Evaluating performance of shuffling data augmentation techniques for audio event detection.

MSc Research Project
MSc in Computing

David Kelly
Student ID: 13127390

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	David Kelly		
Student ID:	13127390		
Programme:	MSc Cloud Computing	**Year:**	2022
Module:	MSc Research Project		
Supervisor:	Vikas Sahni		
Submission Due Date:	10/11/2022		
Project Title:	Evaluating performance of mixing and shuffling data augmentation techniques for audio scene classification.		
Word Count:	4940		
Page Count	18		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: David Kelly……………………………………………

Date: 10/11/2022..……………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Evaluating performance of mixing and shuffling data augmentation techniques in audio event detection domains.

David Kelly
13127390

Abstract.
Audio Event Detection is an emergent field of machine learning. The goal of is to the world around us through sound. Animal habitat preservation, ambient assisted living and preventative machine maintenance are all fields exploring the commercial use of AED to augment human decision machine. A key challenge stalling the development of AED systems is the lack of high-quality audio data labelled audio data. Producing such data is an expensive and time-consuming task. Techniques which offer a classification performance uplift in data constrained domains are of particular interest to AED research. This technical report describes the design and replication of the best performant submission from DCASE 2018 Task 5. The research critically evaluates the technique proposed by Inoue. Through experimentation, a set of alternatives of experimental parameters were discovered which exceed the performance of the reference implementation on the same challenge dataset. This research shows promise for the field of audio event detection where the use of mitigation to deal with a lack of high-quality labelled data for a given classification scenario is common. This work also shows an improvement in performance over the baseline work when comparing GPU compute effort required to reach equivalent performance classification performance.

1 Introduction

1.1 Audio Event Detection and Audio Scene Classification

Audio event detection (AED) and audio scene classification (ACS) are emerging fields within the machine learning domain. Systems within these fields aim to understand the world through ambient sound. Commercial applications of this technology are available via ambient assisted living, ecological animal habitat evaluation, preventative machine maintenance, and smart home automation. For an ageing population in a digital world, ambient assisted living systems promote independence and augment the support provided by caregivers. AED systems may have applications in ambient fall detection. An ACS such as [1] may provide care givers tools to quantify a client's independence and daily routines to tailor specific lifestyle interventions. Predictive maintenance is an embryonic area of machine learning which has already seen commercialisation. RDI Technologies use machine vision and motion amplification to visualise machine health to a human reviewer. In 2019, Henze et al [2] proposed a novel system for predictive maintenance in industrial settings using audio classifiers for anomaly detection. IBM in 2020 pursued a similar goal [3] of detecting anomalous sounds in order to infer machine health. AED and ACS borrow techniques from the voice recognition and image recognition machine learning space. In image recognition, image pixel grids become reduced to arrays for evaluation through loss functions. Through successive training attempts and providing further input data, a model can learn overtime to detect certain elements from an image and link these features with known data labels[4] This has been highlighted with voice recognition as when audio is represented as a spectrogram, similar machine learning techniques from image recognition can be used. These techniques typically require domain specific tailoring to accommodate the time-dimension nature of audio as it is conveyed within a spectrogram.

1.2 Aim and Objective

The research question investigated by this work is the critical evaluation of the performance of novel data augmentation techniques from Inoue proposed in DCASE 2018.

The aim is to implement a proposed data augmentation technique which is tailored to audio event detection. Through implementation, this research will critically evaluate performance of the technique and discover how accuracy responds to alternative experimental parameters similar to what was initially specified. This research used a neural network architecture as a testbed to collect learning metrics throughout the training process. The network was repeatedly reset and retrained with augmented data generated by a parameterised implementation of IBM’s algorithm.

The objective of this research is to understand if the original augmentation parameters as specified from IBM were the best possible set of parameters or if their performance can be exceeded through the discovery of alternative parameters. The architecture of this project was inspired by the original experiment from 2018 alongside the baseline system as provided by the challenge organisers. The algorithm for the chosen augmentation technique was engineered based on the functional description as provided in the original work.

1.3 Motivation

Data augmentation techniques provide a novel way of improving classifier accuracy. A key challenge in audio recognition is gathering a large and diverse corpus of labelled data. Dataset curation for audio scenes may create distinct challenges. Collecting audio requires specific environmental conditions and human intervention to ensure its capture and labelling. Dekkers [5] described the collection of the DCASE dataset as requiring a human volunteer to live in a lab setting for a period of 1 week. The significant investment in human effort was only able to generate around 70,000 training samples. Data Augmentation and techniques that can provide more useful training samples from a labelled corpus therefore are of key importance to audio scene classification. The 2018 submission stood out among the challenge entrants as it aimed to achieve accuracy driven primarily through augmentation. Therefore, this research mitigated against one of the key challenges in audio classification. For this research, the original authors of the 2018 paper were contacted to seek clarification questions around performance. One question queried a parameter setting which the authors stated their choice was driven by a length that appeared to be contextually meaningful, however they also commented they did not explore the relationship between performance and the variable in question further. This paper continues to explore this pathway and attempts to indicate areas for further enhancements.

2 Related Work

2.1 Audio Representation

Digital audio is a representation of sound in a digital domain. Sound waves and the variance in sound pressure is captured through microphones. It is then converted into samples and encoded for future playback. The quality of frequency information in the sample conversion is determined by the "sample rate" or Shannon-Nyquist frequency of the encoder[6], with pulse code modulation and bit depth determining the quality of amplitude information. For example, audio from a normal CD contains audio represented in samples captured at a rate of 44.1 Khz at 16 bits of depth per channel, typically a left and right channel. A single second of CD quality audio can therefore be represented in 1.4Kb of information.

