Data Augmentation and its Effects on Audio Recognition by Neural Nets

1st Grayson Cordell
Department of Computer Science
Middle Tennessee State University
Murfreesboro, United States
glc3k@mtmail.mtsu.edu

2nd Jesse Gailbreath
Department of Computer Science
Middle Tennessee State University
Murfreesboro, United States
jgg2x@mtmail.mtsu.edu

3rd Noah Norrod
Department of Computer Science
Middle Tennessee State University
Murfreesboro, United States
nan2q@mtmail.mtsu.edu

4th Jacob Swindell
Department of Computer Science
Middle Tennessee State University
Murfreesboro, United States
jds2fe@mtmail.mtsu.edu

***Abstract*—Small data sets pose a problem when using neural networks. Limited amounts of data can cause a litany of issues, the largest being the possibility of overfitting a model. One way to improve a model's performance is through the augmentation of the existing data set. This method allows for an artificial expansion of the data set. In our research, we compared the results of sound augmentations on a CNN-LSTM network. The model used was the "free-spoken-digit" data set. The unmodified data set contains 3,000 samples of speakers saying digits zero through nine. Once all augmentations were completed, we reached over 30,000 categorized samples. After testing both data sets, we were able to demonstrate that with augmentation, the performance of our network improved compared to when the non-augmented data set is run through. The augmented data set improved the model's validation accuracy by anywhere from 6 to 21 percent, with an average of around 11 percent. For smaller data sets, attempting augmentation techniques can be very advantageous.**

***Index Terms*—neural nets, audio, data augmentation, CNN, LSTM, spectrogram.**

I. Introduction and Background

While classification using machine learning has been evolving and many different architectures exist, audio classification is often considered a more challenging and less researched area. By applying techniques and methods used for image classification, results in audio classification have been able to achieve solid results. Currently, this method is quite prevalent. There are wide varieties of voice dictation products where audio classification is utilized, but while these corporations have truly immense data sets for training, this is not always the case for researchers. [1] Sometimes, a more niche and specific classification system may be necessary, but without proper data scraping, this proves to be difficult. Small data sets tend to result in training models that are not capable of generalization. Generalization is needed for the effective classification of unseen samples. That's where the idea of data augmentation comes into play. Artificially increasing the amount of available data by creating new instances based on the original data can help compensate for this lack of training data. The new data is created through altering an existing set of data in such a way to produce new instances. [1] In the scope of our project, data augmentation takes the form of changing audio samples through pitch shifting, frequency suppression, additive background noise, and Randomized reverb. Pitch is the more common term for the "frequency" of a soundwave, or, more scientifically, how fast a soundwave modulates, in terms of pressure, between compression and rarefaction. The higher or lower the frequency, the higher or lower the pitch of the sound, respectively. [2] In this case, we have taken the original sound files and either increased or decreased the frequency by preset and random intervals. Frequency suppression incorporates what is known as band stop, which is when a set range of frequencies from a signal is masked or silenced. The additive noise process introduced a static signal and combined it with a target signal. Our randomized reverb effect added a reverb with a randomized "room size" to the signal, creating an echoing effect. Reverb is an effect manipulated in audio production to create an illusion of differently sized spaces. For example, audio that sounds as though it was recorded in a hallway small room, or any other location where an echo could be found. By adding a reverb effect to our audio samples, we can emulate the sound of those spaces. These methods create comprehensible samples that still maintain the speaker's message. With each new sample, we can add it into our data set, and while maybe not quite as adequate as having more untouched samples thus creating a much larger data set than we originally started with.

II. Methods

Our chosen sample set was the relatively small Free Spoken Digit Dataset (FSDD). [4] We used the original set that contains just 3,000 samples of waveform audio (denoted by the file extension ".wav") files from 6 speakers repeating the individual digits 0 through 9 multiple times. All work was performed using Python in Jupyter Notebook format. Augmentations to the samples were done with WavAugment, which is recognized for use with speech. [5] 10 unique

