolutions allows MobileNetV1 to be highly efficient in terms of both model size and speed. Additionally, the model introduces width and resolution multipliers as hyperparameters to further adjust the model's size and computational cost. The specific configuration used for the MobileNetv1 can be seen below: MobileNetV1 has been widely adopted in scenarios where computational resources are limited but competitive accuracy is still desired. Using MobileNetV1 for audio classification might be unconventional, but by converting audio data into a visual format, the model's strengths in image classification can be leveraged in the audio domain.

Layer	Architecture
input	log mel spectrogram
conv1	Basic Block (32)
	Average Pooling
block1	[BasicBlock (32) BasicBlock(32)] x1
block2	[BasicBlock (64) BasicBlock(64)] x1
block3	[BasicBlock (128) BasicBlock(256)] x1
block4	[BasicBlock(256)]x1 [(BasicBlock(512)]x5
block5	[(BasicBlock(512) BasicBlock(1024)]x1
	Mean Pooling
	Max Pooling + Average Pooling
	ReLU

Table 4.2: MobileNetv1 Architecture

4.1.3 VGG

VGG[22] , developed by the Visual Geometry Group at the University of Oxford in 2014, is a deep convolutional neural network designed for image classification. Renowned for its depth, VGG architectures, particularly VGG16 and VGG19, consist of 16 and 19 layers respectively. The model is characterized by its consistent use of 3x3 convolutional filters and 2x2 max-pooling

operations. Despite its architectural simplicity, VGG achieved state-of-the-art performance on the ImageNet challenge during its debut.
The configuration used for the task is: Owing to its ability to extract hierarchical features,

Layer	Architecture
input	Log mel spectrogram
conv1	Basic Block (64)
	Max Pooling
block1	[BasicBlock (64) BasicBlock(64)] x1
block2	[BasicBlock (128) BasicBlock(128) BasicBlock(128)] x1
block3	[BasicBlock (256) BasicBlock(256) BasicBlock(256)] x1
block4	[BasicBlock(512) BasicBlock(512) BasicBlock(512)]x1
block5	[BasicBlock(512) BasicBlock(512) BasicBlock(512)]x1
	Max Pooling
	Softmax

Table 4.3: VGG architecture

VGG remains a popular choice for transfer learning across various domains. However, its depth and dense architecture can make it computationally demanding in terms of processing and memory requirements.

4.2 Mixmatch

Mixmatch was first introduced by David Bertholot et al. [3]. It is a data augmentation method that leverages both labeled and unlabeled data during the training process to improve the performance of the model. The key idea is to create augmented samples from the unlabeled data and then combine them with the augmented unlabeled data to create the pseudolabels. These pseudolabels are then used along side the original labeled data to train the model. Both the labeled and unlabeled data are augmented using strong data augmentation techniques, such as Occlusion, CutOut, and Speed Perturbation as introduced in 3. Occlusion involves introducing occluding patches to the audio spectrogram, simulating real-world noise or obstructions. CutOut entails masking out random patches of the spectrogram, encouraging the model to focus on the remaining features. Speed Perturbation involves changing the playback speed of the audio, capturing variations in speaking rate. These augmentation strategies introduce realistic variations, enhancing the model's ability to generalize across diverse audio instances and improving its robustness against noise and distortions. During the learning phase, each minibatch is composed of labeled x_s and unlabeled x_u samples in equivalent proportions. The

first step consists of applying an augmentation to the labeled part of the mini-batch and K augmentations to the unlabeled part in parallel. These K augmentations are sampled from the three augmentations (weak).
Mixmatch uses a combination of data augmentation techniques and sharpening to create consistent and reliable pseudolabels. The main steps involved in mixmatch are Data Augmentation, Mixup, Consistency Regularization and Loss Function. This process, known as sharpening, enhances the accuracy and reliability of pseudo-labels, making them more informative for training. Sharpening is particularly useful in semi-supervised learning scenarios like MixMatch, where it improves the quality of model training by leveraging unlabeled data effectively.