2.2 Raw Audio Feature Classification

The raw data rate for audio is seen as a challenge for machine learning tasks. Model architects are typically forced to produce shallow networks which are unable to detect high level features in each sample. Dai et al proposed a very deep network technique in 2016 which processed raw audio from the Urban8K dataset [7]. Their results were state-of-art in relation to raw audio classification as their accuracy scores were in the range of 62% to 67% which was unseen at that point in time. Work from Google in 2015 also supported the idea of using raw audio [8]. Their work looked at word detection and found an equivalent performance between raw audio representation and lower dimension audio representation.

2.3 Spectrum Feature Classification

Though classification through raw audio is possible, the best performing systems typically feature a lower-level representation of audio in the form of a spectrogram. A spectrogram is an image-like representation of audio with time information preserved on the X axis, frequency information encoded on the Y axis and intensity being indicated through pixel intensity. In 2019, In 2019, Su et al developed a multi-level network trained on lower dimension representations of audio but despite this achieved accuracies on the Urban8K dataset in the range of 97%[9]. In 2017, Wyse proposed training the network using an image-like representation of sound in the form of a spectrogram[10]. This work further builds on the approaches from 2015[11].

2.4 Data Augmentation

Data augmentation is an approach which aims to improve the generalisation capabilities of a neural network by providing a series of generated or distorted samples during the training phase. The survey work of Shorten and Khoshgoftaar in 2019 enumerated the myriad of ways of performing augmentations for image classification tasks[12] which included approaches such as rotating, scaling, zooming, shearing, and cropping. They stated that data augmentation for images as a series of filters or transformations on a source image to produce a new, humanly recognisable sample belonging to the same original class but with modified characteristics. These characteristics included brightness, noise level, orientation, or colour adjustments. Bello and Salamon [13] in 2016 suggested a collection of audio specific augmentations for audio and reported that on average, there was a 5% increase over baseline performance when all techniques were combined. Their techniques originated from audio signals processing and include time-stretching (while maintaining pitch), pitch shifting (while maintaining sample duration), dynamic range compression and adding background noise to samples.

Wang and Perez ultimately summarised the role of data augmentation[14] as:
"It is common knowledge that the more data an ML algorithm has access to, the more effective it can be. Even when the data is of lower quality, algorithms can perform better, as long as useful data can be extracted by the model from the original data set"

2.5 Data Augmentation through Shuffling

A team of researchers from IBM led by Tadanobu Inoue [15] placed first in DCASE 2018 Task 5 through the implementation of two novel data augmentation techniques. These augmentation techniques were combined and achieved an 88.4% and 90% F1 score on evaluation and development datasets [5] respectively.

2.5.1 Shuffling

Inoue used shuffling techniques to take a file and slice it into n-many segments along its time axis. The segments were then shuffled, recombined, and added to the training corpus.

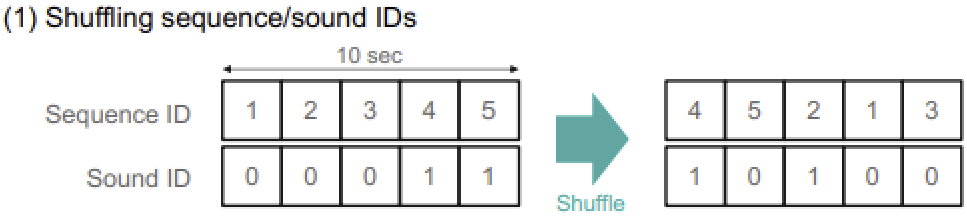

Figure 1 – Descripting of Shuffling and Mixing from original IBM work [15]

2.5.2 Background to approach

Inoue cited two key works as the origin to the team's novel technique. Takahashi et al proposed a technique to increase data variance and posits that by mixing two sounds from the same class, for example bird song or ocean surf, then the result must be of the same class [16]. In Takahashi's EMDA, two samples were mixed equally. They were presented as being played one on top of another, or could be "heard at the same time" to a listener. In 2017, Zhang and Facebook researched a data agnostic way of generating new training sample data from vectors. At the forefront of Zhang's Mixup [17] techniques was the generation of new samples where the training data and labels were manipulated through a random linear interpolation. Zhang's work was not specific to images or audio and they proposed that the data technique is data agnostic.

Finally, the work of Tokozume proposed a data augmentation technique specifically for audio convolutional neural networks[18]. Their technique mixed files from between classes to generate a new mixed sample for addition to the original training dataset. Inoue's technique aimed to mitigate a fundamental issue with the foundational works. Each of the mentioned techniques increased sound density within a generated sample. This may result in an audio sample that is not understandable by humans and results in overfitting the network. Therefore the network is encouraged to make predictions on samples with simply a higher event density.

