Piano Recognition - The Creation of a Music Recognizer Program

Travis Osgood

Brendon Williams

Caleb Hart

Joshua Aguayo

Parker Pratt

Abstract

This paper details the development of a Piano Recognition program designed to identify and classify piano music amidst background noise and various audio distortions. Building upon existing methods such as Wiener filtering, spectral subtraction, our approach leverages advanced algorithms and techniques, including Dynamic Time Warping (DTW), and multi-pitch analysis.

Our project utilized the MusicNet dataset combined with additional piano recordings from sources like YouTube to simulate real-world conditions. We intentionally introduced Gaussian noise and ambient sounds to enhance model robustness. Feature extraction involved analyzing waveform representations, spectrograms, Mel-frequency cepstral coefficients (MFCCs), chromograms, and tempograms. DTW was important in aligning audio sequences for accurate comparison.

Prior Work

Prior work in this field has established the use of noise reduction algorithms, like Wiener filtering, wavelet transform, spectral subtraction, and improved spectral subtraction. However, with these algorithms, there has been a lack of professional data sets in music noise reduction [5]. Other works have established mathematical principles of music analysis, like Markov note recognition algorithms [2] as well as Fast Fourier Transform [0][6] for decomposition and extraction. As mentioned in “Unsupervised note activity detection in NMF-based automatic transcription of piano music” [1], Hidden Markov Models show much promise in problems like ours and are already fairly widely

used. This is because they extrapolate results, which usually yields better results than thresholding, which was the method used in that paper.

Other methods used in prior works include Convolutional Neural Networks. In 2021, this was used to identify whether the noise provided was from a piano or not [3]. This was very interesting for us to read and was something to consider, but at the moment, we do not have the hardware to run a model similar to this one.

Our work was trained and tested on synthesized and real-world examples of piano music similar to the hierarchical model used for Robust Real-Time Music Transcription in 2017 [9]. This was a suggestion for future work provided by a group at a conference in 2005 [12]. By doing it this way, we were able to have a more robust model that could determine what song was being played even with noise in the background.

With the authors' emphasis on multi-pitch detection and source separation [4], we initially looked at the maestro dataset [11], which gave us an indication of the type of data we needed and gave us the foundation that led us to the dataset titled “Musicnet” [8]. We used Musicnet to obtain piano pieces with note annotations and supplemented them with songs scraped from YouTube, such as Hot Cross Buns [10]. This introduced distortions such as slight pitch shifts, amplitude variations, and random note substitutions. Using algorithms and techniques found in a journal [7], we added Gaussian noise to the data as well as ambient sounds so that we could mimic real-world complications. Finally, to determine which cleaned-up track most closely resembled a

user-submitted recording, we employed a Hamming distance measure across the denoised signal because of the success it had from the 2005 conference [12]. We used the Hamming distance measure to identify which of the denoised recordings were closely related using non-negative matrix factorization and multi-pitch analysis for handling overlapping notes.

Visualizations/Analysis

Data Source: Musicnet Random Sampling [8].

Data Acquisition

The dataset used in this project is sourced from Kaggle and is accessed by mounting Google Drive to facilitate storage and retrieval. The Kaggle API is employed to download the MusicNet dataset, which consists of labeled musical recordings. This dataset contains multiple .wav files, each representing a musical piece that will be analyzed. The songs were pulled from youtube.

Data Processing

Once the .wav files are identified and stored in a structured manner. The librosa library that was mentioned in a journal that we read [7] is used for loading and processing these audio files. These .wav audio files are what are used to train our model. Several key features are extracted, including waveform representation, spectrograms, and Mel-frequency cepstral coefficients (MFCCs).

In addition to feature extraction, Dynamic Time Warping (DTW) is applied to compare different audio sequences. This method is effective in aligning sequences that may vary in tempo but share similar melodic structures, making it particularly useful for comparing identical songs that have experienced some sort of distortion.