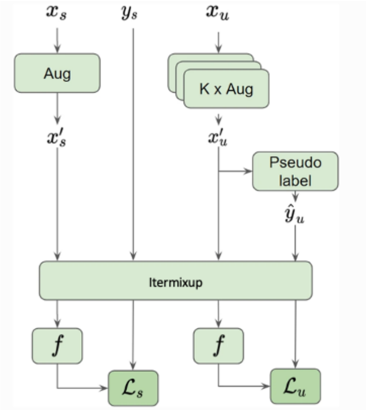


Figure 4.1: Mixmatch Workflow

Data Augmentation: Both the labeled and unlabeled data are augmented using strong data augmentation techniques. Augmentation helps in creating diverse samples, making the model more robust.
Mixup: Mixup is applied to the labeled data, which involves mixing two examples (input, target) together with a random weight to create a new augmented example. This encourages the model to make predictions that are linearly interpolated between the target labels.
Consistency Regularization: The augmented unlabeled data is used to generate pseudo-labels for the data points. The model's predictions on the unlabeled data are sharpened (temperature scaling) to make them more confident. This ensures that pseudo-labels are consistent and less noisy.
Loss Function: The loss function combines the supervised cross-entropy loss for labeled data and the unsupervised consistency loss for unlabeled data. The model is optimized to minimize this combined loss. The loss function for labeled data (L_s) is calculated as

$$L_s = \frac{1}{B_S} \sum_{(x_s'^{mix}, y_s^{mix})} CE(f(x_s'^{mix}), y_s^{mix})$$

and the loss for the unlabeled data can be calculated as:

$$L_u = \frac{1}{K.B_u} \sum_{(x_u'^{mix}, \hat{y}_u^{mix})} CE(f(x_u'^{mix}), \hat{y}_u^{mix})$$

Training: The model is trained using both labeled and pseudo-labeled data. The process of creating pseudo-labels and updating the model is repeated in an iterative manner.

4.3 Mixup

The main idea behind Mixup[25] is to augment the training data by creating new virtual samples that are a linear combination of pairs of original samples and their corresponding labels. This helps to regularize the model and improve its generalization ability. The method is particularly effective when dealing with limited labeled data. For each training iteration (or batch), randomly select two samples and their labels from the dataset. Create a new virtual sample and label as a linear combination of the two selected samples and their labels. The combination is defined as follows: Let x_1 be the feature vector of the first sample. Let x_2 be the feature vector of the second sample. Let λ be a random value drawn from a Beta distribution (usually with parameters α, α). The mixed input x_mix is computed as:

$$x_{mix} = \lambda * x_1 + (1 - \lambda) * x_2$$

The mixed label y_{mix} is computed similarly for categorical labels. Feed the mixed input x_mix through the model and obtain the model's predicted output y_pred_mix. Calculate the loss between the predicted output y_pred_mix and the mixed label y_mix. This loss is used to update the model's parameters through backpropagation and gradient descent.The idea of combining samples and labels through linear interpolation (controlled by the parameter λ) encourages the model to learn more smoothly in the input space, making it robust to variations and improving its generalization performance. The method effectively regularizes the model, reducing overfitting and making it less prone to memorizing the training data.

4.4 Itermixup

MixMatch, the current state-of-the-art method for data augmentation in audio classification, heavily relies on the mixup technique, particularly the parameter λ. The λ value, which lies between 0 and 1, plays a pivotal role in determining the mixing ratio of two distinct samples during augmentation. Specifically, when λ is closer to 1, the augmented sample predominantly reflects the characteristics of the first sample. Conversely, a λ value nearing 0 means the augmented sample will lean more towards the second sample's attributes.

In the MixMatch method, a λ value of 0.75 is conventionally employed. However, this approach might not harness the full potential of mixup, especially when considering the vast spectrum of possible λ values and their respective mixing ratios. To address this and further improve on the results obtained by the baseline methods, we have experimented with three distinct values of λ - [0.25, 0.5, 0.75]. By iteratively employing mixup with this diverse range of λ values, our proposed IterMixup function aims to generate a broader set of augmented samples, potentially capturing a more comprehensive representation of the data's inherent variations. This enriched set of samples can offer the model a more robust learning experience, potentially leading to enhanced performance.