2.6 Improvements on DCASE 2018 Task 5

The Detection and Classification of Acoustic Scenes and Events is an open challenge originally established within Queen Mary University in 2013. It is now affiliated with signal processing conventions such as ICASSP and EUSIPCO. The 2018 Task 5 challenge created an objective to quantify human activity in a domestic setting . Labelled audio data was captured over one week and contained the activities of a person on vacation via a series of microphone arrays. Since DCASE 2018, several independent researchers have continued to tackle the Task 5 challenge. Zhang in 2020 [19] compared their work in transformer-based encoders against the top 3 classifiers from the DCASE 2018 Task 5. Zhang's work achieved an accuracy in training at 91 and 87.5 in evaluation; remarkably close to the peak performance of Inoue's 90 and 88.4 scores respectively. Also in 2020, Amiriparian et al evaluated [20] what performance can be established by applying a pre-trained network from outside the world of Audio Scene Classification to audio domain problems. They used popular models such as ImageNet and ResNet. Their analysis yielded interesting results. When analysing the DCASE 2018 Task 5 data set, Inoue's system dominated with the highest performance. A DenseNet121 + ImageNet system scored 81.1% on the evaluation dataset. Finally, the current state-of-the-art is an implementation that comes from 2019 [19] where researchers used a mixture of pretrained models and known architectures (E.G AlexNet [4]) as inputs to support vector machines. This approach was applied to the Task 5 corpus and achieved an accuracy of F1 Score of 97.46 and did not use any data augmentation.

2.7 Implementations of Shuffling and Mixing

Technology Company Xiaomi in 2020 proposed a neural searching architecture [21]. For its implementation, they used the data augmentation techniques from Inoue's work and evaluated their accuracy on the DCASE 2018 Task 5 dataset. They deviated from Inoue with their network architecture and used MobileNetV2's network architecture instead. Their work also performed the data augmentation "offline", which suggested they are preprocessing the data and wrote the results to disk. Xiaomi's work achieved similar results to Inoue, with a classification accuracy of 90.3%. Xiaomi also highlighted that their work required 25% less FLOPS, suggesting that their approach may have suitability in CPU restrained mobile systems.

3 Research Methodology

The research approached improving classification performance on the SINS[5] dataset through architecture of an experiment to replicate and implement the shuffling technique from Inoue.

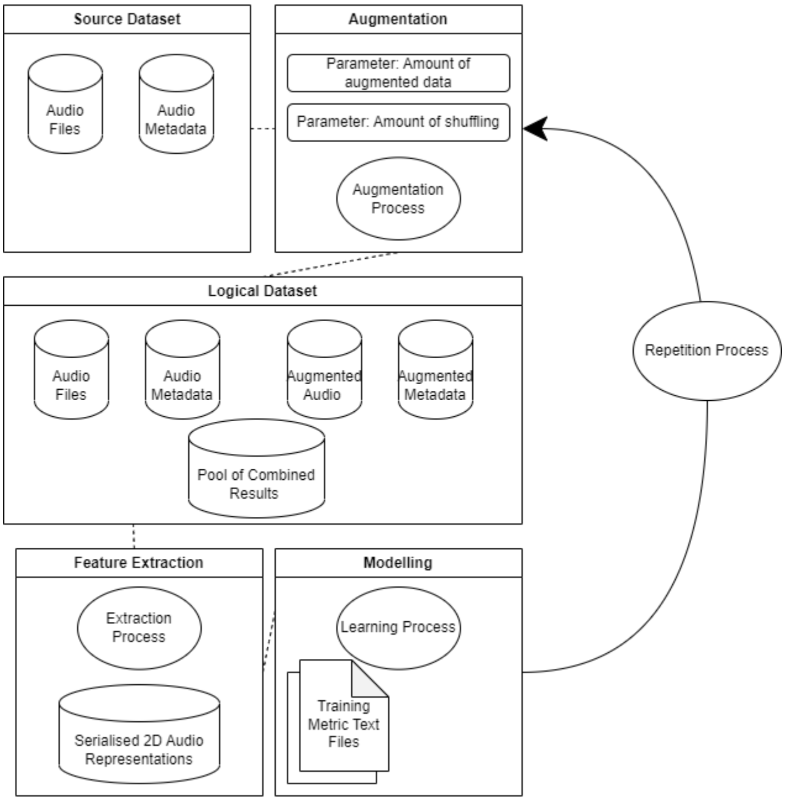

Figure 2: Flow of interaction and data between phases

The research methodology uses a given neural network architecture from Qiuqiang[21] as a testbed to evaluate performance of augmentation parameters and the effect of the augmentation parameters on event detection accuracy.

The neural network operated on the SINS Audio Event dataset of 72,000 file samples as was utilised in Inoue's work. An augmentation generator was created to create new audio samples based on existing samples from the dataset. The generator also made the appropriate amendments to the metadata files that are relied upon for feature extraction, training, and testing of the neural network. The experiment was then repeated for a given set of augmentation parameters and desired percentage of generated samples to be added to the dataset.

In Feature Extraction, audio files were read from disk and converted from an audio format into a serialised stereo log-mel format. The feature extraction steps took the metadata files as inputs and retrieved both filepath and event classes for each file. The feature extraction then programmatically extracted the log-mel features of the audio into a single HDF5 file.

In Data Augmentation, a subset of the SINS dataset was selected based on a given percentage. Each file from the subset is used to produce an augmented copy based on chosen the augmentation parameters. The class of the newly generated sample was then considered the same as the input sample. The class and the filename for the newly generated sample were appended to the relevant metadata files.

In Modelling, the neural network received the features and event classes from the HDF5 file as an input. The input data was typically divided into a train and holdout dataset at a ratio of 4:1. The model architecture that was defined by Qiuqiang was an implementation of the baseline DCASE 2018 proposed by Dekkers[22]. Compared to the baseline, Qiuqiang's model deviated by reducing the log-mel input size along with compressing the 4 channels of audio to 2 channels of audio.