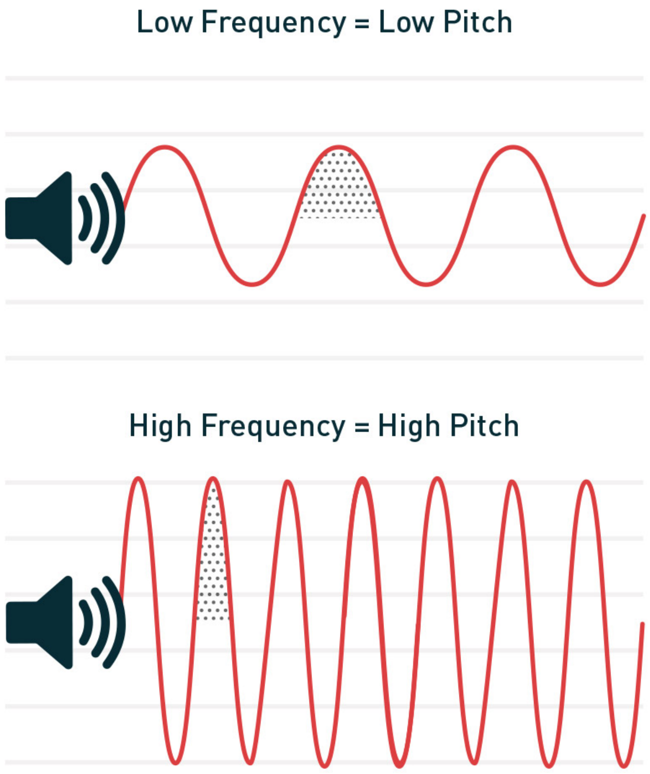


Fig. 1. Diagram showing how frequency affects a sound wave. [3]

augmentations were made to each sample to grow the data set to 33,000 samples. The samples were then processed using

Effects Applied To Samples	
Augmentation Technique	**Resulting Effect**
Clipped	Amplitude is increased beyond representation range, creating distortion
Minus 200 Pitch	Pitch of the sample is decreased by two semitones
Plus 200 Pitch	Pitch of the sample is increased by two semitones
Audio Silence (approx 0.3s)	A section of the sample is masked
Additive Noise	Background noise is introduced to the sample
Band Stop	A set range of frequencies in the sample is blocked out
Random Pitch 1	A random pitch adjustment between -4 to +4 semitones is made
Random Pitch 2	A random pitch adjustment between -4 to +4 semitones is made
Randomized Reverb 1	A reverb with a random room size is added to the sample
Randomized Reverb 2	A reverb with a random room size is added to the sample

Fig. 2. Table showing the different effects we used to augment our samples.

the Librosa package in Python to convert the audio samples from .wav files to spectrograms. [6] Spectrogram images take the audio signals and give us a visual representation of the signal's amplitude, or volume, across multiple frequencies over time. Visually, it is now possible to see what small changes to a single sound sample look like. These three examples are directly from our data set. We can visibly see they are almost identical, but a deep learning model will be able to identify the minor differences. The shape stays consistent in both images, but there are unique shifts that have occurred that can be recognized when processed by a deep learning model. The images were then stored in batch data sets using Tensorflow. This led to the chosen architecture;

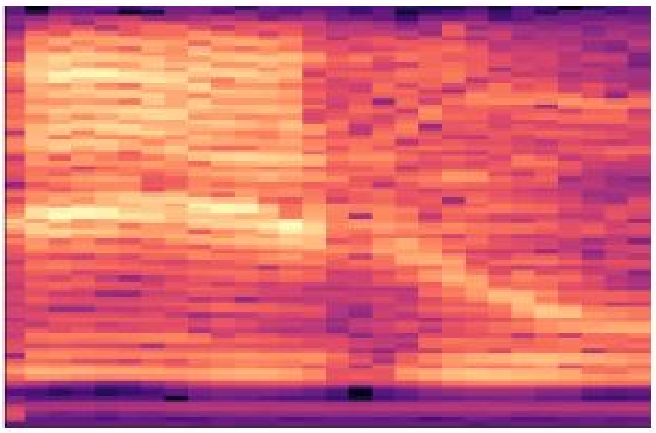


Fig. 3. Spectrogram of speaker saying "zero" from the original data set.

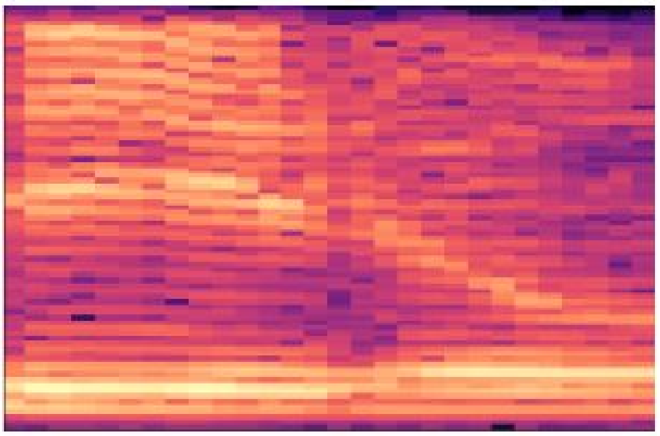


Fig. 4. Spectrogram of speaker saying "zero" after being pitched up two semitones.