Furthermore, the MixMatch method incorporates specific data augmentation techniques for weak data augmentation, namely Occlusion, Speed Perturbation, and Cutout. These methods have been meticulously chosen based on their widespread acclaim and proven effectiveness in numerous research papers. Occlusion introduces parts of silence or masking certain audio segments, providing a challenge for the model to predict based on incomplete data. Speed Perturbation slightly alters the playback speed of the audio samples, ensuring the model is robust to natural variations in speech or sound pace. Lastly, Cutout involves removing random sections of the spectrogram, compelling the model to make predictions even when parts of the data are missing. Collectively, these techniques not only enhance the diversity of the training data but also bolster the model's resilience to various real-world scenarios and imperfections in audio data.

By amalgamating the benefits of IterMixup with these tried-and-tested augmentation techniques, we aspire to push the boundaries of what's achievable in audio classification, striving for even greater accuracy and generalization capabilities.

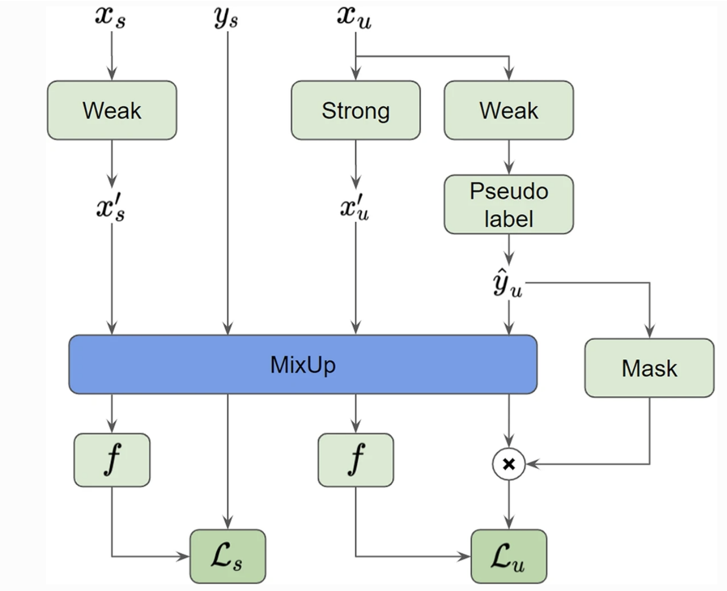


Figure 4.2: Itermixup integrating with Mixmatch

You can see how IterMixup is integrated with Mixmatch in the Figure ??.

Chapter 5

Dataset

The UrbanSound8K[4] dataset consists of 8732 samples (both mono and stereo) belonging to 10 classes: "air conditioner", "car horn", "children playing", "dog bark", "drilling", "engine idling", "gun shot", "jackhammer", "siren", and "street music". The classes are not balanced in terms of overall recording lengths per class. Each track has variable length up to 4 seconds, the native sample rate varies from 16kHz to 48kHz. The dataset was divided into 10 folds by its authors that we used in current work to perform our evaluation. This dataset has been widely used in research and also serves as a benchmark for evaluating the performance of any audio classification model. It has been used in various studies related to machine learning, deep learning feature extraction, audio signal processing and urban sound analysis. The dataset has complete annotations for each audio excerpt. The annotations include the class label, event start and end time inside the sample. These annotation help with the evaluation and bench marking of the audio classification models. This dataset uses real world audio recording rather than lab generated datasets like ESC-50 [20] which have more classes but are not useful in the real world applications. The urban environments from which the recordings are sourced can introduce more diverse and complex acoustic characteristics, potentially making the dataset more challenging. Key features of the UrbanSound8K[4] dataset:

1. Number of Recordings: UrbanSound8K contains 8,732 audio clips, providing a larger dataset size for training and evaluation.