The modelling and learning architecture was developed using the Pytorch machine learning framework. The performance of the network during training and testing was logged at key iteration milestones using SciKitLearn. This allowed for the observation of accuracy and loss function behaviour over time in response to the augmented data.

In Repetition, the steps from Data Augmentation to Modelling were repeated. First, metadata files were reset to a base state with no pointers to augmenting data. Next, the data augmentation step was provided with new input parameters. Finally, training continued as normal thus allowing the rest of the architecture to remain unchanged. This provided a fair and stable comparison between runs.

In Comparison, the performance metrics from previous runs were reviewed. Where a trend emerged from the results, subsequent experiment runs were carried out to monitor performance considering any new insights.

4 Design Specification

A key aspect of the experimental design is in the implementation of augmentation. The chosen augmentation algorithm was inspired by Inoue's "shuffling" approach. The implementation can be described as follows:

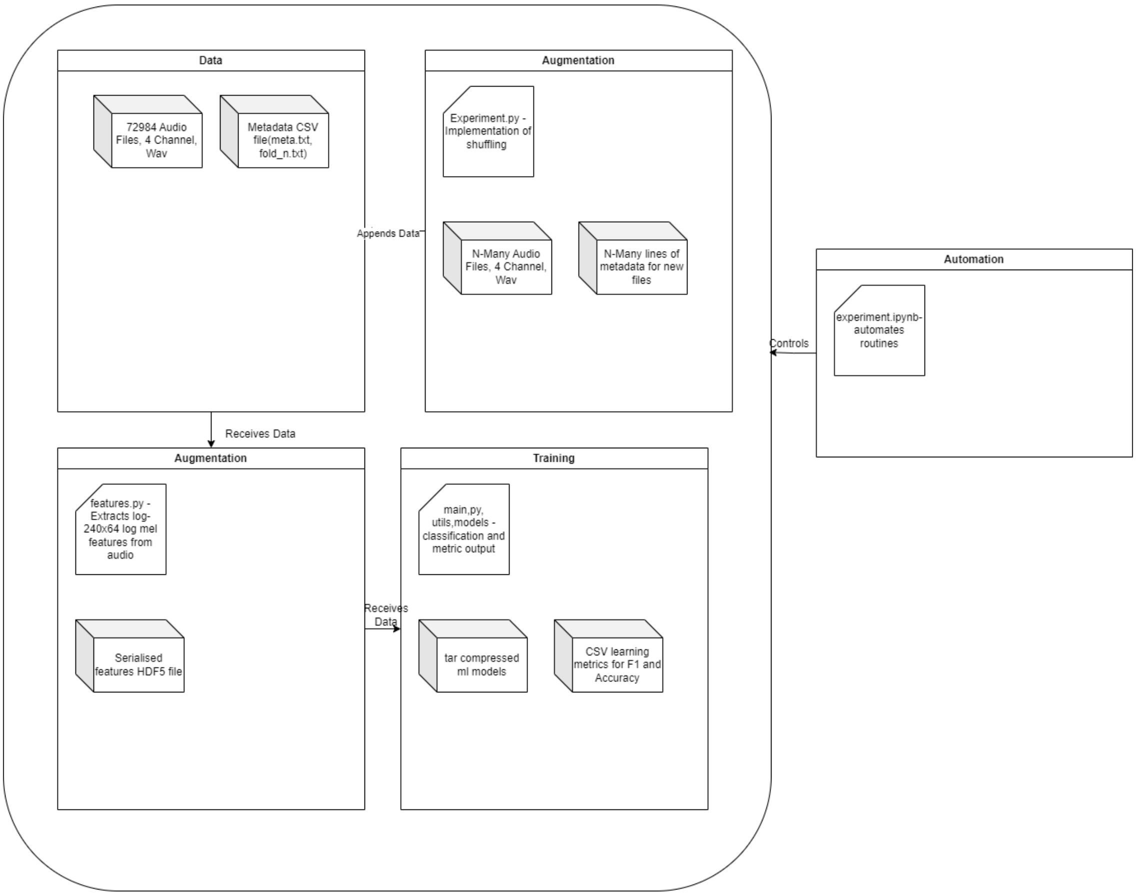
Data
72984 Audio Files, 4 Channel, Wav
Metadata CSV file(meta.txt, fold_n.txt)
Appends Data
Augmentation
Experiment.py - Implementation of shuffling
N-Many Audio Files, 4 Channel, Wav
N-Many lines of metadata for new files
Automation
experiment.ipynb- automates routines
Controls
Receives Data
Augmentation
features.py - Extracts log-240x64 log mel features from audio
Serialised features HDF5 file
Receives Data
Training
main.py, utils,models - classification and metric output
tar compressed ml models
CSV learning metrics for F1 and Accuracy

Figure 3 – Details of inputs, outputs and processes within system.

For a given audio sample, subdivide the file into n-many audio segments and store these segments. If the order of the stored segments can be considered the same order as the originating file, repeat the shuffling again. If the two files do not match, a new file is now considered to have been created. The new sample is written to disk. The meta data for the new sample is written to relevant metadata files for future use.