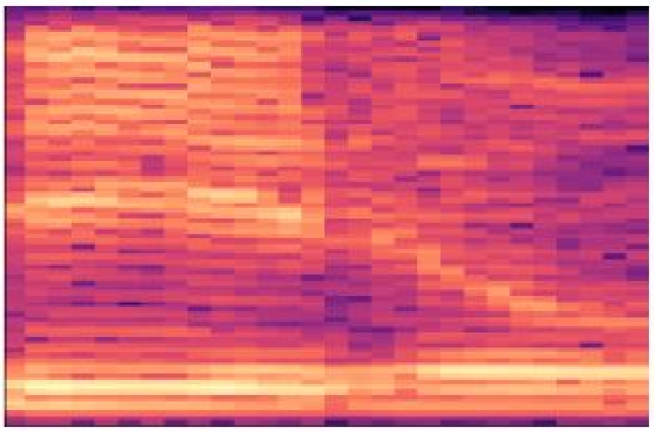
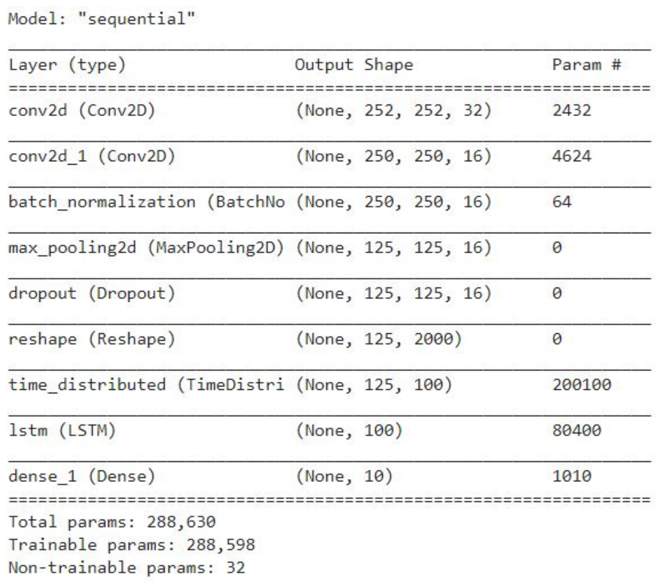


Fig. 5. Spectrogram of speaker saying "zero" after randomized reverb was applied.

a CNN-LSTM. This architecture provides improved feature extraction of the images with the ability to evaluate with respect to time. By using this architecture, we can address the spatial and temporal aspects of audio within the same neural network. The spectrogram images are fed into two convolutional layers, processed for normalization and dropout, and then reshaped into 3-dimensional data. From here, it is moved into a time-distributed fully-connected layer before passing off to an LSTM layer before output. The network was tested using two different amounts of inputs for the layers. This was done to make sure a network with more connections would still produce improved accuracy. The resulting output is a categorical assignment to one of the ten spoken digits. The models were run with a 70/30 training and testing split from each of the data sets.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 252, 252, 32)      2432
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 250, 250, 16)      4624
_________________________________________________________________
batch_normalization (BatchNo (None, 250, 250, 16)      64
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 125, 125, 16)      0
_________________________________________________________________
dropout (Dropout)            (None, 125, 125, 16)      0
_________________________________________________________________
reshape (Reshape)            (None, 125, 2000)         0
_________________________________________________________________
time_distributed (TimeDistri (None, 125, 100)          200100
_________________________________________________________________
lstm (LSTM)                  (None, 100)               80400
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1010
=================================================================
Total params: 288,630
Trainable params: 288,598
Non-trainable params: 32
```

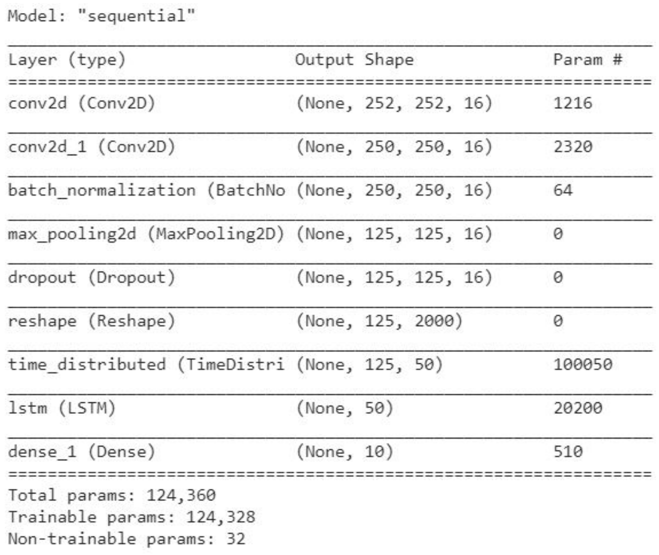
Fig. 6. CNN LSTM model structure.

III. Results

The end result from multiple training sessions are as follows: Upon review, it is evident that the results from the larger augmented data sets show a noticeable improvement over the original. Adjusting the layers did alter the end percentages, but the variance between the data sets is still the same. Using multiple augmentation techniques on an original smaller data set has allowed the network access to more samples, resulting in improved accuracy from the model. After performing seven individual sessions, the average variance for accuracy during testing was 11.45 percentage points in favor of the augmented data set. The most substantial variance was 21.08 percentage points in favor of the augmented set. It should be noted the augmented data set outperformed the original data set each session.

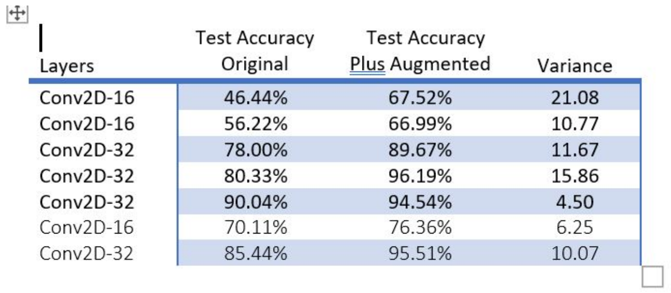
IV. Discussion and Conclusion

The rapid growth in the implementation of machine learning by such a wide range of industries and institutions has shown



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 252, 252, 16)      1216
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 250, 250, 16)      2320
_________________________________________________________________
batch_normalization (BatchNo (None, 250, 250, 16)      64
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 125, 125, 16)      0
_________________________________________________________________
dropout (Dropout)            (None, 125, 125, 16)      0
_________________________________________________________________
reshape (Reshape)            (None, 125, 2000)         0
_________________________________________________________________
time_distributed (TimeDistri (None, 125, 50)           100050
_________________________________________________________________
lstm (LSTM)                  (None, 50)                20200
_________________________________________________________________
dense_1 (Dense)              (None, 10)                510
=================================================================
Total params: 124,360
Trainable params: 124,328
Non-trainable params: 32
```