2. Audio Clips: The dataset consists of 10 different classes of urban sounds, including air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren, and street music.

3. Duration: Each audio clip is 4 seconds long and is sampled at a rate of 44.1 kHz.

4. Audio Format: The audio clips are provided in WAV format, ensuring lossless and high-fidelity representation of the sounds.

5. Real-world Recordings: The audio clips were recorded in diverse urban environments, making the dataset representative of real-world urban soundscapes.

6. Class Imbalance: The dataset exhibits class imbalance, with some classes having more instances than others. This characteristic poses a challenge for sound event classification algorithms.

The majority of papers use accuracy, error rates and f-score as metrics while using this dataset for the task of Environmental sound classification.

Chapter 6

Experiments and Results

In this section we will discuss how the experiments were run, the hyperparamaters and the results obtained.

6.1 Experimental Setup

We ran the experiment for the UrbanSound8K dataset using the WideResNet-28-2 model. The experiment is a multiclass classification task for 10 classes of urban sounds. We ran multiple experiments using 3 different models namely WideResNet-28-2, MobileNetv1, Vgg. The experiments were conducted for the classification task for 10 classes on the UrbanSound8K dataset. For the baseline we ran the code available from GitHub for mixmatch-mixup and mixmatch+mixup with the 3 models. To improve on the results obtained by the baseline methods, we have used 3 values of λ which are $[0.25, 0.5, 0.75]$. This will generate three sets of augmented examples with varying degrees of interpolation between the original data points. To implement this we run the mixup method 6 times instead of 2 since mixup runs for both x and y. This creates 3 sets of augmented samples which all have different levels of regularization. We chose the different values of lambda as $0.25, 0.5 and 0.75$ it gives both the the samples equal representation in the augmented samples. With 0.25 the second sample gets more representation and with 0.75 the first sample gets more representation. When its 0.5, both the the samples get equal representation in the augmented sample. This helps us create more augmented samples which makes the model more robust and reliable as it introduces diverse perturbations to the training data and encourages the model to learn from different perspectives. The augmentation methods used for the weak and strong augmentations are Occlusion, Speed perturbation and CutOut as explained in 3.3. We have also added some more audio augmentation methods to our best result with IterMixup to check if adding Pitch Shifting and Time Stretching helps us get even better results. All the experiments were run on the Alice servers which are provided by the Leiden university. The server uses Intel Xeon Gold 6126 2.6GHz with 12 cores and PNY GeForce RTX 2080TI with 11GB memory This server helps us run the experiments using multithreading which significantly decreases the training time from about 30 minutes per epoch to 4-5 minutes.

6.2 Results

The experiment in 6.1 gave the accuracy of our model as about 89%. It is a significant improvement over the previous method as can be seen in the table below.

Model	Method	Error Rate	Accuracy	F-score
WideResNet	Mixmatch(no mixup)	20.42	79.58	0.84
	Mixmatch+ mixup	18.02	81.98	0.88
	Mixmatch + Itermixup	10.34	89.66	0.90
	CNN (WideResNet-28-2)	15	85	0.85
MobileNetV1	Mixmatch(no mixup)	19.36	80.64	0.83
	Mixmatch+ mixup	15.47	84.53	0.84
	Mixmatch + Itermixup	13.4	86.6	0.88
	CNN (MobileNetV1)	16.23	83.77	0.86
VGG	Mixmatch(no mixup)	20.22	79.78	0.85
	Mixmatch+ mixup	18.59	81.41	0.88
	Mixmatch + Itermixup	11.4	88.6	0.89
	CNN (Vgg)	15.39	84.61	0.83

Table 6.1: Results for the Mixmatch with no mixup, Mixmatch, Mixmtach with IterMixup data augmentation methods and only CNN on 3 different Convolutional Neural Networks for the Audio Classification task

The results in Table 6.1can more easily be analyzed using the given graph in Graph 6.1.