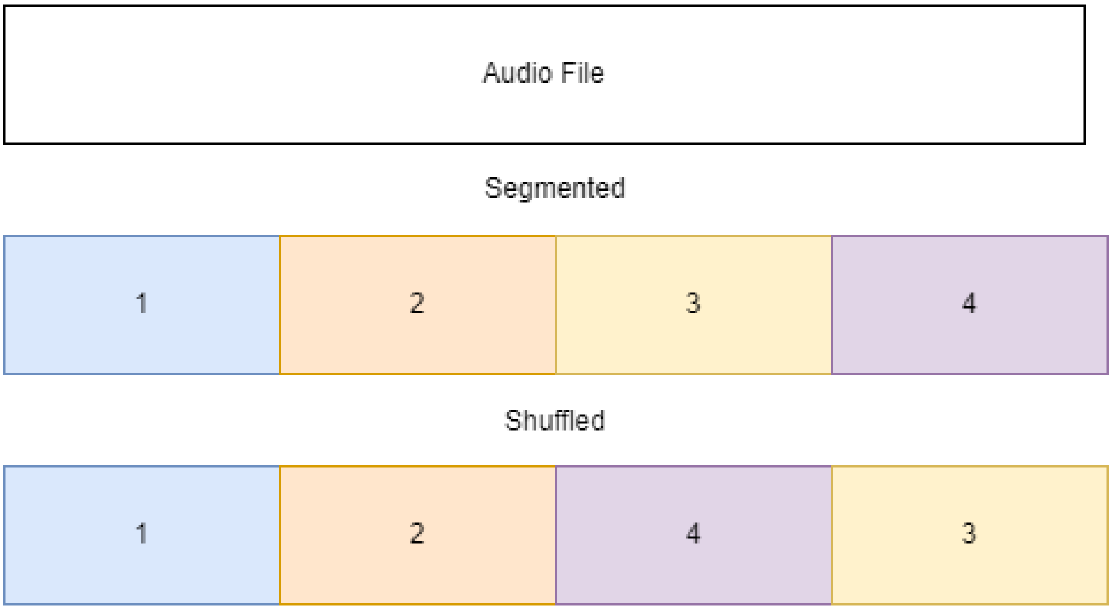

Figure 4: Illustration of audio file segmentation and shuffling

The next key aspect of the design is a mechanism to automate and parameterise the augmentation, feature extraction and training steps. A routine is defined through code to isolate augmentation from the feature extraction and training. Dividing the tasks in this way allows for each to be experimented with in isolation if required but also scripted to run a series of procedural tasks such as changing one or two parameters.

The final part of the design required to critically evaluate training performance in the system was the learning metric capture and analysis. The system will record F1 performance scores throughout all training activities. The metrics are stored in labelled CSV files for further analysis outside of the code programming environment.

F1 scores are one of the key performance metrics used to understand how well a machine learning system is performing. It is a statistical test of accuracy and can be defined as `2 x [(Precision x Recall) / (Precision + Recall)]` where recall is defined as capturing of positive cases and precision is defined as ability to capture the correct positive item.

5 Implementation

The experiment was implemented as a Juptyer notebook containing python and pytorch artefacts. The key artefact for augmentation is contained with the python modules experiment.py. The shuffle method provides the implementation of Inoue’s algorithm.

```
def shuffle(path):
    audio = AudioSegment.from_file(path)
    seg_len = len(audio) // cuts
    audio_container = []
    for i in range(0, cuts):
        if i == 0:
            audio_container.append(audio[0:seg_len])
        else:
            start = seg_len * i
            end = seg_len * (i+1)
            audio_container.append(audio[start:end])
    res = random.sample(audio_container,cuts)
    while(res == audio_container):
        res = random.sample(audio_container,cuts)
    return sum(res)
```

One tool of note in this implementation is the use of the library PyDub. This allows audio to be treated much like an array and there provides suitable tools to manipulate the audio along its time domain by manipulating the order of the audio array.

The remaining code from experiment.py provide clean up functions to reset and augment metadata files along providing timing estimates through the TQDM library.

Experiments were conducted through VS Code and Jupyter. Jupyter allowed for a scripting and automation-like approach to be taken in development. Hooks were placed in the codebase allowing for Jupyter cells to inject code or variables into the running routines. Jupyter allowed whole routines to be looped and procedurally run. Architecting for procedural and idempotent operation was a critical requirement of the experiment.

Though the development workstation is a PC for development tasks, the typically workstation run time to produce a single result was over 60 minutes for a 10% augmentation dataset and 125 minutes for a 50% augmented dataset. Generating the full set of experimental results takes 16 hours. As such, being about to loop, interrupt and trigger routines became a vital functional requirement of the experiment. The data from each experiment was captured during each run and during key milestones during training. Each network was trained up to a maximum of 10,000 iterations at a batch size of 70. That is to say, 700,000 files passed through the network by the end of a training run. This parameter was chosen based on GPU memory capacity and wall clock time. For comparison, Inoue trained their network 500 times on the full corpus of data, coming to a total maximum of 36,000,000 file operations[15].

The following augmentation parameters were chosen to explore performance beyond the ranges of Inoue's five slices implementation.

Slices	Percentage of Augmented Data	Dataset
2	10	Development Fold 1
3	10	Development Fold 1
5	10	Development Fold 1
10	10	Development Fold 1
20	10	Development Fold 1

100	10	Development Fold 1
2	50	Development Fold 1
3	50	Development Fold 1
5	50	Development Fold 1
10	50	Development Fold 1
3	50	Development Fold 1-4

Figure 5: Experiment Plan

To facilitate easy comparison between the experiment results, a PowerBI model is provided with preconfigured dashboards to analyse the experimental results.

