EE476 Audio-Visual Perception Model

Term Project

Bereket Eshete   20150923

Due: 11:59 pm, June 18, 2018

Data augmentation and Explainable AI
For improving neural networks

1. Introduction

In homework four, we fine-tuned using fixed weights from CNN whereas for this term project, we fine-tune all the four weights of audio classification neural network. Fine-tuning is the process of updating the weights usually using stochastic gradient method, as we implement here.

Data augmentation is an algorithm implementation using limited resource data for training. There are three main themes in data augmentation. The first one is linear/non-linear transformation of data, which includes varying size, rotation, shitting. Scaling, addition, and subtraction of noise to data are also included in this category. The second method is nonlinear data dimension reduction; this includes LLE (locally linear embedding and auto encoder). The third one is data from nearest neighbors. Data augmentation has been shown to produce promising ways to increase the accuracy of classification tasks. [1] We would expect that such data augmentation techniques might be used to benefit not only classification tasks lacking sufficient data, but also help improve the current state of the art algorithms for classification. For my project, I will focus on linear/non-linear transformation of data for data augmentation.

The second new idea is relatively new-fangled ‘Explainable AI’. Using this concept, I will try to explain which neurons and neural connection is responsible for specific learning process. A New Approach Create a suite of machine learning techniques that produce more explainable models, while maintaining a high level of learning performance. Modified deep learning techniques to learn explainable features [2]. Researchers at UC Berkeley have recently extended this idea to generate explanations of bird classifications. The system learns to: Classify bird species with 85% accuracy. Associate image descriptions (discriminative features of the image) with class definitions (image-independent discriminative features of the class). However, there are some limitations to this idea. Firstly, they are Limited (indirect at best) explanation of internal logic. Second, they are Limited utility for understanding classification errors.

Generally comparing with other existing algorithms. Data augmentation is an easy and practical implementation to improve performance whereas explainable AI gives us better understanding of network in solving the mystery of deep learning black box. I will first describe

2. Theory/ Methods (1-2 page)

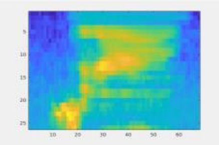
- Fine-tuning, we fine-tune the CNN and DNN weights for supervised learning. main_hw4_audio_together tunes all the weights together; however, the test data accuracy is 10% and takes a minimum of one day to train as every iteration is 50000 it has to go through all the four layers. These is very inefficient to study behavior of the network, hence I choose to tune the weights separately. This way we fine-tune all the weights in the four layers.

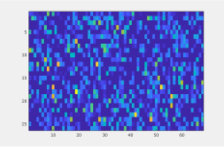
- Data augmentation

  For data augmentation, I will use optimum parameters possible to yield best results. I will use PCA feature extraction because it is much faster than sae and I will use max global pooling because it yields better accuracy. This conclusion is from Homework 4. I want to use faster extraction method because data augmentation requires test various training data frequently.

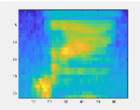
  - Additive/multiplicative noise
    We will use Gaussian noise as our first data augmentation experiment. We can further intensify our training data set by varying the mean and variance of Gaussian noise. Gaussian noise increases accuracy by making the accuracy of classification less dependent on pixel value variation and to focus on the general content of the image. By using matlab built in function, we add Gaussian noise to the training data, making our data set twice as much data. However, in image data containing small pixels, Gaussian noise may cover up too much of the training data leading to weaker performance.

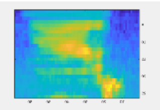
    J = imnoise(I,'gaussian',m,var_gauss) … by default Gaussian noise of zero mean and 0.01 variance

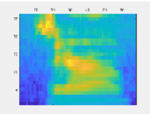




  - Increasing data size by repeating the same data as training data
    Use the same data repeatedly by expanding the training data set; even though this method is trivial, dull and barely improves accuracy, it can help us raise the accuracy by single digit number when supplying repeated data multiple times as training input.
  - Flipping
    This is another technique to increase the size of the data set, flipping is the same as rotating an image 180 degrees along a certain axis. This method is especially widely used in Image recognition to increase accuracy and reliability.
    m_hor=flip(I,2);… horizontal flip
    m_ver=flip(I,1);….Vertical flip



