University of Tennessee, Knoxville

RECSEM REU 2024 Final Paper

A Comparative Analysis Of Mel-Scale And STFT Feature Extraction Methods For Audio Classification

Cisco Hadden

Supervised by
Dr. Kwai Wong

August 1, 2024

Nomenclature

ω	Frequency bin index (Hz)
b	Frequency bin index (mels)
f	Frequency (Hz)
FN	False Negative
FP	False Positive
h	hop-length: sample overlap of each STFT window
m	Time frame index
N	Number of training classes
n	number of samples in STFT window
TN	True Negative
TP	True Positive

Table of Contents

1 Abstract

This paper uses a standardized CNN model to compare Mel-Scale and short time Fourier-transform (STFT) spectrogram feature extraction methods to classify urban, human, and bird sounds. While many studies have compared Mel-Scale Spectrograms and STFT Spectrogram features, they often use different datasets and models for each feature. This study addresses the gap in understanding which method is more effective for general audio classification models. After an analysis of a single optimized model trained on urban sounds, bird sounds, and human speech using Mel-Scale spectrogram and STFT spectrogram features it was concluded that STFT features are most effective for a classification model on all audio datasets. It was found that the Mel-Scale features resulted in an undergeneralization of the model to unseen data thus resulting in less accurate classification than when trained with STFT features.

2 Introduction

This paper aims to provide a simple comparison of Mel-Scale and linearly scaled STFT feature extraction methods for classifying audio files. This research will standardize each part of the training process for both feature extraction methods to provide a clear difference in the efficacy of each method. There are many feature extraction methods used in audio classification, each used for a unique purpose. For example Mel-Scale features are commonly used for classification of human speech [1]. This research will use a convolution neural network (CNN) model trained on a variety of datasets to determine which of these feature extraction methods is optimal to use in general applications of audio classification models.

2.1 Background

Audio classification is increasingly used across diverse fields including medicine [2], agriculture [3], and transportation [4]. This growing reliance on classification technology highlights the need for optimizing audio classification models. Enhancements in model efficacy, efficiency, and modularity are becoming crucial. Identifying the most effective audio features for classifying various types of audio is one approach to achieving this optimization.

Before machine learning algorithms can classify audio files, the raw audio must be processed into a numerical format. This process involves extracting features such as energy, frequency, time, and amplitude from the audio waveform and converting it into a tensor [5]. Common feature extraction methods include the STF) and Mel-Scale spectrograms. Both methods provide time-frequency representations but differ in axis scaling and output details (see Mel-Scale Features (2.4) and STFT Features (2.3) for more information).

2.2 Previous Works

Sunee Pongpinigpinyo et al. conducted a similar analysis to this paper utilizing a standardized model and datasets to compare MFCC to Mel-Scale spectrograms for features in audio classification of the juiciness of a pineapple [6]. They concluded that MFCC slightly outperformed the Mel-Scale spectrogram, however both performed well. While generally MFCC and MEl-Scale spectrograms are used for classification of human speach [5], these results show both are effective feature extraction methods for a classification model applied to agriculture. The efficacy of this wider application of Mel-Scale spectrogram is further analyzed in this paper.

Alessandro Terenzi et al. [3] used a model with a similar architecture to compare Mel-Scale spectrogram with STFT, alongside other features, to classify honey bee activity. The results of this paper were comparable to those collected in this paper, however the audio data of honey bee sounds that the model was trained on is much different from human, birds, or urban sounds. This paper expands on the results found in [3] to evaluate these two features on more robust audio datasets using a similar methodology.

Shunsuke Hidaka et al. [7] analyzed the performance of an audio classification model using phase information versus discarding phase information in feature extraction. They found in most datasets including phase information in the audio features resulted in higher performance of the model. Mel-Scale spectrograms discard phase information by taking the magnitude of the data while STFT Spectrograms keep phase information in its complex values. In this paper this is not explicitly explored but is a key difference in the two features and is a possible explanation for the results we collected.

2.3 Short Time Fourier Transform (STFT) Features

The STFT Spectrogram is one of the most common ways to visualize and render audio files for computation. The Fourier transform is a mathematical operation that takes a real valued function as an input and outputs a complex value for each frequency and time index of the input wave. See Equation 1 for the formula below.