Fig. 7. CNN LSTM model structure with reduced inputs.



Layers	Test Accuracy Original	Test Accuracy Plus Augmented	Variance
Conv2D-16	46.44%	67.52%	21.08
Conv2D-16	56.22%	66.99%	10.77
Conv2D-32	78.00%	89.67%	11.67
Conv2D-32	80.33%	96.19%	15.86
Conv2D-32	90.04%	94.54%	4.50
Conv2D-16	70.11%	76.36%	6.25
Conv2D-32	85.44%	95.51%	10.07

Fig. 8. Test Accuracy and Variance results.

the need not just for reliable data, but for reliable data in large quantities. Augmentation of data, specifically audio, is proving to be an invaluable tool in achieving this goal. Using these types of methods, as in the field of natural language processing, can bolster the effectiveness of machine learning in ways that would not be possible otherwise. The testing also indicates that using augmented data sets in more complex models would still yield improved accuracy and possibly help prevent overfitting. Not only do these results solidify the value of using data augmentation for improved results, but they also represent a measurable reduction in time and resources needed to amass a quality data set. This alone makes data augmentation a smart choice for research and development whenever possible. Further study into which types of augmentation can lead to the greatest range of improvement would be beneficial. This alone makes data augmentation a smart choice for research and development when possible. Due to the sheer size of the model, the weights lost upon return and time constraints, we did not get the chance to test the augmented dataset model on a non-augmented testing set. Although this was not explicitly seen, we are confident that it would perform better when used on the same testing sets after seeing the current validation achieved. We encourage further study into the degree to which specific types of audio augmentations improve their models.

References

[1] A. Ragni, K. M. Knill, S. P. Rath, and M. J. F. Gates, "Data augmentation for low resource languages."

[2] "Pitch." [Online]. Available: https://www.britannica.com/art/pitch-music

[3] "Gb foam direct."

[4] Jakobovski, "Jakobovski/free-spoken-digit-dataset." [Online]. Available: https://github.com/Jakobovski/free-spoken-digit-dataset

[5] Facebookresearch, "facebookresearch/wavaugment." [Online]. Available: https://github.com/facebookresearch/WavAugment

[6] K. Chaudhary, "Understanding audio data, fourier transform, fft and spectrogram features for a speech recognition system," Jul 2020. [Online]. Available: https://dropsofai.com/understanding-audio-data-fourier-transform-fft-and-spectrogram-features-for-a-speech-recognition-system/