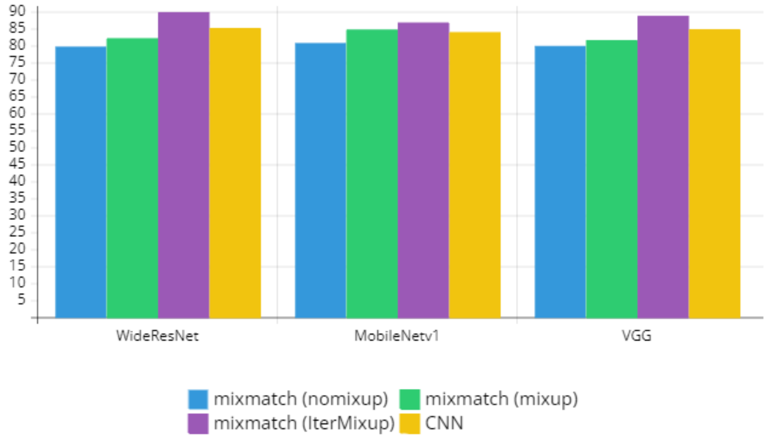
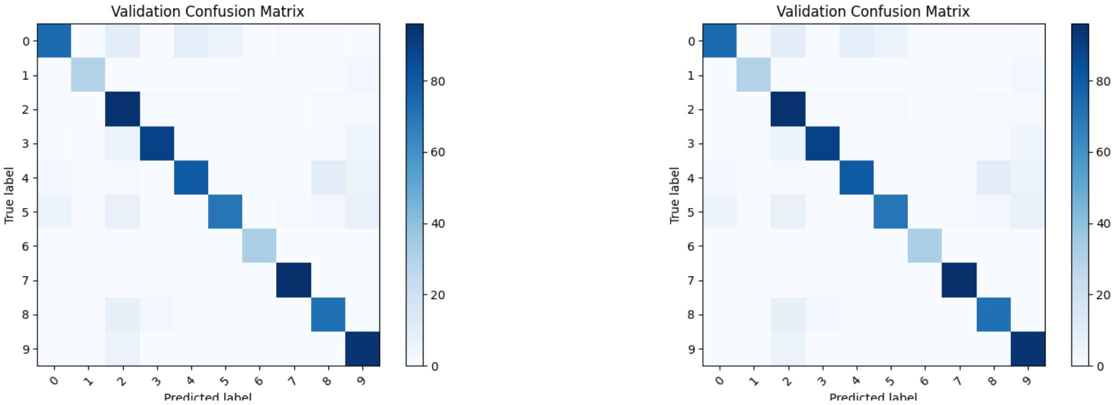


Figure 6.1: Comparing results between the implemented models and the models used

In the table 6.2, there are the results for the model WideResNet, implementing different values of λ for the Mixup and Itermixup method. This helps us understand the effects of the IterMixup today and how the amount of λ values affects the training and learning process. We can see where the differences in the results came from the following confusion matrices:

λ values	Error Rate	Accuracy	F-score
[0.25]	15.27	84.73	0.78
[0.25,0.5]	17.69	82.31	0.81
[0.25,0.5,0.75]	10.34	89.66	0.90
[0.2,0.4,0.6,0.8]	10.49	89,51	0.81