This transformation segments the input audio file into several windows of a specific number of samples and processes one window at a time. Each window is made up of 1024 samples and overlaps with the next by 512 samples. This windowing utilizes the Hann function (the first operation within the summation of equation 1). This function is used to minimize discontinuities between the ends of each windowed section.

$$X[\omega, m] = \sum_{k=0}^{n-1} \sin^2\left(\frac{(n)\pi k}{n+1}\right) \cdot \text{input}[m \times \text{h} + k] \cdot e^{\left(-i\frac{2\pi\omega k}{n}\right)}[8] \tag{1}$$

This operation takes the signal within each windowed segment and transforms it from the time domain to the frequency domain. This outputs a complex number encoding phase and magnitude of the signal situated at a specific index of time and frequency of the input wave. See Figure 9 for a visualization of this operation.

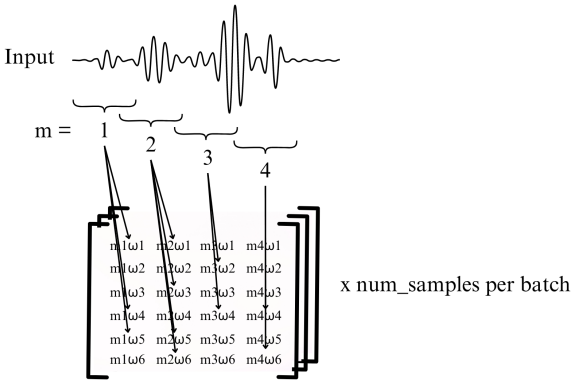

Figure 1: Visual representation of STFT operation

The STFT features for each audio file were computed using the PyTorch spectrogram function in the torchaudio library. This outputs a complex tensor of size [batch size, number of time frames, number of frequency indexes]. The complex values in this output tensor necessitated a slight difference in the models between the STFT and Mel-Scale feature-extraction methods. This is discussed further in the Model Optimization section 3.1. See Figure 2 for a visualization of the STFT spectrogram of a single male and female voice audio file.

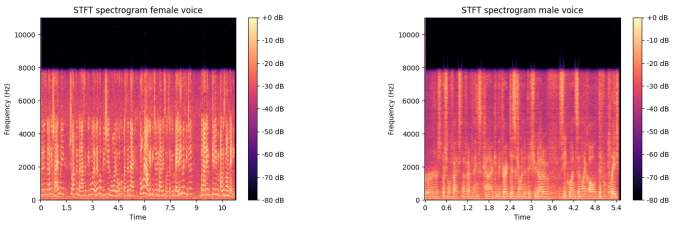

Figure 2: STFT Spectrogram of single male and female audio file [1]

[1] Visualizations of each spectrograms were generated using Librosa display libraries

2.4 Mel-Scale Features

The Mel-Scale features for each audio file were computed using the PyTorch MelSpectrogram function in the torchaudio library. This function first computes the magnitudes of the STFT features as outlined in the section 2.3, thus discarding phase information of the input [7]. The function then converts each frequency from the linear scaled Hertz to mels which is a nonlinear scaling of frequencies. This transformation is computed using Equation 2. This produces a new set of frequency bins that are then passed through 64 mel filterbanks. Each filterbank is modeled by Equation 3 where the center of each filterbank is around f_m [9].

$$\text{mel}(f) = 2595 \log_{10}\left(1 + \frac{700}{f}\right)[9] \tag{2}$$

$$H_m(b) = \begin{cases} 0 & \text{if } b < f_{m-1} \\ \frac{b - f_{m-1}}{f_m - f_{m-1}} & \text{if } f_{m-1} \leq b < f_m \\ 1 & \text{if } b = f_m \\ \frac{f_{m+1} - b}{f_{m+1} - f_m} & \text{if } f_m < b \leq f_{m+1} \\ 0 & \text{if } b > f_{m+1} \end{cases}[9] \tag{3}$$

The purpose of mel-scaling is to mimic human auditory capabilities and compress the higher frequencies in the sample to fit within the range in which humans can hear [10]. This compression of the higher frequencies can be seen in the Mel-Scale Spectrograms in Figure 3. This can especially be observed in comparison to the STFT Spectrograms in Figure 2. The STFT Spectrograms are very dense at high frequencies with a large band of dark coloration across the top layer of the frequency scale indicating a high decibel level. In the Mel-Scale Spectrograms this dark layer is not present in the higher frequency range as the higher frequencies in Hertz are logarithmically transformed and lowered in Mels. Thus, the dense layer of high frequency sound in the STFT spectrogram is spread out and lowered in the Mel-Scale Spectrogram.