Example output:

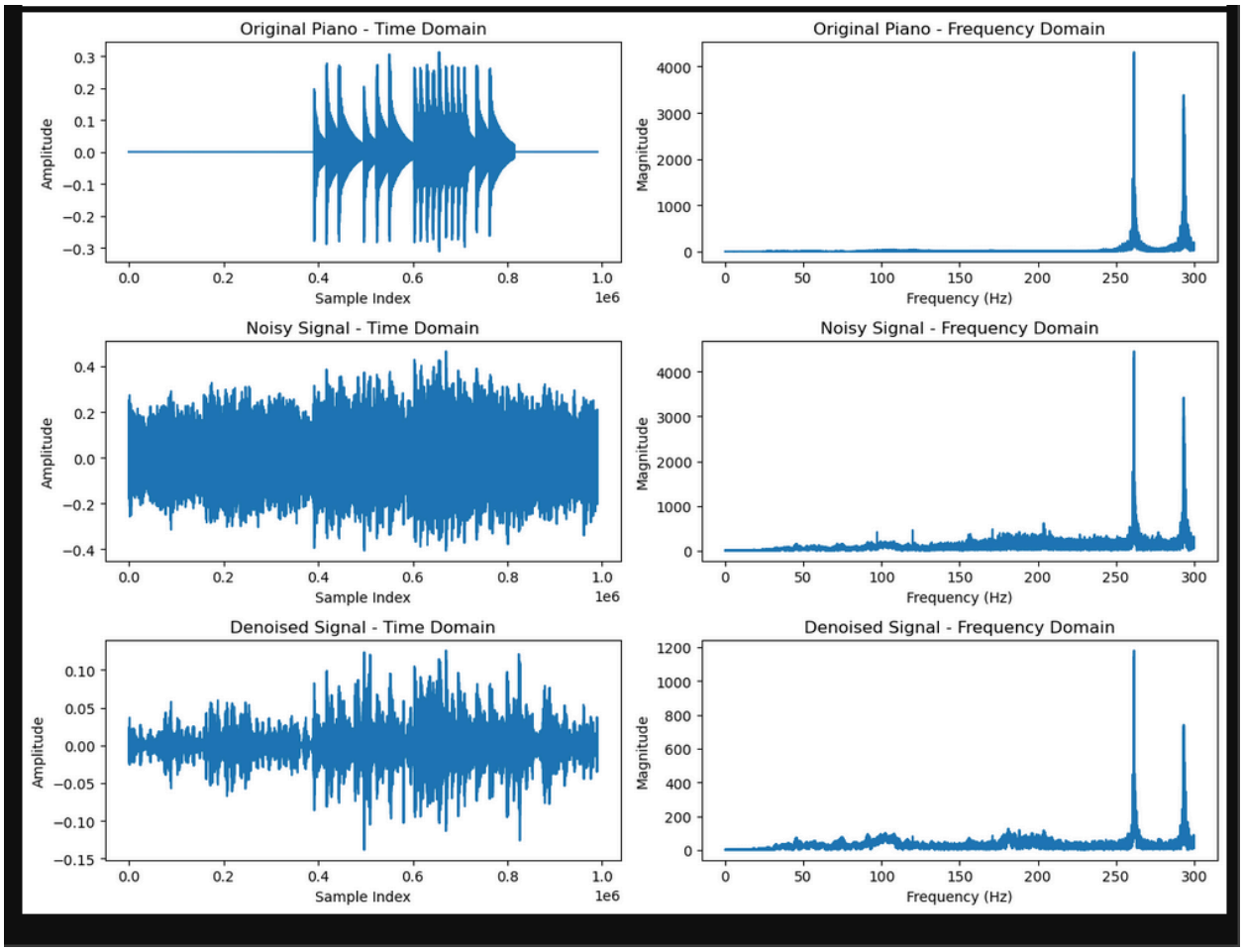
```
# print top N matches
top_n = 5
print("\nTop {} matches:".format(top_n))
for i in range(top_n):
    print(f"{i+1}. {distances[i][0]} (distance = {distances[i][1]:.3f})")
```

```
Top 5 matches:
1. /content/drive/MyDrive/HotCrossBuns.wav (distance = 25.759)
2. /content/drive/MyDrive/Kaggle/musicnet-dataset/musicnet/musicnet/train_data/2158.wav (distance = 47.984)
3. /content/drive/MyDrive/Kaggle/musicnet-dataset/musicnet/musicnet/train_data/2282.wav (distance = 48.245)
4. /content/drive/MyDrive/Kaggle/musicnet-dataset/musicnet/musicnet/train_data/1931.wav (distance = 48.345)
5. /content/drive/MyDrive/Kaggle/musicnet-dataset/musicnet/musicnet/train_data/2315.wav (distance = 48.989)
```