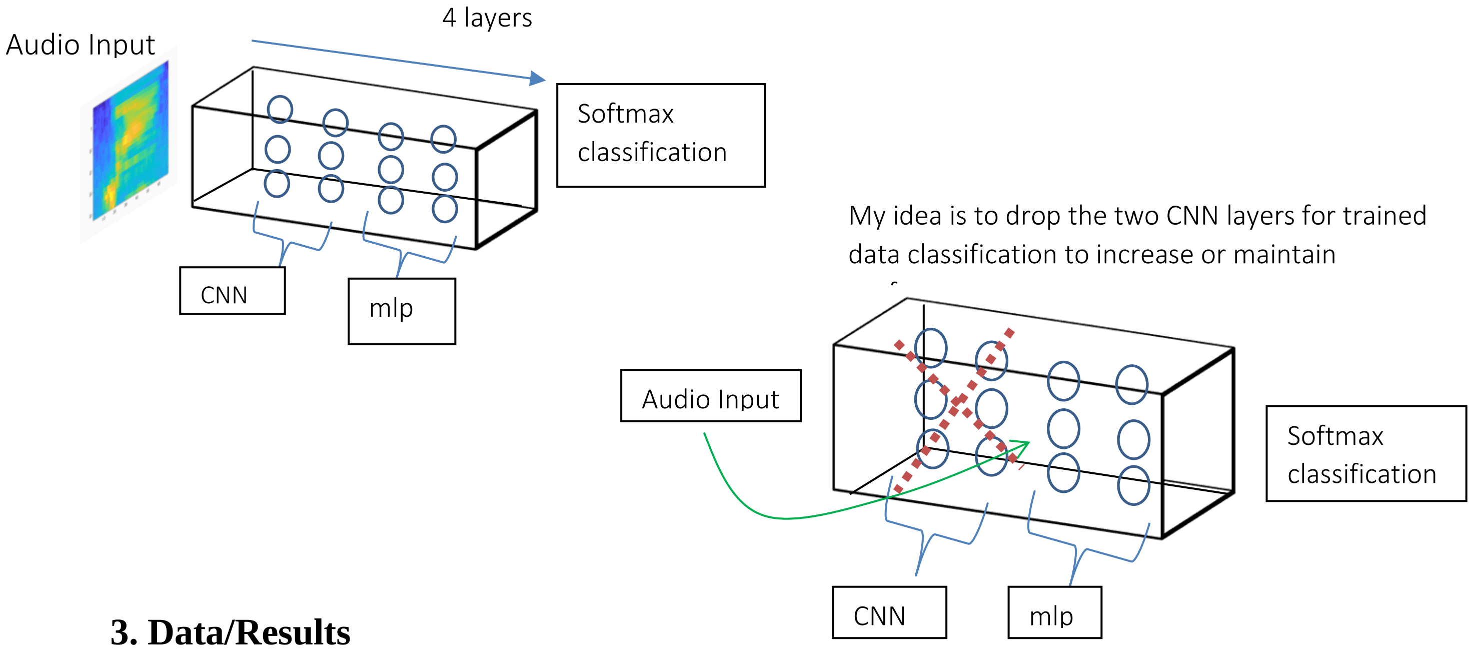




  - Resizing/Scaling – This technique involves stretching or compressing of training image by a factor of two to increase training data. The following commands will compress and stretch the width of the training image respectively.
    m_half=imresize(m,[size(m,1) size(m,2)/2]);
    m_twice=imresize(m,[size(m,1) size(m,2)*2])

- Explainable AI

In explainable AI, we try to understand the identify the weakness of the network using glass model and test if it improves accuracy. In my case, I will single out the CNN layer and try to make hypothesis to describe the function of the CNN layer. Hence, we first measure the accuracy of the four layers, and then we remove the CNN and use only mlp to produce accuracy results. By comparing these two, we might be able to understand what the CNN layer function is and whether it is useful or if we can drop the CNN layer and in what conditions can we drop the layer?

