HIGH SCHOOL EDITION
Journal of Student Research

Musical Instrument Identification Using Machine Learning

Evelyn Ding[1] and Emily Sharma[#]

[1]Plano West Senior High School, USA
[#]Advisor

ABSTRACT

Music is a beautiful form of expression that is universally understood. In fact, people across all cultural backgrounds, even those without extensive exposure to musical education, can easily recognize the unique sounds of different instruments. This paper explores the use of machine learning to identify various instruments. The scope of this project focuses on audio recordings with one instrument playing one pitch, and the specific instruments in this study were viola, piano, and ukulele. This paper analyzes ways to quantify the differences in sounds of instruments using the harmonic frequency content apparent in spectrograms. It proposes the use of a simple but efficient K-Nearest-Neighbors machine learning algorithm. The results achieved 80% accuracy; using a larger dataset and using convolutional neural networks could improve accuracy of classification. The machine learning algorithm can be applied to the broader world of sound classification, and it can eventually surpass a human's ability at identifying sounds (e.g. telling apart viola from violin, which the average human cannot do). There are numerous real-world applications for musical instrument recognition. It can enable automatic sorting and searching of massive musical collections with ease, which could be useful to streaming platforms that provide personalized recommendations to their users.

Introduction

Selected Instruments

This paper focuses on the three instruments that I can play: ukulele, viola, and piano. The ukulele is a member of the lute family of instruments and is seen as a miniature guitar-like instrument. The viola is a string instrument that is slightly larger than a violin but with a deeper and more resonant sound. As the alto or 'middle' voice of the string family, its size and pitch is between the violin and the cello. The piano is a keyboard instrument typically with 36 black and 52 white keys, each of which produces sound when it is pressed. [1]

Table 1. The instruments selected for this study, along with the pitch of each string or key and the respective frequencies.

Ukulele		Viola		Piano	
String	**Frequency**	**String**	**Frequency**	**Key**	**Frequency**
G	392 Hz	C	130.4 Hz	A	27.50 Hz
C	262 Hz	G	195.6 Hz	B	30.50 Hz
E	330 Hz	D	293.3 Hz	…	…
A	440 Hz	A	440 Hz	C	4186.00 Hz

Timbre

There are many ways that have been proposed to tell apart musical instruments. Each instrument's unique tone is referred to as its timbre. [2] It comes from the sonic properties and characteristics of each instrument, based on how it is constructed and how it produces sound. Words to describe timbre include bright, dark, mellow, and warm. More specifically, warm tones are sounds that are soothing and comforting; bright tones are sounds that have a vibrant and sparkling quality.

Unique timbres can distinguish different instruments that are playing the exact same note. The same note means the same pitch and same fundamental frequency. But even when the fundamental frequency is the same, the harmonic frequency content is not; the vibrations and resonance varies across instruments. This is what leads to distinct timbres.

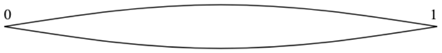
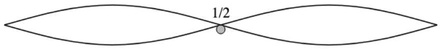
Table 2. Example of words used to describe timbre and instruments that are often described using those words. The three instruments used in this study (piano, viola, ukulele) are bolded.

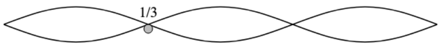
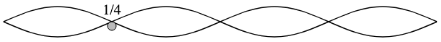
Timbre	Instruments
Brassy	Trumpet, Trombone
Harsh	Distorted electric guitar, **Loud piano**
Warm	**Viola**, Cello, Saxophone
Soft	Recorder, Quiet piano
Bright	**Ukulele**, Trumpet, Piccolo, Xylophone

Harmonic Frequencies

The timbre of the instrument can be quantified by its harmonic frequency content. Each instrument produces its own unique vibrational or standing wave pattern. These patterns are only created within the instrument at specific frequencies, called harmonic frequencies. The harmonic frequencies are multiples of the fundamental frequency, also known as the first harmonic frequency. For example, if the pitch A3 is played, it has a fundamental frequency of 220 Hz; the harmonics would be 440 Hz, 660 Hz, 880 Hz, and so on.