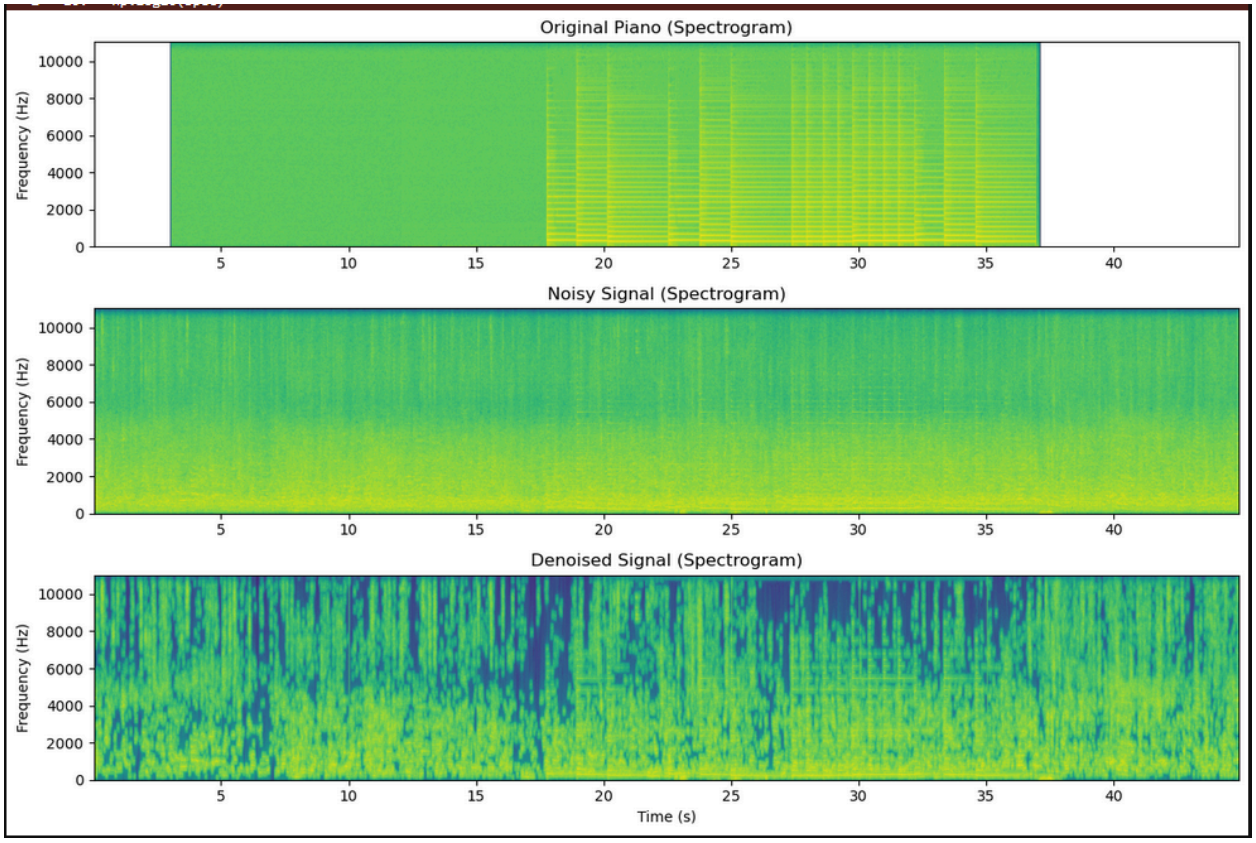
Visualization methods

Waveform: This function directly plots the amplitude of the audio signal as a function of time. The waveform represents the raw audio signal, showing the variations in air