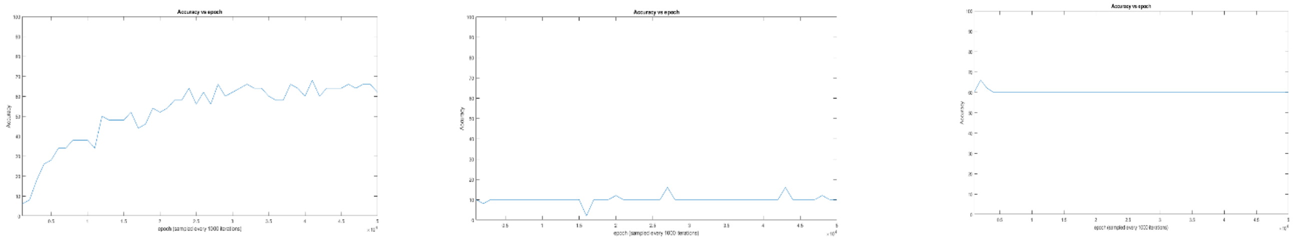


My idea is to drop the two CNN layers for trained data classification to increase or maintain

3. Data/Results

I. Fine-tuning all CNN + DNN weights

Starting from the results of Homework 4, we fine-tune synaptic weights for all 4-layers by supervised learning, and analyze the results for audio classification tasks in the discussion section. We will use the following default parameters for training: learning rate 0.02, 50000 iteration, 1000 check valid frequency, 96 hidden layer neurons, 0.01 standard deviation to initialize weights. Starting with our first result let us compare fixed CNN weights and all four fine-tuned weight in the figures below.



(a) (b) (c)

Figure 3.1. Test data accuracy vs iteration (a) [sae, max] Baseline model using a fixed CNN weights, sparse auto encoder and max pooling (b) [sae, max] after fine-tuning all four weights. (c) [pca, max] after fine-tuning all four weights.

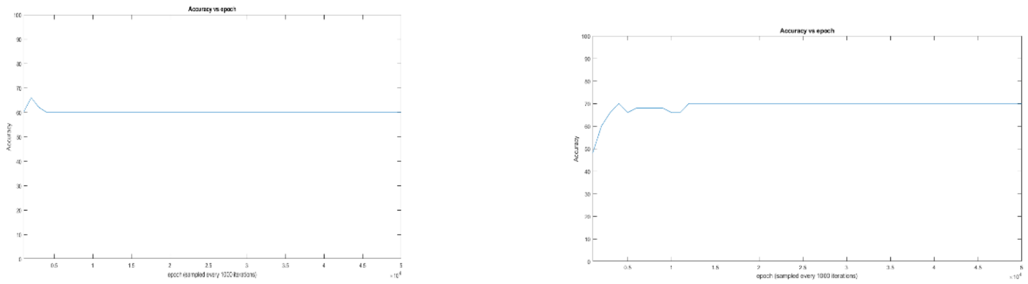


Figure 3.2. Test data accuracy vs iteration, after all four weights are fine-tuned (a) PCA max (b) PCA mean. PCA max converges to 60% accuracy whereas PCA mean converges to 70% accuracy.

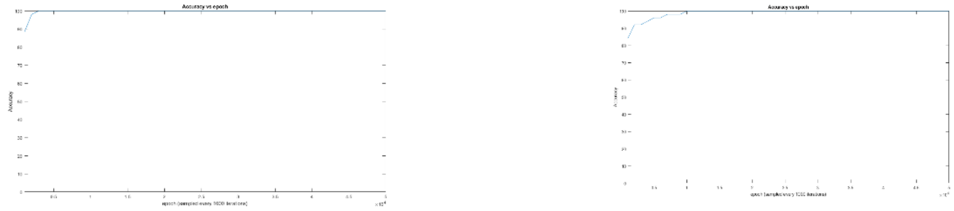
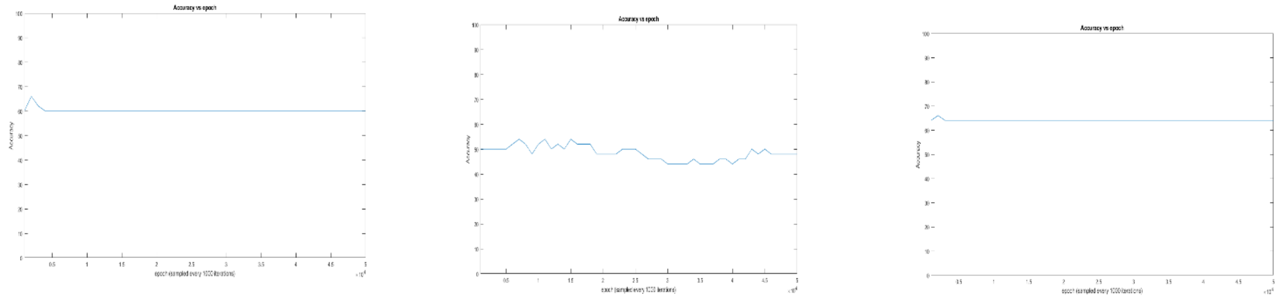