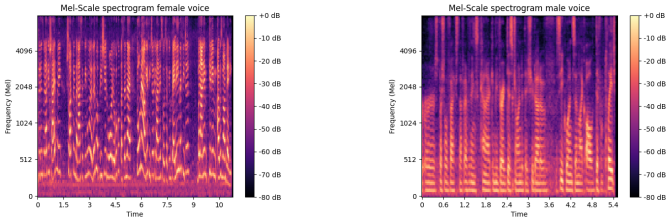

Figure 3: Mel-Scale Spectrogram of single male and female audio file [1]

3 Methodology

The first step in comparing Mel-Scale and STFT features is to optimize a model that can be used on both spectrograms. This CNN model was derived from implementation of a PyTorch model for audio classification GitHub repository by

Valerio Velardo [11]. See section 3.1 for more information on model optimization. Once the model was optimized it was trained on several datasets of various sound types and labels. See section 3.2 for more information on the datasets used for training and verification. Lastly, each feature method was evaluated on a number of different parameters and data points.

The CNN model used in this research is a simple 6 layered model. Each layer consists of a 2 dimensional convolution, non-linearization using the Rectified Linear Unit (ReLU) function, 2 dimensional batch normalization and a pass through a 2 dimensional Max Pool operation. Layers 1, 3, and 5 undergo pacial dropout on layer p = 0.07. The last layer outputs 128 channels, doubling after every layer. We used the Adam optimizer function and the Cross Entropy loss function.

3.1 Model Optimization

Each iteration of the CNN model was tested on all three datasets and both Mel-Scale and STFT features before making a change. The goal is to maximize accuracy in classification for each dataset and for each feature. To visualize the progress of each model iteration the validation curve for accuracy and losses were produces every time the model was run. An inferencing program was also run on training and validation datasets and computed the accuracy, precision, recall, F1 score, and Matthew Correlation Coefficients (MCC) on the results.

The first modification made to the model was implemented to handle the complex outputs of the STFT feature extraction. For this change the model applied to the STFT features takes in 2 channels as the initial input. One channel takes the real part and one takes the imaginary part of the STFT input tensor. The model for the Mel-Scale features has only one input channel as the input tensor is real valued. This modification was necessary to train on the complex values created by the STFT features. To alter the STFT features to be real valued would require computing its magnitude or phase which would would decrease the information the model is trained on and potentially negatively impact the model's performance [7]. While this change does not alter the overall architecture of the model, it is acknowledged as a limitation in this research and should be noted in future work.

The model in [11] used four layers however we found it necessary to increase complexity of the model [12]. This helped to mitigate extreme oscillations in both validation and training losses that were previously observed between epochs. Additionally, to aid in this effort the input data was normalized before training and a normalization function (BatchNorm2d) was implemented within each layer [13]. Lastly, a scheduler (StepLR) was implemented to decrease the learning rate by a factor of 0.5 every 5 epochs. The learning rate was set to 0.001 at the start of the training program to minimize initial oscillations in losses.

The model began over fitting on more complex data in the bird and urban datasets. To prevent this early stopping and spacial dropout functions were implemented. The early stopping function ran with patience of five epochs and minimum difference in validation loss of 0.01. The spacial dropout (Pytorch's BatchNorm2d) function with a dropout layer of 0.07 was applied to the first, third, and fifth layer [14].

3.2 Datasets

The model was trained and tested on the following three datasets:

- 5993 human speech audio sounds from [15] with 2 classifications: male and female.
- 8728 urban audio sounds from [16] with 10 classifications: dog bark, children playing, car horn, air conditioner, street music, siren, engine idling, jackhammer, drilling, and gun shot.
- 5424 bird sounds from [17] with 5 classifications: Bewick's Wren, Northern Cardinal, American Robin, Song Sparrow, and Northern Mockingbird.

The Human Sound and Urban Sound datasets were augmented once to double the number of files the model was trained on. The Bird Sound dataset was determined to be too small to accurately classify five classes, so this dataset was augmented twice to triple the initial dataset size. The augmentation process applied one random function from torchaudio's PitchShift, FrequencyMasking, TimeMasking, Resample, Vol(gain=0.5), and Vol(gain=1.5) to each file in the dataset.