Table 3. Fundamental frequency (row 1) and harmonic frequencies (other rows) of a note.

Frequency	Order	Name 1	Name 2	Standing wave representation
$1 \times f = 440$ Hz	$n = 1$	1st harmonic	fundamental tone	
$2 \times f = 880$ Hz	$n = 2$	2nd harmonic	1st overtone	

3 × *f* = 1320 Hz	*n* = 3	3rd harmonic	2nd overtone	
4 × *f* = 1760 Hz	*n* = 4	4th harmonic	3rd overtone	

Every single note, when played on an instrument, has all of these different frequencies seamlessly overlapping together. For example, if the pitch A4 (440 Hz) was played on a piano, then the listener would mainly hear the fundamental frequency A4 (row 1), but also A5 (row 2), A6 (row 3), and higher octaves of a C.

Each instrument has harmonic frequencies that possess its own relative strength, resulting in differing harmonic frequency content. It was thus proposed that harmonic frequencies in spectrograms could be used to train the model to classify the various instruments.

Methods

Materials

The three instruments to be analyzed are the viola, ukulele, and piano.

For data collection, the three instruments were recorded playing two sets of pitches. The specific instruments used were a Donner Concert Ukulele, 15.5 inch Viola, and a Kawai Upright Piano. The recording was done with a RØDE VideoMicro Microphone and the Quicktime Player app.

The instruments were first tuned with a Korg TM-50 to ensure maximum accuracy and precision for pitch. Then, the following pitches were played and recorded on all three instruments.

Table 4. The datasets used in this study and their respective pitches.

Dataset	Number of Data Samples	Pitches
Training dataset	8 pitches per instrument	C4, D4, E4, F4, G4, A4, B4, C5
Testing dataset	5 pitches per instrument	C4, D4, E4, F4, G4

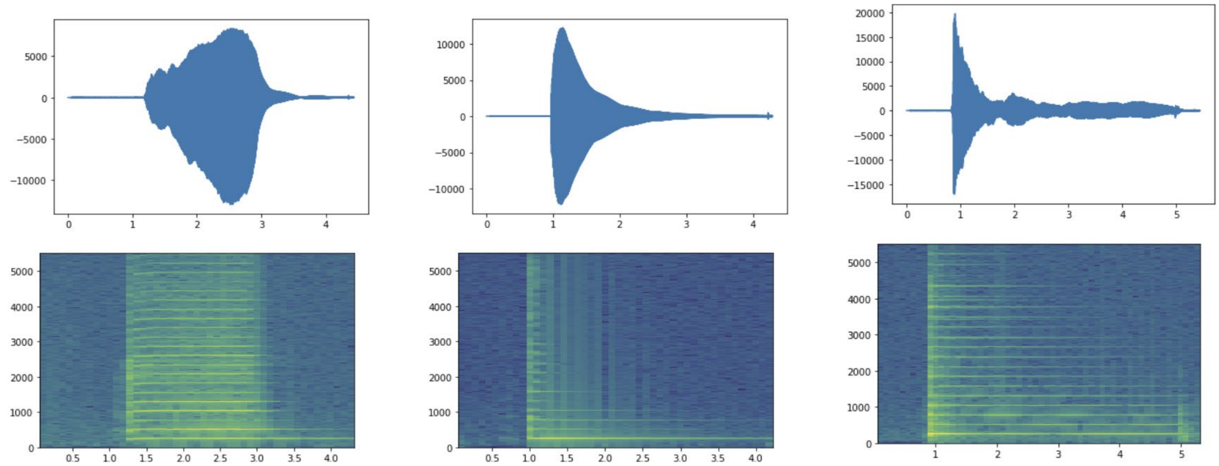
Two sets of audio recordings will ensure that the model is reliably recognizing all variations of what is essentially the same sound, even if there are subtle differences. Each recording was around five seconds long and contained the pure sound of the instrument, without any background noise. The MP3 files were converted into WAV files before being used for model training.

a) Instruments b) Microphone c) Tuner

Figure 1. a) The three instruments: viola (left), ukulele (right), and piano (center). b) The RØDE VideoMicro Microphone used to record. c) The Korg TM-50 tuner used to ensure a consistently accurate pitch for each recording.