6 Findings

6.1 10 Percent Augmentation

Slices	F1 Average Score after 2000 Training Iterations
Baseline (No Augmentation)	.86
2	.89
3	.90
5	.90
10	.89
20	.89
100	.88

The initial set of experiments which added 10% of augmented data demonstrated approached using 3 and 5 cuts were the best performers. Another insight of from the data is the 2 to 5% improvement over baseline with only 10% additional data. Finally, this data indicates performance tapering off with slice amounts of 10 and greater.

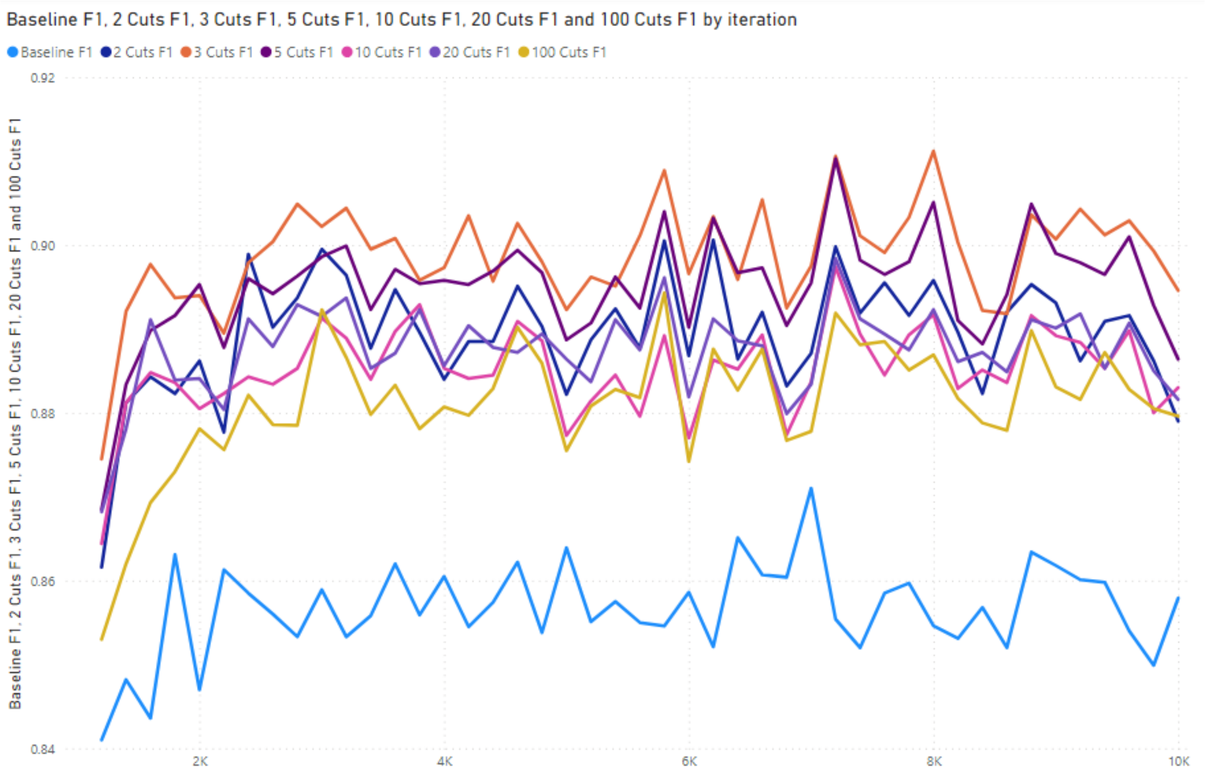

Figure 6: Training Performance over Iterations – 10% Augmented Data

6.2 50 Percent Augmentation

Slices	F1 Average Score after 2000 Training Iterations
2	.91
3	.95
5	.94
10	.93

In 50% augmentation scenarios we the previous trend continue with three slices becoming the standout leader at all points during training. The trend continues to be followed with ten slices seeing a similar drop in performance compared to five slice scenarios. A curious observation from this data is noted in two slice scenarios, as it improved the least in response to additional data.