The validation datasets for the Urban and Bird audio pulled 10 audio files from each class from the initial dataset before. The Human Sound dataset was validated using a completely different audio dataset of 39 files from [18].

4 Results

The performance metrics evaluated to compare the Mel-Scale and STFT models were computed using the sklearn python library. The computations are as follows:

$$\text{Accuracy} = \frac{\text{correct}}{\text{total}} \times 100$$

$$\text{Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i}$$

$$\text{Recall Score} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FN_i}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}}$$

The performance of this model trained on each feature and dataset are shown in table 1. Overall it can be seen that the STFT features outperformed the mel-scale features for every validation dataset. The STFT features also saw a narrower margin between training and validation performance. It is hypothesised that the greater difference observed from the Mel-Scale model in validation vs training performance was due to undergeneralization. As the same model architecture was applied to both features this undergeneralization would be an effect of the specifics of the Mel-Scale Spectrogram features in comparison to the STFT Spectrogram features. This could be due to the lack of phase information as it was found by [7] that discarding of phase information for audio classification can weaken a model's performance. There is not enough data to support this hypothesis for these results, but one could further determine the cause of this to be the dropping of the phase information for each frequency when the Mel-Scale Spectrogram calculation takes the magnitude of the STFT features. Further research could explore this and is discussed in section 5.2.

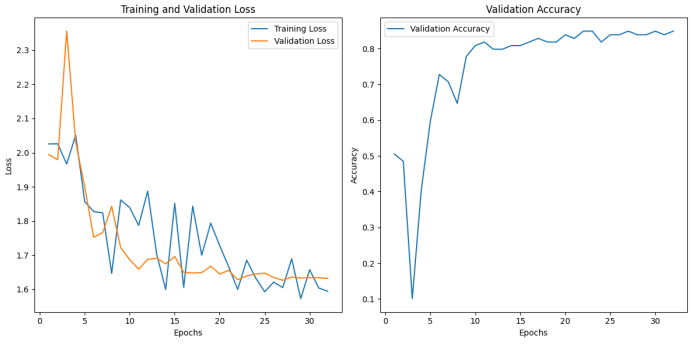

Figure 4: Validation curve for Mel-Scale features model trained on Urban Sound Dataset

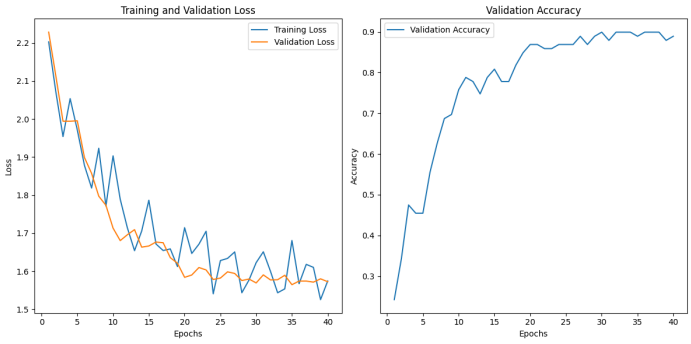

Figure 5: Validation curve for STFT features model trained on Urban Sound Dataset

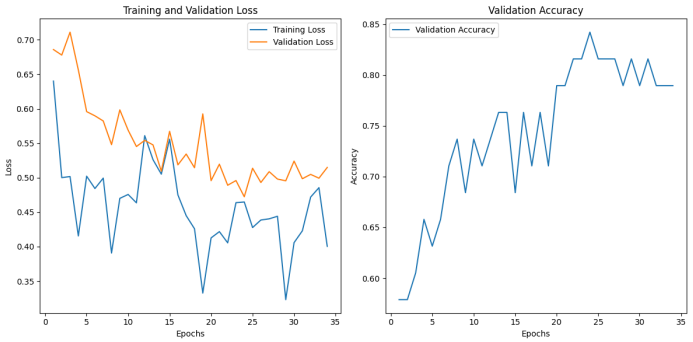

Figure 6: Validation curve for Mel-Scale features model trained on Human Sound Dataset