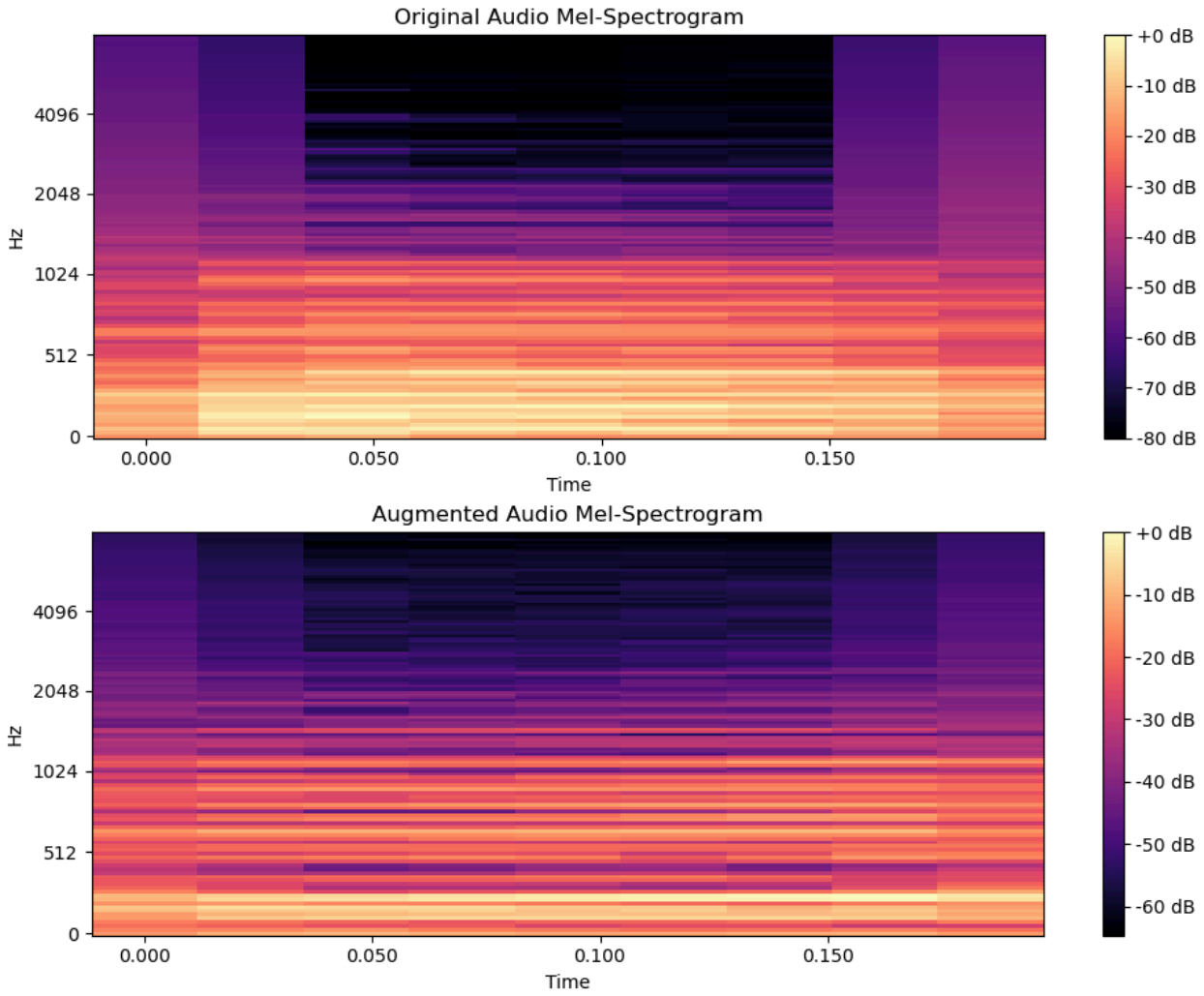
pressure over time. It's a fundamental representation of sound.