First, analysis of spectrogram and waveform graphs for the audio recordings yielded insights to guide the development of the model. [3] The model assumed that the differences in harmonic frequencies was a primary cause of the distinct sounds of each instrument tested. A viola has a warm, ringing sound, with a high level of harmonic frequency content. A piano's sound is round and melodious, so it has a medium amount of harmonic frequencies. Plucking a ukulele string produces a short, clear sound, with little to no harmonic frequencies. It was hypothesized that the differences in the instruments' harmonic frequency content would be significant enough to distinguish them from each other.

Spectrogram Frequency Analysis



a) Viola b) Ukulele c) Piano

Figure 2. Waveform graphs (top) and spectrogram graphs (bottom) show clear differences in the harmonic frequencies. a) A viola has a warm, ringing sound, with a high level of harmonic frequency content (a lot more green than blue in the spectrogram graphs). b) Plucking a ukulele string produces a short, crisp sound, with little to no harmonic frequencies. c) A piano's sound is round and melodious, and a medium amount of harmonic frequencies.

Data Pre-Processing

During the computer's training stage, it split each audio recording into 5 frames and ran a Fast Fourier Transform (FFT) on each frame to convert a signal from its time domain to a representation in the frequency domain. [4] The FFT converts a signal from the time-domain to frequency-domain. The FFT divides the signal into its frequency components which are single sinusoidal oscillations at distinct frequencies each with their own amplitude.

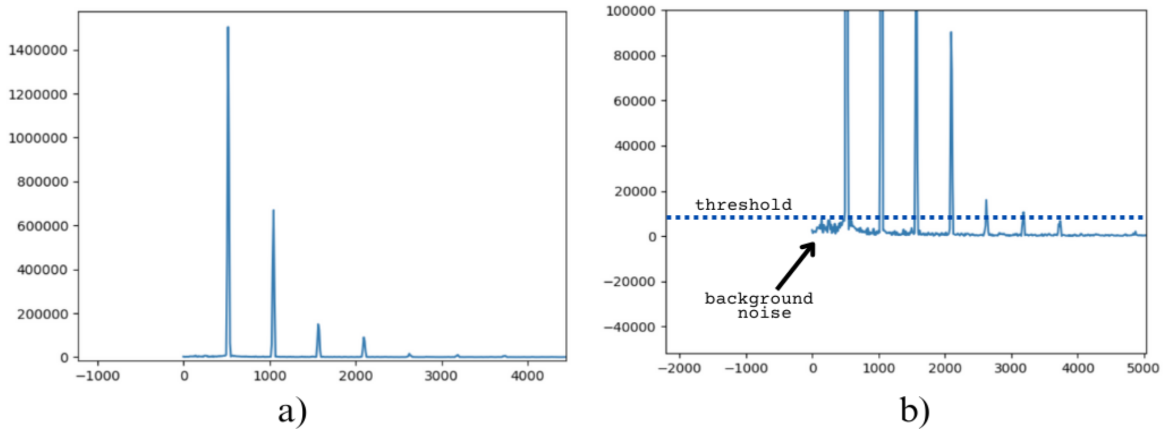


Figure 3. a) An FFT for an audio recording lasting approximately 5 seconds. The major peak with a high amplitude represents the fundamental frequency. b) A zoomed-in version of the same FFT shows a threshold line, where everything below it is considered background noise.

First, the computer found the amplitude of the fundamental frequency, the first peak that occurs above a certain threshold. There was a minimum threshold that the amplitude had to meet, in order to eliminate quiet background noise from being mistaken for the fundamental frequency. This was necessary because there was a period of silence of a few seconds before the instrument begins to play.