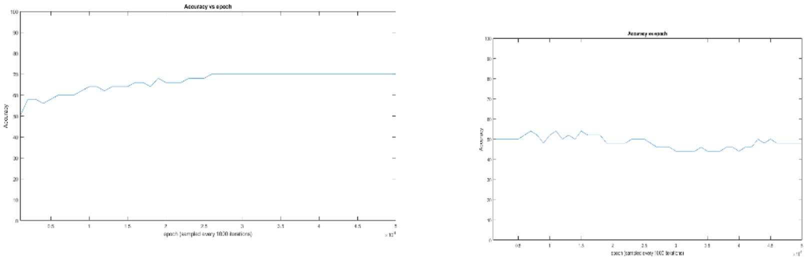
Figure 3.3. Train data accuracy vs iteration, after all four weights are fine-tuned (a) PCA max (b) PCA mean. PCA max converges to 100% faster than PCA mean.

II. Implement data augmentation

After seeing the result of fine-tuning of the four layers. We proceed to the results of data augmentation. One thing to notice here is even though PCA mean performs better alone, PCA max performs better when used together with data augmentation. For instance, PCA mean converges to 66% accuracy whereas PCA max converges to 70% for Gaussian noise augmentation. Similar trend is observed for the other types of data augmentation. Hence, we decide to use PCA max as comparison baseline model to compare data augmentation results.



(a) (b) (c)



(d) (e)

Figure 3.4. Implementing data augmentation to improve test data accuracy (a) Baseline PCA max (b) after Gaussian noise data augmentation (0,0.01) (c) after expanding training data through four times repetition data (2% improvement from (a)) (d) horizontal and vertical flipping data augmentation (e) resizing

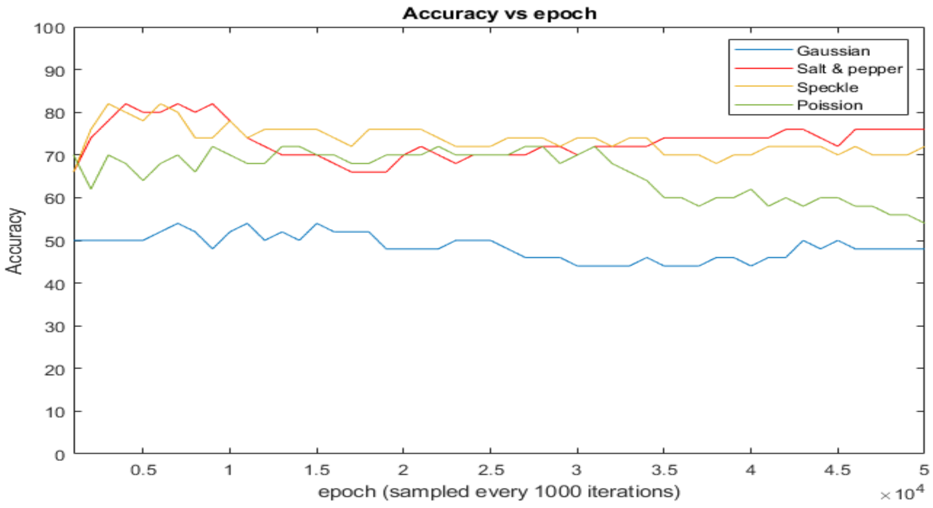


Figure 3.5. Different types of additive/multiplicative noise effect on test data accuracy. Salt and pepper with 0.02 noise density performs the best whereas Gaussian noise with zero mean and 0.01 variance performs the worst.

Accuracy comparison for test data		
Implementation	Method	Best accuracy convergence
Fine tuning	Fixed CNN weights	62%
	Fine-tuned CNN weights	60%
Data Augmentation (Linear/Nonlinear transform)	Shift/ Rotation	70%
	Scaling	48%
	Additive/Multiplicative distortion (Noise)	76%