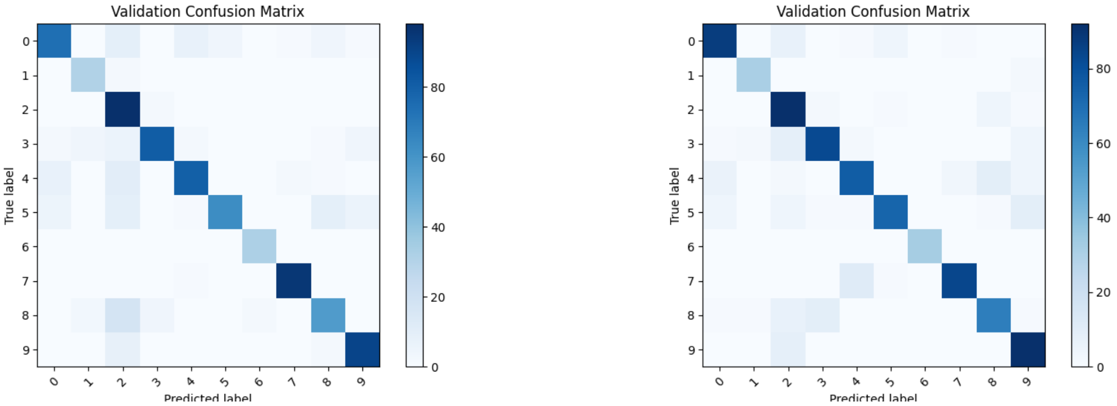
Table 6.2: Comparison between different values of λ for the WideResNet model



(a) Mixmatch

(b) Mixmatch with IterMixup

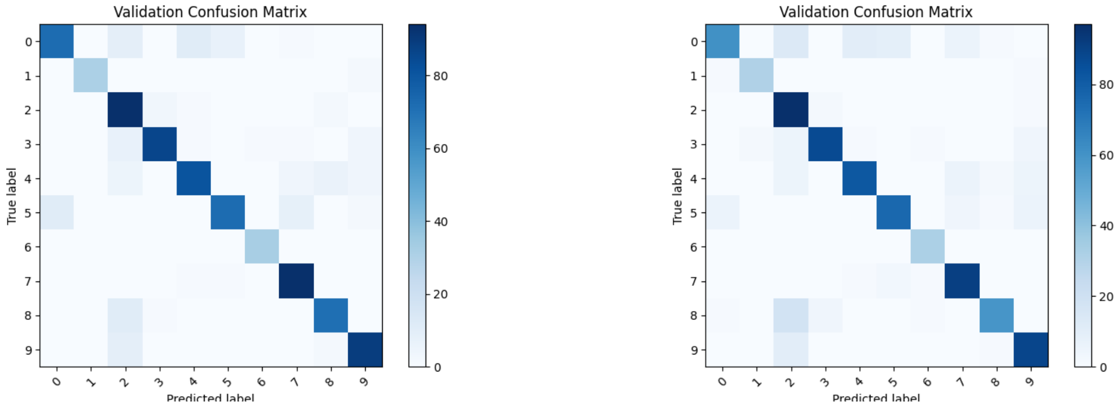
Figure 6.2: Model: WideResNet



(a) Mixmatch

(b) Mixmatch with IterMixup

Figure 6.3: Model: MobileNetv1



(a) Mixmatch (b) Mixmatch with IterMixup

Figure 6.4: Model: VGG

The above matrices demonstrate an improvement in the classes successfully predicted in each of these models following the application of IterMixup. The matrices illustrate where the model has improved by learning from more data produced by the IterMixup method in addition to the MixMatch approach.

Mixmatch + IterMixup	Error Rate	Accuracy	F-score
Occlusion, Speed Perturbation and CutOut	10.34	84.73	0.78
Occlusion, Speed Perturbation, CutOut, Pitch Shifting,Time Stretching	17.69	82.31	0.81

Table 6.3: Comparison of Mixmatch + IterMixup with 3 augemntation methods and 5 augmentation methods

In the Table 6.3 we can see there is only a slight increase in the accuracy even by adding Pitch Shifting and Time Stretching in the mix.