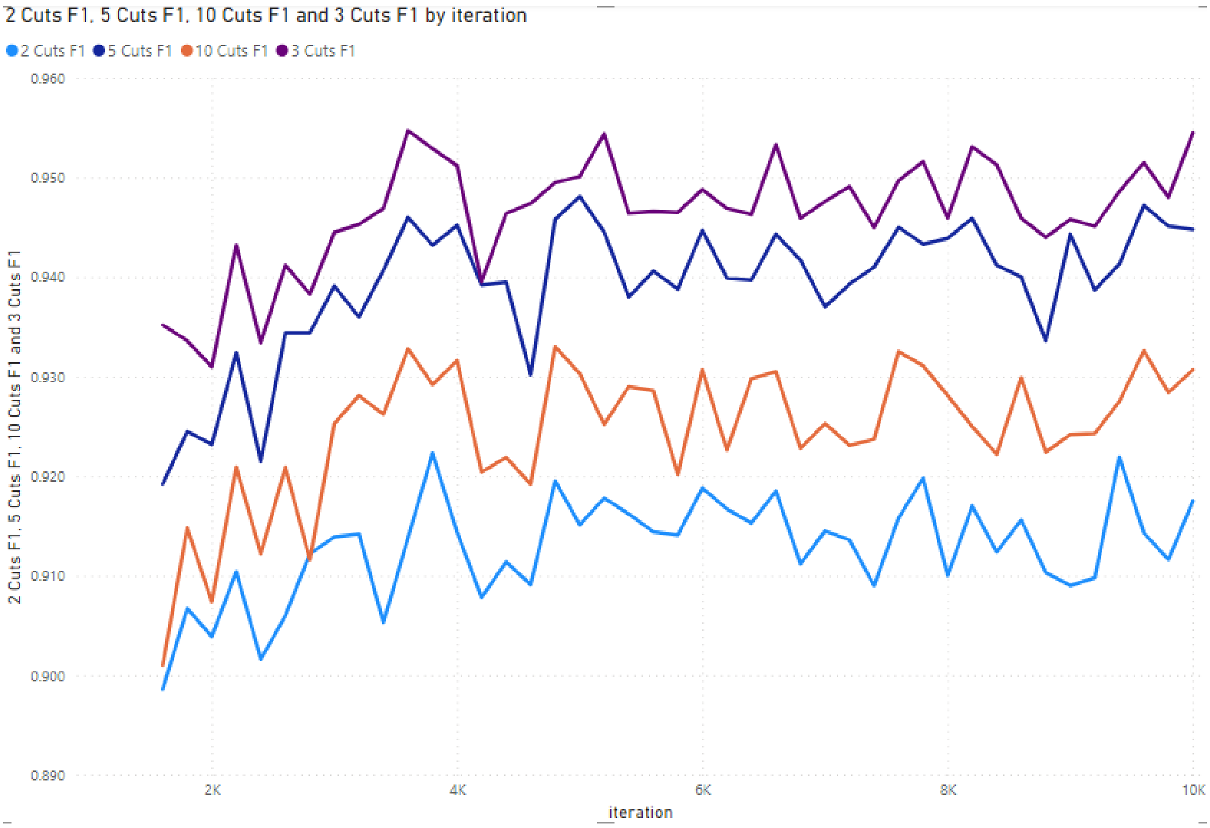

Figure 7: Training Performance over Iterations – 50% Augmented Data

6.3 50 Percent Augmented Data – 4 Folds Macro Averaged

Slices	Fold	F1 Average Score after 2000 Training Iterations
3	1	.95
3	2	.95
3	3	.94
3	4	.95
3	-	.95 Macro Average Total

In this experiment, the network was trained on all four folds of holdout data with each training run delivering a F1 Score. The F1 scores from all folds are then averaged to produce a macro average F1 score of 0.956.

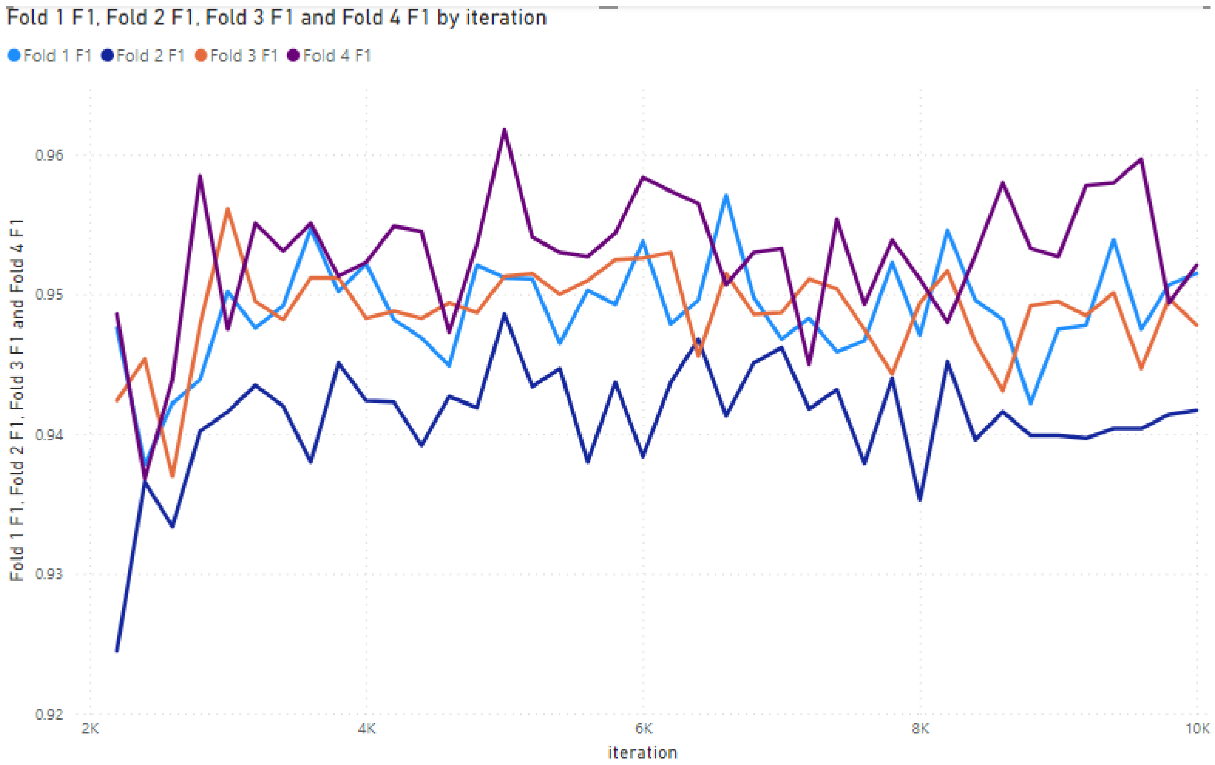

Figure 8: Training Performance over Iterations – 50% Augmented Data – All Four Folds