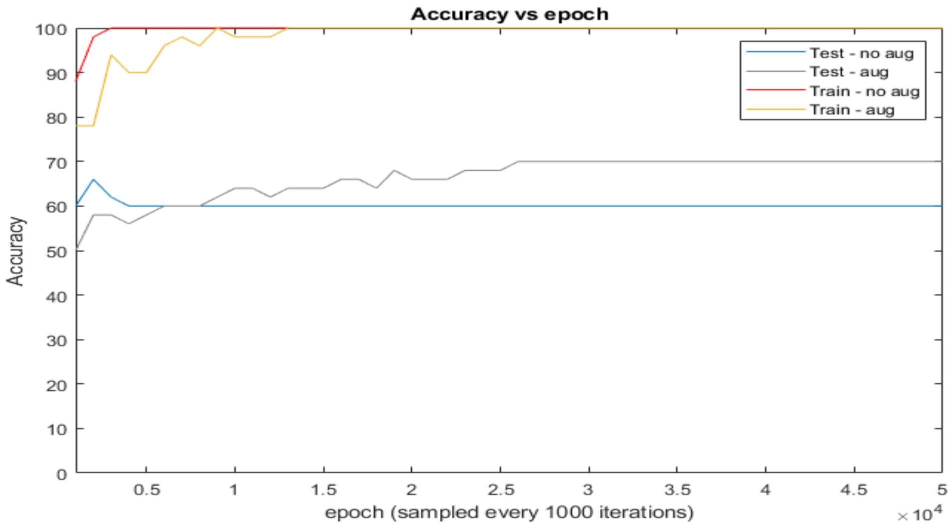


Figure 3.6. Comparison of train and test data accuracy with and without data augmentation. See the discussion for explanation.

III. Modeling explainable AI

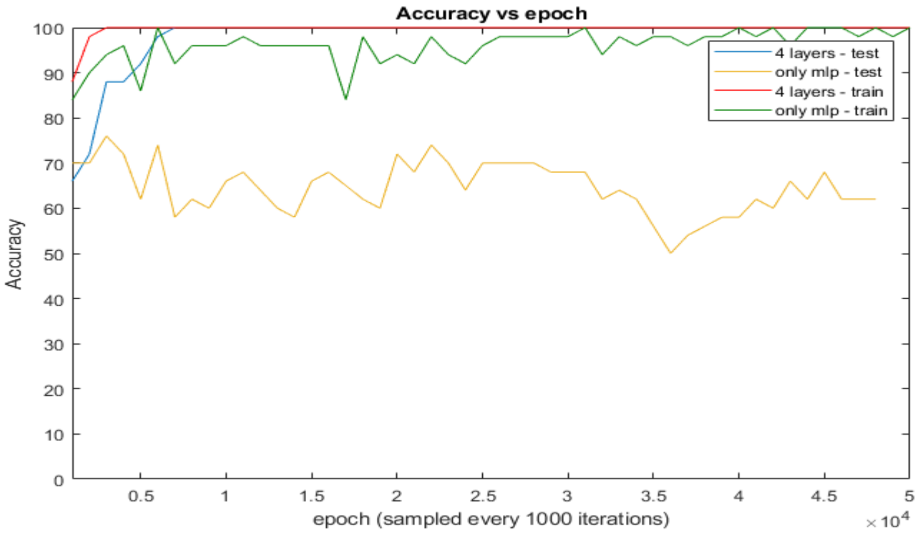


Fig 3.7. Comparing the effect of dropping the CNN layer for test and train data

4. Discussion and Conclusion

Discussion

I. Fine-tuning all CNN + DNN weights

- [Fig 3.1]. observing at (a) and (c) , the test accuracy of fixed weight (a) starts with low accuracy because the CNN weights aren't fine-tuned, however due to the fine tuning of the mlp weights the accuracy starts increasing at an alarming rate. Given enough iterations the accuracy will catch up to the four layer fine-tuned accuracy. The alarming increase in accuracy is because the weights are fixed, and it is just the mlp layer that is active however; (c) remains constant starting with high accuracy as all the weights are being trained in the four layers. We chose not to regard (b) because of its bad accuracy and therefore we will use PCA for most part.

- [Fig 3.2.] PCA mean converges to a higher accuracy than PCA max; hence given other parameters are constant; PCA mean is preferred for test data classification. However, this is not always the case, as we will see later in data augmentation case.

- [Fig 3.3]. Even though PCA mean performs better in test data classification, in train data classification PCA max converges to 100% faster than PCA mean. This might be due to max pooling have better memory of past training data whereas mean pooling handles new incoming data better for audio classification.