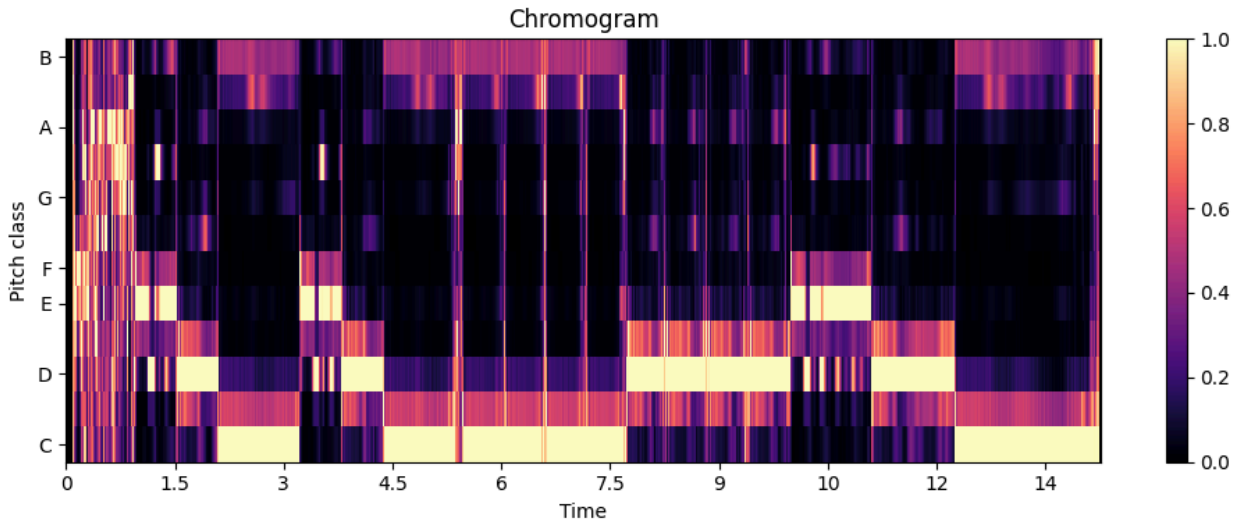
Spectrogram: Convert amplitude to Dbs for human readability. The Y-axis is on a log scale. We are using librosa.stft's built-in methods, which computes the Short-Time Fourier Transform (STFT) of the audio signal. STFT breaks the signal into short time windows and calculates the frequency content within each window. Librosa.amplitude_to_db converts the STFT magnitudes to decibels (dB), a logarithmic scale that better represents human perception of loudness. A spectrogram shows how the frequency content of a signal changes over time. It reveals the distribution of energy across different frequencies.



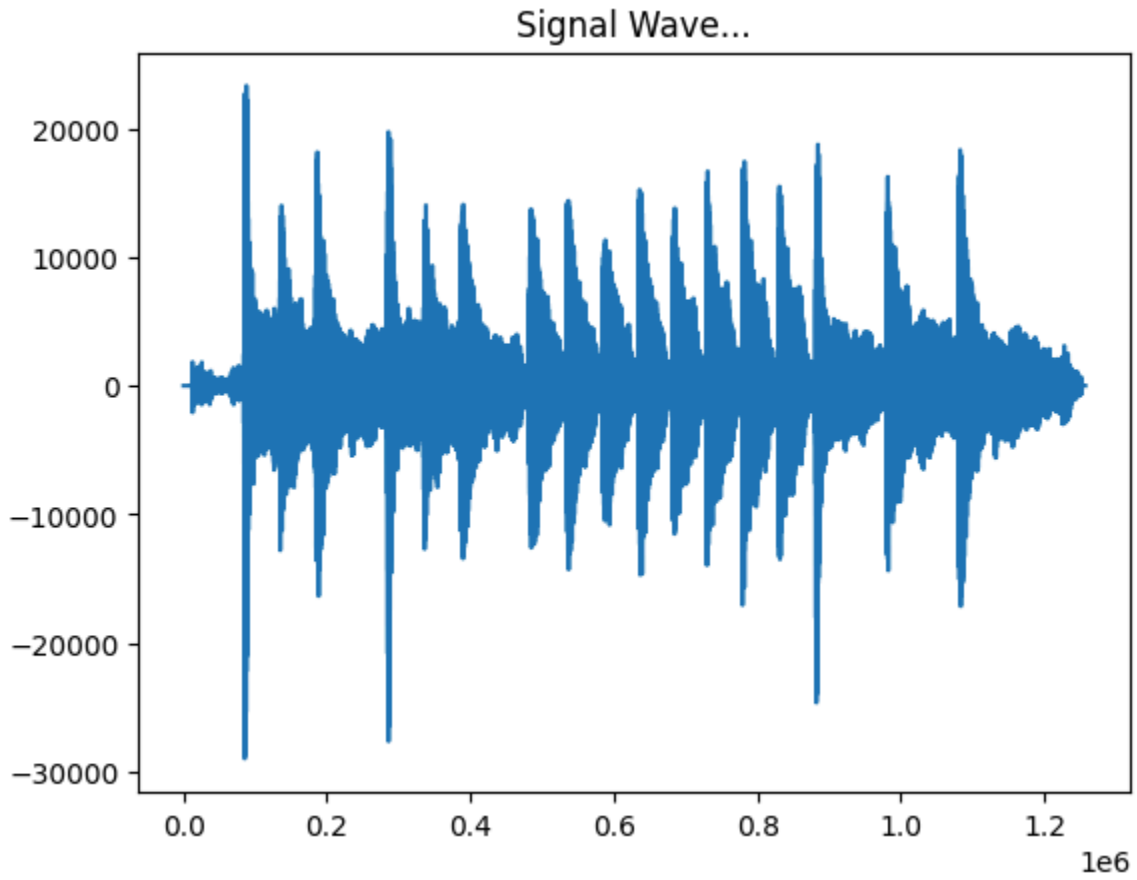
MFCC Visualization: This function computes the Mel-Frequency Cepstral Coefficients (MFCCs) of the audio signal. MFCCs are a compact representation of the spectral envelope, emphasizing perceptually important features. MFCCs capture the timbre of the sound, which is important for distinguishing between different instruments or voices.