Chapter 7

Conclusion and Discussion

For answering the first question "What are the current state of the art methods for data augmentation for the audio classification task?" We can see in the paper [6] that the state of the art method for this dataset is The Mixmatch method with Fixmatch + mixup coming in very close. MM uses pseudo-labeling, with explicit entropy minimization, sharpening in case of Mixmatch. In some other methods, no entropy minimization is used, the predictions on the unlabeled part of the data are used as is for a consistency criterion between the two collaborating networks. This method achieves about 82% accuracy for this classification task. Mixmatch uses Mixup and consistency regularization which are two powerful techniques that can improve model generalization. Mixup encourages the model to learn more robust features by creating augmented examples that are combinations of multiple training samples. Consistency regularization enforces the model's predictions to be consistent when applied to perturbed versions of the same input. MixMatch combines these two techniques, leveraging their complementary benefits to enhance the model's performance. This method is particularly designed for semi-supervised learning, where the labeled data is limited, and the model needs to utilize both labeled and unlabeled data for training. By using unlabeled data with consistency regularization, MixMatch enables the model to leverage large amounts of unlabeled data to improve performance. In datasets with class imbalance like the dataset we have used, traditional data augmentation methods may not address the issue adequately. MixMatch can help mitigate class imbalance by generating augmented examples for underrepresented classes, promoting better class representation and reducing bias.

For answering our second question "Is there a way to increase the performance of the current methods using a new data augmentation technique?". We tried to improve the current state-of-the-art method which is Mixmatch by adding multiple iterations of mixup each of which are using different values of lambda (λ) [0.25, 0.50, 0.75]. Using different lambda values for Mixup results in a more diverse set of augmented examples. Each lambda value determines the mixing ratio between two data points, and different ratios introduce distinct types of perturbations. This diversity in augmentation helps the model to learn from a broader range of data variations, leading to improved generalization. Running Mixup with multiple lambda values enables the model to explore various regions of the data space more thoroughly. This exploration can help the model identify challenging data instances and learn more meaningful representations, leading to improved performance on complex or ambiguous samples. Running Mixup with different lambda values allows for hyperparameter tuning of lambda. By experimenting with a range of lambda values, you can identify the most suitable lambda settings for

your specific dataset and task. Different lambda values offer different levels of regularization during training. Larger lambda values result in stronger regularization, which can be beneficial for reducing overfitting. This improved method gives us the accuracy of about 90% which is about a 7-8% increase from the original MixMatch method. So, to answer our question, Yes, we can improve the original Mixmatch method by running the mixup iteratively.
To further improve upon MixMatch, we incorporated multiple iterations of mixup using varied lambda λ values [0.25, 0.50, 0.75]. Different λ values introduce distinct perturbations, enhancing model generalization. This iterative mixup approach achieved about 90% accuracy on the UrbanSound8K dataset, an 8% improvement over the original MixMatch. We tried adding additional audio augmentations namely Time Stretching and Pitch Shifting. This helps the model learn not a lot but makes the model feel more robust. UrbanSound8K, a widely-used dataset for sound classification, comprises 10 classes from the urban sound taxonomy. Our experiments utilized the updated mixup alongside the WideResNet-28-2 architecture. This study both reviews and extends current audio classification data augmentation techniques. Potential future research directions include exploring reinforcement or unsupervised learning and fine-tuning mixup's alpha values for task-specific optimization.

7.1 Future Work

In light of the advancements made in audio classification using the MixMatch approach, several avenues present themselves for further exploration. The landscape of learning paradigms, including Reinforcement Learning and Unsupervised Learning, remains largely untapped for audio tasks and could yield insightful comparisons with current semi-supervised methodologies. While the iterative mixup approach has shown promise, a comprehensive hyperparameter optimization using methods like Bayesian Optimization might fine-tune the mixup's λ values and other model intricacies. The vast realm of deep learning offers alternative architectures, such as Transformer-based models or Capsule Networks, which, when paired with ensemble techniques, could elevate performance metrics. Leveraging pre-trained models from expansive audio datasets, akin to the BERT model in NLP, and fine-tuning them for specific tasks like those in UrbanSound8K might usher in significant improvements. The concept of audio embeddings, borrowing inspiration from word embeddings in NLP, could encapsulate richer audio information, enhancing classification efficacy. A multi-modal learning approach, integrating audio with visual or textual data, can be a game-changer, especially in multifaceted environments. Validating models in real-world, noisy scenarios, possibly via deployment on IoT devices, remains crucial for assessing practical applicability. The inclusion of attention mechanisms might refine the model's focus on pivotal audio segments, and sourcing additional labeled and unlabeled data can bolster the semi-supervised paradigm, potentially pivoting towards weakly labeled datasets. As we navigate these potential enhancements, the overarching goal remains to develop audio classification models that are not only academically robust but also pragmatically effective.