Next, the computer marked the amplitude of each harmonic frequency, located at every integer multiple of the fundamental frequency. Then the computer compared the ratio of these harmonic frequency amplitudes to the amplitude of the fundamental frequency. The more harmonic frequency content there was, the higher this ratio was, because later harmonic frequencies still had a relatively high amplitude instead of its sound quickly fading away. For example, viola had the highest ratio and the most harmonic frequency content, leading to a warm resonant sound that lasted multiple seconds longer than the piano and ukulele.

These results provided insights the features to be extracted for the algorithm. [5] The features extracted from each audio recording were the ratios of the amplitudes of the first ten harmonic frequencies relative to that of the fundamental frequency for each of five frames. These fifty features per audio recording were used to train the K-Nearest Neighbors algorithm.

K-Nearest Neighbors Algorithm

The K-Nearest Neighbors (KNN) algorithm is one of the simplest yet most effective machine learning algorithms. [6] KNN uses supervised learning, meaning it uses labeled training data. In this context, KNN is performing classification, where a class label is predicted based on plurality vote.

KNN is based on the idea that similar points can be found near one another, as they have similar labels and values. When making predictions, the algorithm calculates the distances between the input data point and all the training examples, using the Euclidean distance formula. [7]

Equation 1: Euclidean distance formula

$$d(p,q) = \sqrt{\sum_{i=1}^{n} \blacksquare (q_i - p_i)^2}$$

The algorithm identifies the 'k' (where k is an integer) nearest neighbors of an input point based on their distances. Ultimately, the algorithm predicts the label based on the most common class label among these k neighbors. The KNN algorithm can be considered a voting system because the class label of a new data point is predicted based on the k nearest neighbors in the feature space.

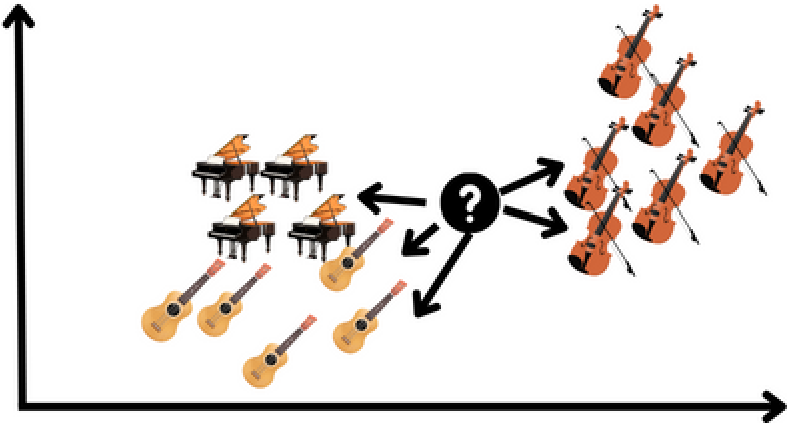


Figure 4. An example of the KNN algorithm trying to classify a new input data point. In this example, k=5 so the model seeks out the five closest neighbors in terms of distance. These include instruments from each class, making it a slightly tricky problem to solve.

Defining K Value

The k-value determines the number of neighbors that will be used to determine the classification of a given data point. It is important to optimize k for the specific dataset at hand to decrease overfitting or underfitting as much as possible. [8] The trade-off is between variance and bias, with low values of k offering low bias but high variance, and high values of k offering high bias and low variance. [9]

The value of k used to classify the instruments was k=5.

Advantages and Disadvantages

The KNN algorithm offers the advantages of ease of implementation and adaptation. Because KNN only requires a k value and a distance metric as the parameters, it has few hyperparameters compared to other machine learning algorithms. Furthermore, it adapts easily because all training data is stored in memory, so when new training samples are added, the algorithm adjusts without needing to start over.