Chromogram: We are using librosa.feature.chroma_stft: This function computes a chromagram from the raw audio signal. A chromagram represents the tonal content of the audio, mapping it to 12 pitch classes (C, C#, D, etc.). Chromagrams are useful for analyzing the harmonic structure of music and identifying notes as well as changes in key. Converts audio pitch to music scales for human readability. The X-axis represents time, and the Y-axis represents scale.



Signal Wave: Visualizes the frequency of audio for human readability. The Y-axis represents the pitch in hertz.



Tempogram: We are using librosa.onset.onset_strength: This function estimates the strength of onsets (sudden increases in energy) in the audio signal. Librosa.feature.tempogram computes a tempogram, which represents the prevalence of different tempos over time. Tempograms are used to analyze the rhythmic structure of music and identify tempo changes.

Visualization of analysis: Displaying short-time Fourier transforms allows us to compare the intensity and distribution of frequency components in the noise signal against the clean signal. Through this, some peaks are smeared or broad in noise and can be overlaid with the clean data, showing how the characteristics alter both amplitude and frequency profiles. Time-domain waveforms can be overlaid to reveal differences in transient attacks or decay characteristics. By integrating visual cues with analytical measures such as signal to distortion ratio or hamming distance, as well as researcher and engineer denoising algorithms, we can create visuals that reveal useful insights.

Feature Comparison

A significant aspect of this analysis involves comparing extracted features across different musical recordings. Dynamic Time Warping (DTW) is used as a primary method to measure the similarity between different audio sequences. Since musical recordings may vary in tempo and timing, DTW helps align these sequences in a way that highlights their similarities. By applying DTW, it is possible to determine how closely two musical pieces resemble each other despite variations in tempo and minor

differences in execution. Once features have been extracted, they are then saved using pickle files for future computation and analysis.

Classification and Identification

Once we have the distances between the different audio files, the song with the shortest distance is classified as the song that we are trying to identify.

Future Development

Future development of Piano Recognition would include recognition of more varied common piano tracks and further distancing the distinction between the sounds from a piano and general environmental ambiance.

Future projects involve refining pitch estimation algorithms to better handle expressive elements such as vibrato. Past pitch tracking, these networks could embed traditional musical knowledge. Understanding harmonic progression rules aiming to achieve marginal improvements. Finding the sweet spot that balances computation and real-world complexity. Noise reduction metrics, precision/recall for multi-pitch detection. Precisely capturing or simplifying subtle hinge shifts and calibration of instruments.