7 Conclusion

The aim of this research was to critically evaluate the proposed data augmentation techniques from Inoue. The evaluation took the form of a replication experiment and implementation of Inoue's algorithm with tunable parameters. Performance metrics were captured through the training of a neural network architecture driven by an automated augmentation generator. The metrics indicate the proposed algorithm from Inoue can be further optimized for the SINS dataset by reducing the amount of audio segment division to three rather than five. The metrics also show performance degrades at slice values higher than five. In evaluating the performance of all four folds, the research observed a macro average score of 0.94. This result exceeds Inoue's performance on the development dataset along with other task entrants from the 2018 Task 5 challenge. This result was achieved using 80% less training epochs when compared to Inoue's work. The boost in classification performance clearly demonstrates the benefit of observing learning metrics of a neural network using this information to influence your data augmentation approach. This work shows the promise and the applicability of Inoue's work even in the face of different neural network architectures and software implementations.

8 Further Work

For further work, this research can be extended by taking this slicing approach this and applying it to other labelled audio event datasets and any future DCASE challenges with a focus on low-quality or limit data training scenarios. This research could further be extended by adapting the algorithm to introduce variance into the slicing, such as producing unequal slices or subdividing files through a range of values. This future work would be able to make deeper inferences into how a neural network learns from audio by adapting the length of audio segments based on the underlying sound class.

9 References

[1] A. Copiaco, C. Ritz, N. Abdulaziz, and S. Fasciani, “Identifying Optimal Features for Multi-channel Acoustic Scene Classification,” in *2019 2nd International Conference on Signal Processing and Information Security (ICSPIS)*, Oct. 2019, pp. 1–4. doi: 10.1109/ICSPIS48135.2019.9045907.

[2] D. Henze, K. Gorishti, B. Bruegge, and J.-P. Simen, “AudioForesight: A Process Model for Audio Predictive Maintenance in Industrial Environments,” in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, Dec. 2019, pp. 352–357. doi: 10.1109/ICMLA.2019.00066.

[3] T. Inoue *et al.*, “DETECTION OF ANOMALOUS SOUNDS FOR MACHINE CONDITION MONITORING USING CLASSIFICATION CONFIDENCE,” 2020.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[5] G. Dekkers *et al.*, “The SINS database for detection of daily activities in a home environment using an Acoustic Sensor Network,” in *Detection and Classification of Acoustic Scenes and Events 2017*, 2017, pp. 1–5. [Online]. Available: Uhttps://lirias.kuleuven.be/retrieve/525662D18-151.pdf

[6] C. E. Shannon, “Communication in the Presence of Noise,” *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, Jan. 1949, doi: 10.1109/JRPROC.1949.232969.

[7] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, “Very Deep Convolutional Neural Networks for Raw Waveforms,” Oct. 2016.

[8] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, “Learning the speech front-end with raw waveform CLDNNs,” in *Interspeech 2015*, Sep. 2015, pp. 1–5. doi: 10.21437/Interspeech.2015-1.

[9] Y. Su, K. Zhang, J. Wang, and K. Madani, “Environment Sound Classification Using a Two-Stream CNN Based on Decision-Level Fusion,” *Sensors*, vol. 19, no. 7, p. 1733, Apr. 2019, doi: 10.3390/s19071733.

[10] L. Wyse, “Audio Spectrogram Representations for Processing with Convolutional Neural Networks,” Jun. 2017.

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

[12] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[13] J. Salamon and J. P. Bello, “Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification,” Aug. 2016, doi: 10.1109/LSP.2017.2657381.

[14] L. Perez and J. Wang, “The Effectiveness of Data Augmentation in Image Classification using Deep Learning,” Dec. 2017.

[15] T. Inoue, P. Vinayavekhin, S. Wang, D. Wood, N. Greco, and R. Tachibana, “Domestic Activities Classification Based on CNN Using Shuffling and Mixing Data Augmentation,” Sep. 2018.

[16] N. Takahashi, M. Gygli, B. Pfister, and L. van Gool, “Deep Convolutional Neural Networks and Data Augmentation for Acoustic Event Detection,” Apr. 2016.

[17] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond Empirical Risk Minimization,” Oct. 2017.

[18] Y. Tokozume, Y. Ushiku, and T. Harada, “Learning from Between-class Examples for Deep Sound Recognition,” Nov. 2017.

[19] R. Zhang, H. Wu, W. Li, D. Jiang, W. Zou, and X. Li, “Transformer based unsupervised pre-training for acoustic representation learning,” Jul. 2020.

[20] S. Amiriparian *et al.*, “Towards cross-modal pre-training and learning tempo-spatial characteristics for audio recognition with convolutional and recurrent neural networks,” *EURASIP J Audio Speech Music Process*, vol. 2020, no. 1, p. 19, Dec. 2020, doi: 10.1186/s13636-020-00186-0.

[21] Q. Kong, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, “DCASE 2018 Challenge Surrey Cross-Task convolutional neural network baseline,” Aug. 2018.

[22] G. Dekkers, L. Vuegen, T. van Waterschoot, B. Vanrumste, and P. Karsmakers, “DCASE 2018 Challenge - Task 5: Monitoring of domestic activities based on multi-channel acoustics,” 2018. [Online]. Available: https://arxiv.org/abs/1807.11246