The KNN algorithm also has some drawbacks. Specifically, it does not scale well and requires lots of memory and data storage compared to other classifiers. However, the instrument classification problem does not have a big enough dataset for this to be an issue. The KNN algorithm also does not perform well with high-dimensional data inputs, so providing too many features may actually increase the errors. This was controlled for in the instrument classification problem by tweaking the number of features until the optimal number was reached.

Model Training

First the model used the training dataset to train using the extracted features, and then tell the model to classify instruments. The parameters were fine tuned to optimize accuracy. Any changeable values were tweaked to try to push the accuracy up, including numFrame, numFreq, numSample, threshold=max(data)/2.

Next, the model was trained on the training dataset and tested on the testing dataset. Having a separate train and test set ensures the accuracy of the code's behavior in real world scenarios. The training data is used by the algorithm to identify a pattern in the training data, specifically the relationship between features and the class label. This identified pattern can be used to make predictions on new, unseen data, which is where the test set comes in. The test data is used to evaluate the performance of the model. The accuracy of the model's prediction is determined by comparing the model's predicted classes to the actual classes.

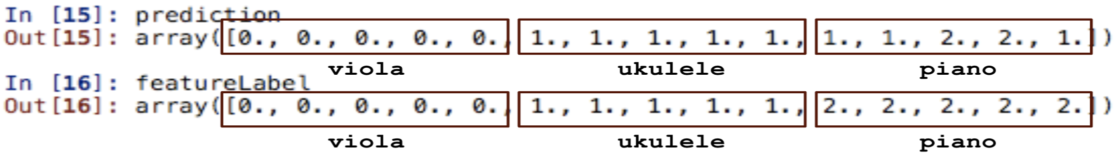
Results

Training Accuracy

After optimizing parameters (including numFrame, numFreq, numSample), the accuracy for the training dataset was 93%.

Testing Accuracy

After training on the data from the training dataset, the KNN model predicted the class of each audio recording in the testing dataset. The model predicted 12 out of 15 correctly which resulted in 80% accuracy.



```
In [15]: prediction
Out[15]: array([0., 0., 0., 0., 0., 1., 1., 1., 1., 1., 1., 1., 2., 2., 1.])

In [16]: featureLabel
Out[16]: array([0., 0., 0., 0., 0., 1., 1., 1., 1., 1., 2., 2., 2., 2., 2.])
```

Figure 5. The console showed the comparison of the computer's predicted label versus the actual label. (0 - Viola; 1 - Ukulele; 2 - Piano)

The confusion matrix is an accuracy metric that represents the prediction summary in matrix form, displaying both correctly predicted classes and errors. Each row represents the label of the true class, and each column represents the label of the predicted class. The diagonal cells show correctly classified samples, while the off-diagonal cells show model errors.

Table 5. The confusion matrix created to visualize the performance of the KNN Model.

		True Label		
		Viola	**Ukulele**	**Piano**
Predicted	**Viola**	5	0	0

Label	**Ukulele**	0	5	0
	Piano	0	3	2

First row: All 5 are correct for viola

Second row: All 5 are correct for ukulele

Third row: This is where the computer messed up. It thought that 3 out of the 5 piano recordings were ukulele recordings, and only got 2 out of 5 correct.

Principal Component Analysis

Principal component analysis (PCA) is a technique for reducing the dimensionality of datasets with lots of information. [10] For example, it can reduce a feature space with hundreds of features into only a few important ones. One of PCA's important features is its ability to increase interpretability but at the same time minimize information loss. [11] A PCA graph was generated to better visualize the differences between the three instruments to understand why the computer was making certain classification errors, specifically mixing up ukulele and piano.