II. Implement data augmentation

- [Fig 3.4] We Implement data augmentation to improve test data accuracy we use (a) Baseline PCA max as comparison point. Observing (b) after Gaussian noise data augmentation (0,0.01) it is very clear that the accuracy has degraded comparing to the baseline PCA, by varying the mean and variance of the Gaussian noise we still obtain bad accuracy, hence we should find other additive/multiplicative noise for noise data augmentation to improve our accuracy. After expanding training data through four times repetition data we get (c) this shows 2% improvement from the baseline PCA which relatively is negligible and can be considered as no accuracy improvement. From the data augmentation, we observed that shows significant accuracy improvement is (d) implemented by horizontal and vertical flipping data augmentation. Finally (e) resizing, we stretch and compress the training data width by two. However, the result turns out to be bad similar to the Gaussian noise, the reason might be the varying sizes of the training data initially. The different pixel sizes for the training data, which is not optimal for learning. Researchers usually convert their training data to similar size pixels before their study. Perhaps if all the training data had similar sizes we might have obtained better result.
- [Fig 3.5] Since Gaussian noise was a bad result, we try to find a better noise augmentation. By looking at different types of additive/multiplicative noise effect on test data accuracy. Salt and pepper with 0.02 noise density performs the best whereas Gaussian noise with zero mean and 0.01 variance performs the worst.
- [Fig 3.6] As stated in my proposal my final goal for data augmentation is comparison of train and test data accuracy with and without data augmentation. In addition, compare it with my reference paper to check my result. Hence, comparing the picture below with my result, we observe similar result; the paper was a research on general images and object classification. Interestingly, we realize data augmentation have similar characteristics for audio classification. What this means is that we can apply my plot as data augmentation reference for image related classification. From the figure, we can observe data augmentation improves test data classification accuracy (in my case my 10% using flipping data augmentation). However, that is not the case for train data classification even though both augmented and non-augmented converge to same accuracy, non-augmented converges faster for train data, which also agrees with the figure below from the paper.

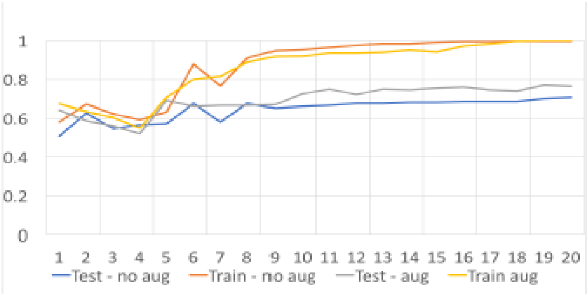


Fig 4.1 [reference] Comparing the effect of data augmentation for train and test data [1] J Wang, L Perez The effectiveness of data augmentation in image classification using deep learning

III. Model explainable AI

- As explained in the theory section, my goal is to compare the result with and without using the CNN layer. My motivation is this might reveal something we do not know about CNN layer.
- [Fig 3.7] For classifying trained data both mlp and cnn+mlp similar convergence accuracy. Hence for audio classification. 'Only mlp' are my models in the figure, the green line on the figure shows my model and shows how it converges to 100% similar with fine-tuned four layers accuracy convergence in red color. Even though the four layer converges faster than my mlp two layer that is the compromise, we have to make by losing two CNN layers and reducing complexity.

For audio classification of trained data, I reach the following hypothesis

$$\boxed{\text{Accuracy convergence}\{\text{CNN} + \text{MLP}\} = \text{Accuracy convergence}\{\text{MLP}\}}$$

i.e. this equivalence doesn't apply for classifying test data,
This result is very essential as it enables us to drop two layer and reduce the complexity of the network for trained data classification.

Conclusion

Finally to conclude I will describe the best and worst case scenario for audio classification. Starting from the worst case, avoid using Gaussian noise for data augmentation, avoid fine tuning PCA together with mlp rather tune them separately else, it should take 5min*50,000 on a fast computer with four GPU's. Data augmentation usually gives us a better accuracy or similar accuracy or rarely bad accuracy. If used correctly it is a very easy method to improve audio classification of test data.

- Use PCA max for test data classification and PCA mean for train data classification
- Salt and pepper noise with 0.02 noise density is the best data augmentation implementation Among what I studied here.
- Accuracy convergence{CNN + MLP} = Accuracy convergence{MLP}
- In doing trained data classification drop the CNN layer (my idea)

References

[1] J Wang, L Perez The effectiveness of data augmentation in image classification using deep learning
[2] D Gunning Explainable Artificial Intelligence (XAI).
[3] EE476 Data augmentation Lecture note Spring 2018
[4] https://simmachines.com/explainable-ai/
[5] Adrian Weller "Changes of Transparency