Bibliography

[1] Asith Abeysinghe, Sitthichart Tohmuang, John Laurence Davy, and Mohammad Fard. Data augmentation on convolutional neural networks to classify mechanical noise. *Applied Acoustics*, 203:109209, 2023.

[2] Vanita Babanne, Nikita Sandeep Mahajan, Renu Lalchand Sharma, and Pratiksha Pradip Gargate. Machine learning based smart surveillance system. In *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 84–86, 2019.

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Avital Oliver, Nicolas Papernot, and Colin Raffel. *MixMatch: A Holistic Approach to Semi-Supervised Learning*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[4] Dmitry Bogdanov, N Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, Oscar Mayor, G Roma, Justin Salamon, Jose Zapata, and Xavier Serra. Essentia: an audio analysis library for music information retrieval. 11 2013.

[5] Adam Bohr and Kaveh Memarzadeh. The rise of artificial intelligence in healthcare applications. In *Artificial Intelligence in healthcare*, pages 25–60. Elsevier, 2020.

[6] Léo Cances, Etienne Labbé, and Thomas Pellegrini. Comparison of semi-supervised deep learning algorithms for audio classification. *EURASIP Journal on Audio, Speech, and Music Processing*, 2022(1):23, September 2022.

[7] Terrance DeVries and Graham Taylor. Improved regularization of convolutional neural networks with cutout. 08 2017.

[8] Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass. Ssast: Self-supervised audio spectrogram transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10699–10709, Jun. 2022.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society.

[10] Andrew Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. 04 2017.

[11] Rongjie Huang, Jia-Bin Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiaoyue Yin, and Zhou Zhao. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. *ArXiv*, abs/2301.12661, 2023.

[12] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. Audio augmentation for speech recognition. In *Sixteenth annual conference of the international speech communication association*, 2015.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[14] Teerath Kumar, Rob Brennan, and Malika Bendechache. Stride random erasing augmentation.

[15] Teerath Kumar, Jinbae Park, and Sung-Ho Bae. Intra-class random erasing (icre) augmentation for audio classification. In *Proceedings Of The Korean Society Of Broadcast Engineers Conference*, pages 244–247. The Korean Institute of Broadcast and Media Engineers, 2020.

[16] Zohaib Mushtaq and Shun-Feng Su. Environmental sound classification using a regularized deep convolutional neural network with data augmentation. *Applied Acoustics*, 167:107389, 2020.

[17] Loris Nanni, Gianluca Maguolo, and Michelangelo Paci. Data augmentation approaches for improving animal audio classification. *Ecological Informatics*, 57:101084, 2020.

[18] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Dogus Cubuk, and Quoc V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. In *Interspeech*, 2019.

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

[20] Karol J. Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, page 1015–1018, New York, NY, USA, 2015. Association for Computing Machinery.

[21] Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, mar 2017.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[23] Salini Suresh, Suneetha V, Niharika Sinha, and Sabyasachi Prusty. Latent approach in entertainment industry using machine learning. *International Research Journal on Advanced Science Hub*, 02(Special Issue ICARD 2020):304–307, 2020.

[24] Yidong Wang, Hao Chen, Yue Fan, Wangbin Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, Heli Qi, Zhen Wu, Yu-Feng Li, Satoshi Nakamura, Weirong Ye, Marios Savvides, Bhiksha Raj, Takahiro Shinozaki, Bernt Schiele, Jindong Wang, Xingxu Xie, and Yue Zhang. Usb: A unified semi-supervised learning benchmark for classification. In *Neural Information Processing Systems*, 2022.

[25] Hongyi Zhang, Moustapha Cissé, Yann Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *ICLR, 2017*, abs/1710.09412, 2017.

[26] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):13001–13008, Apr. 2020.