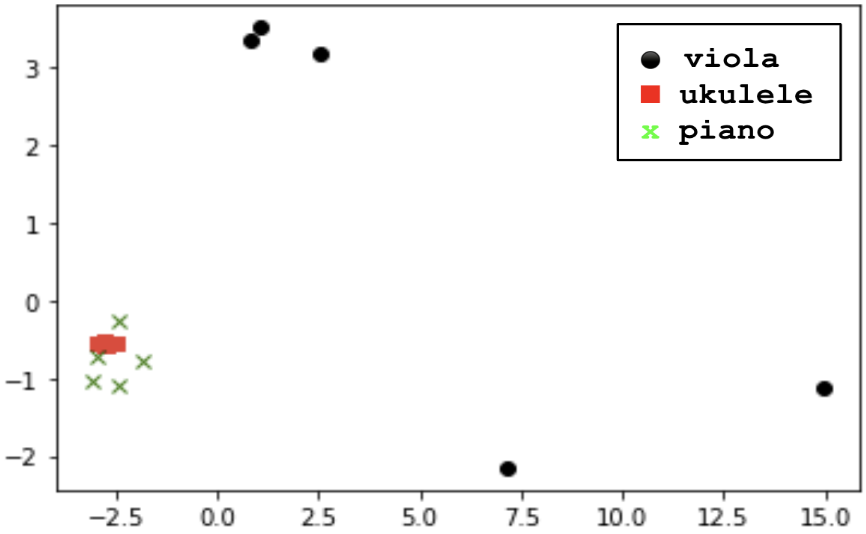


Figure 6. Principal Component Analysis graph**.** It demonstrates why the computer's accuracy for matching an audio recording with an instrument isn't perfect. Viola is very far away from piano and ukulele, so it's accuracy is 100%. However, a lot of the time, piano and ukulele get mixed up. As shown on the graph, the ■′s and x′s are super close together.

Discussion

Using models to classify different instruments is helpful to people who listen to music. The algorithm developed can be applied to the entire field of sound classification; AI can even surpass a human's ability to identify sounds. For example, in a home or industrial setting, an AI algorithm can accurately identify machines that aren't working based on the sounds they produce. Other applications of instrument classification are the automatic sorting and searching of massive musical collections with ease, which could be useful to streaming platforms that provide personalized recommendations to their users.

The data preprocessing quantified the difference in harmonic frequency patterns between viola, ukulele, and piano. The spectrograms clearly visualized the dramatic increase in harmonic frequency content in viola compared to ukulele and piano. This information guided the feature extraction process which compared the ratio of the amplitude of the first ten harmonic frequencies to the amplitude of the fundamental frequency. This resulted in a robust and reliable measure for the algorithm to separate different instruments.

This study also found a way to identify the fundamental frequency in an audio recording. Setting the highest peak to the fundamental frequency assumes that each harmonic frequency has a diminishing amplitude. However, some recordings had an unexpectedly large volume of harmonics, possibly due to echoes around the room; some harmonics even reached higher amplitudes than the original fundamental frequency. As a result, the algorithm falsely believed a harmonic frequency was the fundamental frequency.

To address this, the first peak, rather than the highest peak, was chosen to be the fundamental frequency. The problem with this approach was that many false peaks that occurred before the actual fundamental frequency. During the brief silence before a note was played on the instrument, tiny random vibrations distorted the silence. This was fixed by identifying the top peak overall (which could either be the fundamental or harmonic frequency, depending on the instrument), divided it by half, and setting that value to be a threshold. Then the algorithm calculated the first peak above that threshold to be the first fundamental frequency.

The model utilized the simple but effective K-Nearest Neighbors algorithm to reach an accuracy of 93% on the training dataset and 80% on the testing dataset. Specifically, the first iteration of the model trained the computer to separate instruments through analyzing the frequencies of training set of audio recordings, then tested its knowledge on the same set. The final iteration of the model trained the computer on the training set of audio recordings, and its performance was evaluated using the testing set of audio recordings.

This study shows the feasibility of using machine learning to understand and distinguish between the timbre of different instruments the way that humans do.

To further improve the accuracy rate of this engineering project, a larger and more diverse training dataset is required. With different musicians and settings, the real-world performance and accuracy will increase.

Conclusion

This paper presents a machine learning approach to classify three instruments: viola, ukulele, and piano. This was achieved through analyzing the harmonic frequency content of the respective instruments and using the information in a feature space. The model reached an accuracy of up to 93%.