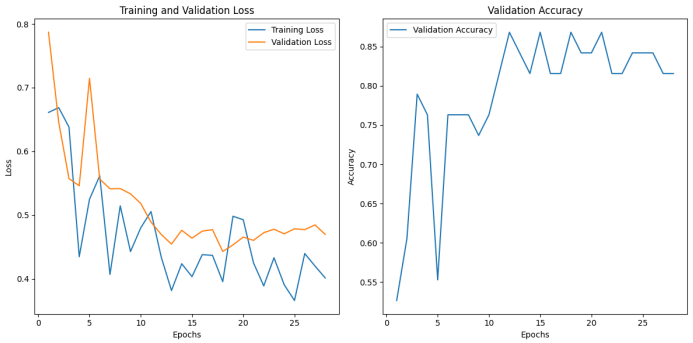

Figure 7: Validation curve for STFT features model trained on Human Sound Dataset

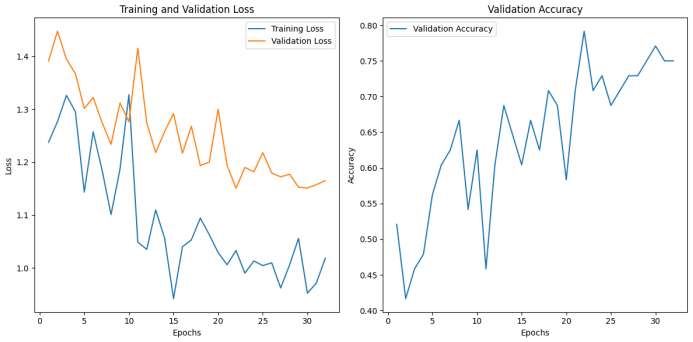

Figure 8: Validation curve for Mel-Scale features model trained on Bird Sound Dataset

Table 1: Model Performance for Mel-Scale and STFT Features

Audio Type	Dataset	Feature	Accuracy	Precision	Recall	F1 Score	MCC
Urban	Train	STFT	97.40%	0.98	0.98	0.98	0.97
		Mel-Scale	93.13%	0.94	0.93	0.93	0.92
	Val	STFT	88.89%	0.90	0.89	0.88	0.88
		Mel-Scale	84.85%	0.87	0.85	0.94	0.84
Human	Train	STFT	95.76%	0.96	0.96	0.96	0.91
		Mel-Scale	94.33%	0.94	0.91	0.93	0.88
	Val	STFT	86.84%	0.87	0.87	0.87	0.74
		Mel-Scale	85.71%	0.90	0.82	0.86	0.72
Bird	Train	STFT	98.50%	0.98	0.98	0.98	0.98
		Mel-Scale	99.16%	0.99	0.99	0.99	0.99
	Val	STFT	75.00%	0.78	0.75	0.76	0.69
		Mel-Scale	79.17%	0.79	0.79	0.78	0.74

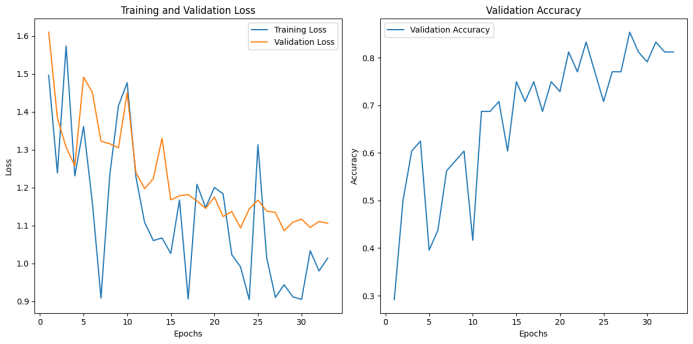

Figure 9: Validation curve for STFT features model trained on Bird Sound Dataset

The validation curves show a very oscilatory change in loss and accuracy as the training progresses. This was minimized through the optimization process, as described in section 3.1 but was not mitigated completely. The spikes in these plots illustrate that there is room for further optimization of the model. This is a limitation in these results and must be considered when referencing this data. However, the bird and human datasets exhibit significantly steeper peaks compared to the urban dataset. This can be partially attributed to their higher initial accuracy, which results from having fewer classifications for the model to train on. Consequently, the learning curve for these datasets is much flatter compared to the urban dataset, which is trained on 10 classifications and starts with an initial accuracy of approximately 10%. However, the overall shape of the validation curve is near expected and the training data is comparable to previous works such as those found in [3], [4], and [7]. Therefore we can report sufficient confidence in this model.