Individual Contributions

Travis Osgood - Co-author, Co-developer

Implemented Chromogram visualization for determining pitch and note class and Signal Wave visualization for hertz display, co-author for project write-up. Co-author of section prior work.

Brendon Williams - Lead Developer

Merged the teams' coding files into one large file, fixed memory issues with the scale of the project, and was co-author for project write-up.

Caleb Hart - Lead Author

Developed and implemented mel-spectrogram, autocorrelation, tempogram, colored noise generation research and development. Author of Visualizations/Analysis section. Co-author of section prior work.

Joshua Aguayo - Lead Researcher

Created skeleton draft of code, visualisation of Fourier transform, spectrogram, and Mel-Freq Cepstral Coefficient, Co-author of section prior work. Author of abstract section.

Parker Pratt - Lead Researcher

Created waveform plot that measures distances between all predictions, assisted with other visualizations, co-authored analysis code, researched methods to be used, co-author for project write-up.

References

[0] Bank, B., Avanzini, F., Borin, G., De Poli, G., Fontana, F., & Rocchesso, D. (2003). *Physically informed signal processing methods for piano sound synthesis: A research overview*. EURASIP Journal on Applied Signal Processing, 2003(10), 941–952. https://home.mit.bme.hu/~bank/publist/jasp03.pdf

[1] Fernandes Tavares, T., Garcia Arnal Barbedo, J., & Attux, R. (2016). Unsupervised note activity detection in NMF-based automatic transcription of piano music. *Journal of New Music Research*, *45*(2), 118–123. https://doi-org.oregontech.idm.oclc.org/10.1080/09298215.2016.1177552

[2] Fu, T. (2022). [retracted] model of markov-based piano note recognition algorithm and Piano Teaching Model Construction. Journal of Environmental and Public Health, 2022(1). https://doi.org/10.1155/2022/6045597

[3] Girbés Mínguez, J. (2021). *Piano Note Recognition: Classification Aided by Convolutional Neural Networks* [Review of *Piano Note Recognition: Classification Aided by Convolutional Neural Networks*]. https://riunet.upv.es/bitstream/handle/10251/182214/Girbes%20-%20Reconocimiento%20de%20Notas%20de%20Piano%20Clasificacion%20Usando%20una%20Red%20Neuronal%20Convolucional.pdf?sequence=1&isAllowed=y

[4] Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.-Z. A., Dieleman, S., Elsen, E., Engel, J., & Eck, D. (2019, January 17). *Enabling factorized piano music modeling and generation with the maestro dataset*. arXiv.org. [1810.12247] Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset

[5] Ma, S. (2023). A comparative study of multiple music signal noise reduction algorithms in music processing. Noise & Vibration Worldwide, 54(6), 291–296. https://doi.org/10.1177/09574565231179732

[6] Müller, M., Ellis, D. P. W., Klapuri, A., & Richard, G. (2011). *Signal processing for music analysis*. IEEE Journal of Selected Topics in Signal Processing, 5(6), 1083–1096. https://www.ee.columbia.edu/~dpwe/pubs/MuEKR11-spmus.pdf

[7] Müller, M., & Zalkow, F. (2021). libfmp: A Python Package for Fundamentals of Music Processing. *Journal of Open Source Software*, *6*(63), 3326. https://doi.org/10.21105/joss.03326

[8] *Papers with Code - MusicNet Dataset*. (n.d.). Paperswithcode.com. https://paperswithcode.com/dataset/musicnet

[9] Pesek, M., Leonardis, A., & Marolt, M. (2017). Robust Real-Time Music Transcription with a Compositional Hierarchical Model. *PLoS ONE*, *12*(1), 1–21. https://doi-org.oregontech.idm.oclc.org/10.1371/journal.pone.0169411

[10] SingToMyPiano. (2020, March 16). *Hot Cross Buns Piano Tutorial / Karaoke*. YouTube. https://www.youtube.com/watch?v=IINrLbigEdU

[11] *The MAESTRO Dataset*. (2018, October 29). Magenta. https://magenta.tensorflow.org/datasets/maestro#dataset

[12] Yan Ke, D. Hoiem and R. Sukthankar (2005). "Computer vision for music identification," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Computer vision for music identification | IEEE Conference Publication | IEEE Xplore