This model allows anyone to figure out what instrument (piano, ukulele, viola) is being played in any audio recording of their choice. This can be applied to the broader world of sound classification, and machines can eventually surpass a human’s ability at identifying sounds. For example, it may be able to tell apart viola from violin, which the average person cannot do.

Limitations

The main limitations were that the audio recording had to be of one instrument and one pitch, instead of a string of various notes on an instrument, or many instruments playing together and overlapping. Only one instrument per pitch could be played because if there were multiple pitches being played, the frequencies would overlap and make it harder for clear patterns and relationships to be drawn by the algorithm.

Furthermore, there are limitations to the current model that need to be considered before being implemented into real world scenarios. Because the model was collected in perfect conditions—in a quiet room with minimal background disruptions—selection bias could have occurred since it is not representative of all deployment conditions.

Furthermore, if the fundamental frequency is incorrectly classified, it throws off the rest of the algorithm because the algorithm relies on the fundamental frequency to compute its integer multiples and find each harmonic frequency. For example, during the initial development of the model, consistent errors with detecting the fundamental frequency resulted in accuracies as low as 50%. These errors were solved by setting a minimum amplitude threshold and detecting the first peak that occurred above that threshold.

Acknowledgments

I would like to thank my family and my teachers for always encouraging my passion for science and engineering.

References

Parncutt, R., et al. “Review of the Science and Psychology of Music Performance: Creative Strategies for Teaching and Learning.” Bulletin of the Council for Research in Music Education, no. 160, 2004, pp. 76–86, www.jstor.org/stable/40319221.

Saldanha, E. L., and John F. Corso. “Timbre Cues and the Identification of Musical Instruments.” The Journal of the Acoustical Society of America, vol. 36, no. 11, Nov. 1964, pp. 2021–2026, https://doi.org/10.1121/1.1919317.

“On the Relevance of Spectral Features for Instrument Classification | IEEE Conference Publication | IEEE Xplore.” Ieeexplore.ieee.org, ieeexplore.ieee.org/abstract/document/4217451.

Weisstein, Eric W. “Fast Fourier Transform.” Mathworld.wolfram.com, mathworld.wolfram.com/FastFourierTransform.html.

“Comparison of Features for Musical Instrument Recognition | IEEE Conference Publication | IEEE Xplore.” Ieeexplore.ieee.org, ieeexplore.ieee.org/abstract/document/969532.

Kramer, Oliver. “K-Nearest Neighbors.” Dimensionality Reduction with Unsupervised Nearest Neighbors, vol. 51, 2013, pp. 13–23, https://doi.org/10.1007/978-3-642-38652-7_2.

Chérifa Boucetta, et al. “Improved Euclidean Distance in the K Nearest Neighbors Method.” Communications in Computer and Information Science, 1 Jan. 2023, pp. 315–324, https://doi.org/10.1007/978-3-031-40852-6_17.

Wang, Jigang, et al. “Neighborhood Size Selection in the K-Nearest-Neighbor Rule Using Statistical Confidence.” Pattern Recognition, vol. 39, no. 3, Mar. 2006, pp. 417–423, https://doi.org/10.1016/j.patcog.2005.08.009.

Anava, Oren, and Kfir Levy. “K\Ast -Nearest Neighbors: From Global to Local.” Neural Information Processing Systems, Curran Associates, Inc., 2016, proceedings.neurips.cc/paper/2016/hash/2c6ae45a3e88aee548c0714fad7f8269-Abstract.html.

Shlens, Jonathon. “A Tutorial on Principal Component Analysis.” ArXiv:1404.1100 [Cs, Stat], 3 Apr. 2014, arxiv.org/abs/1404.1100.

Chipman, Hugh A., and Hong Gu. “Interpretable Dimension Reduction.” Journal of Applied Statistics, vol. 32, no. 9, Nov. 2005, pp. 969–987, https://doi.org/10.1080/02664760500168648.