5 Conclusion

In this paper the efficacy of the use of Mel-Scale Spectrogram and STFT Spectrogram features were compared using the same model and three audio datasets. The results show that the STFT features outperformed the Mel-Scale features in the validation dataset and narrower margin between the efficacy in inferencing on the training and validation datasets. As every step of the training process is standardized within the process of training on Mel-Scale and STFT features, this discrepancy is due solely to the feature extraction methods. It can be concluded that STFT outperformed Mel-Scale features for all three datasets. Thus this paper determined STFT spectrograms to be a more suitable feature extraction method for an audio classification model used to classify a variety of audio types.

5.1 Limitations

Limitations of this conclusion include the limited datasets used and room for further optimization of the CNN model. Further information could be gathered from collecting data on a larger variety of dataset types and sized. Lastly, while this model was optimized for this paper and we are confident in its results, there were limitations in the processing power, memory, and time used to optimize the model. Thus a more optimal model could be developed to perform this study and may improve this data collection with more powerful resources.

5.2 Future Work

With more processing power and time this CNN model could be further optimized and potentially obtain a more precise comparison. This research was limited to a 4060 GPU with 8GB memory. This limited the size of each dataset and batch size ran through the model. Future work is suggested to perform a similar analysis on various other features. There may exist a feature extraction method that would prove to be more effective on a wider range of audio files that was not studied in this paper.

One further advancement that could me made to this research is to compare complex valued STFT features with real valued magnitude, power, and phase features from the STFT spectrogram. One could also explore a possible Mel-Scale spectrogram that produces a complex valued tensor, thus preserving phase and magnitude information.

6 Acknowledgments

This project was supported by the National Science Foundation through the Research Experience for Undergraduates (REU) award no. 2020534. We acknowledge the University of Tennessee Knoxville for use of their offices and computing resources.

References

[1] Paul Pedersen. The mel scale. *Journal of Music Theory*, 9(2):295–308, 1965. Accessed 15 July 2024.

[2] Mohammed Aly and Nouf Saeed Alotaibi. A novel deep learning model to detect covid-19 based on wavelet features extracted from mel-scale spectrogram of patients' cough and breathing sounds. *Informatics in Medicine Unlocked*, 32:101049, 2022.

[3] Alessandro Terenzi, Nicola Ortolani, Inês Nolasco, Emmanouil Benetos, and Stefania Cecchi. Comparison of feature extraction methods for sound-based classification of honey bee activity. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:112–122, 2022.

[4] Lin Wang and Daniel Roggen. Sound-based transportation mode recognition with smartphones. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 930–934, 2019.

[5] Garima Sharma, Kartikeyan Umapathy, and Sridhar Krishnan. Trends in audio signal feature extraction methods. *Applied Acoustics*, 158:107020, 2020.

[6] Sunee Pongpinigpinyo, Suphachai Phawiakkharakun, and Pinyo Taeprasartsit. Acoustic sensing for quality edible evaluation of sriracha pineapple using convolutional neural network. *Current Applied Science and Technology*, 10(1):22–33, 2021.

[7] Shunsuke Hidaka, Kohei Wakamiya, and Tokihiko Kaburagi. An investigation of the effectiveness of phase for audio classification. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3708–3712, 2022.

[8] torch.stft - pytorch 2.3 documentation. `https://pytorch.org/docs/stable/generated/torch.stft.html`.

[9] Haytham Fayek. Speech processing for machine learning: Filter banks, mel-frequency cepstral coefficients (mfccs) and what's in-between. Online, April 2016.

[10] L. Roberts. Understanding the mel spectrogram. Medium, January 17 2024.

[11] Valerio Velardo. pytorchforaudio. `https://github.com/musikalkemist/pytorchforaudio`, 2021.

[12] Mayank R. Kapadia and Chirag N. Paunwala. Improved cbir system using multilayer cnn. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 840–845, 2018.

[13] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[14] Sanghun Lee and Chulhee Lee. Revisiting spatial dropout for regularizing convolutional neural networks. *Multimedia Tools and Applications*, 79(45):34195–34207, 2020.

[15] Abid Ali Awan. voice_gender_detection. `https://dagshub.com/kingabzpro/voice_gender_detection`, 2022.

[16] J. Salamon, C. Jacoby, and J. P. Bello. A dataset and taxonomy for urban sound research. In *22nd ACM International Conference on Multimedia*, Orlando, USA, November 2014.

[17] Sharing wildlife sounds from around the world.

[18] Freesound. Find any sound you like, n.d. Accessed: 2024-07-31.