Polyphonic Melody Extraction and Annotation Tool

A PROJECT REPORT

SUBMITTED ON COMPLETION OF THE PROJECT

’MELODY EXTRACTION AND ANNOTATION TOOL’

COURSE EE392A (UGP-II)

Under the guidance of

Prof. Vipul Arora

Submitted by:

Saransh Shivhare (200881)

DEPT. OF ELECTRICAL ENGINEERING

IIT KANPUR (U.P)

JAN 2023-APRIL 2023

CERTIFICATE

This is to certify that the project entitled “Polyphonic Melody Extraction and Annotation Tool” submitted by Saransh Shivhare (200881) as a part of Under Graduate Project-II offered by the Department of Electrical Engineering at Indian Institute of Technology, Kanpur, is an authentic record of the work done by him under my guidance and supervision at the Indian Institute of Technology, Kanpur during 2022-23 even semester.

Dr. Vipul Arora
Associate Professor
Department of Electrical Engineering
Indian Institute of Technology Kanpur

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Professor Vipul Arora for allowing me to undertake this project and for his constant guidance and support throughout the project. His valuable feedback, insightful suggestions, and constructive criticism were instrumental in shaping my work and helped me to achieve my goals.

I would also like to thank Department of Electrical Engineering, IIT Kanpur for providing me with the necessary resources and facilities to carry out this project. I would also thank Mrs. Kavya Saxena for guiding me throughout the project.

I would like to extend my heartfelt thanks to my friends and family for their unwavering support, encouragement, and motivation during this project.

Finally, I would like to express my appreciation to all the participants who willingly agreed to take part in this project and shared their valuable insights and experiences with me.

Saransh Shivhare

Contents

1. INTRODUCTION

1.1 Overview

The goal of this project is to create an interface for the task of polyphonic melody extraction and annotation. The tool is useful for creating a dataset for machine learning models related to music. Building an interactive web platform to modify melodic pitch values predicted by the machine learning model is required. The web application makes use of the ”Dash” Python package. Additionally, a javascript library called ”chart.js” adds interactivity to the plots that are shown in the interface. To comprehend and apply domain adaptation approaches to the model, I reviewed a wide range of publications and other works.

2. Literature Survey

2.1 Melody Extraction

2.1.1 About

Melody extraction is the process of extracting pitch of the dominant singing voice from polyphonic audio. There are several applications of melody extraction, which include

- **Music transcription:** Melody extraction is a critical step in music transcription, which involves converting an audio recording into a written score or sheet music. Transcribing music by hand is a time-consuming and error-prone task, and melody extraction algorithms can greatly simplify the process and improve its accuracy.
- **Music analysis:** Melody extraction can be used to analyze the structure and characteristics of a piece of music. By isolating the melody line, researchers can study the melody's rhythm, pitch, and other features in isolation. This can help to identify patterns and relationships between different melodies and musical styles.
- **Music recommendation systems:** Melody extraction can be used to analyze a user's listening habits and recommend new music based on their preferences. By analyzing the melody line of the user's favorite songs, recommendation systems can identify other songs with similar melodic features and suggest them to the user.
- **Music copyright protection:** Melody extraction can be used to identify cases of music plagiarism or copyright infringement. By comparing the melodic features of two songs, researchers can determine if one song was copied from another.

2.1.2 Existing Methods for Polyphonic Melody Extraction

Some of the available approaches for melody extraction is summarized below:
Multicolumn Deep Neural Network [1] is a classification-based approach for melody extraction on vocal segments. In the proposed model, each of the neural networks is trained to predict a pitch label of a singing voice from a spectrogram, but their outputs have different pitch resolutions. The final melody contour is inferred by combining the outputs of the networks and post-processing it with a Hidden Markov Model. Hidden

Markov Models (HMMs) are a powerful tool for modelling sequential data and have many practical applications in machine learning and data analysis.

In paper [2], the author proposes a novel frequency-temporal attention network to mimic the human auditory assigning different weights in the time and frequency axis. A selective fusion module is proposed to dynamically assign weight to the spectral features and temporal features. Then these features are fused for melody extraction.

Another existing approach is of Patch based CNN [4]. It is inspired by object detection in image processing. The input of the model is a novel time-frequency representation which enhances the pitch contours and suppresses the harmonic components of a signal. The proposed system uses a similar strategy to R-CNN. It extracts a CFP representation from a signal, selects patches as candidates of vocal melody objects in the representation, trains a CNN to determine whether a patch corresponds to a singing voice or not, and then localizes a voice melody object both in time and frequency. The Continuous Frequency Pitch (CFP) representation of an audio signal can be defined in terms of its corresponding harmonic representation using the Generalized Cepstrum (GC) and Generalized Cepstrum of Cepstrum (GCoS).

The GC is a mathematical technique that transforms the spectrum of an audio signal into a new domain called the cepstral domain. In the cepstral domain, the harmonic structure of the signal becomes more apparent, and it becomes easier to identify the pitch and other harmonic characteristics.

The GCoS is a further extension of the GC that is used to obtain the CFP representation of an audio signal. It involves taking the cepstrum of the GC of the original signal, which emphasizes the periodicity of the signal and allows the CFP to be extracted as a sequence of peaks in the cepstral domain.

2.2 Domain Adaptation

2.2.1 About

Domain adaptation is a technique used in machine learning to transfer knowledge from a source domain to a target domain, where the source and target domains have different distributions. In other words, it involves adapting a model trained on one set of data to perform well on another set of data that may have slightly different characteristics. This is useful in situations where collecting labeled data for the target domain may be expensive or time-consuming.

2.2.2 Existing Methods for Domain Adaptation

Domain adaptation is a technique in machine learning that involves adapting a model trained on one domain to perform well on another domain. In the context of melody extraction, domain adaptation can be used to improve the performance of a model trained on one type of music (e.g., classical music) to perform well on a different type of music (e.g., pop music). Some methods proposed for domain adaptation are:

- **Domain Adaptation by Fine Tuning:** One common approach for domain adaptation in melody extraction is to use transfer learning, which involves fine-tuning a pre-trained model using a smaller set target domain data. For example, a pre-trained model on a dataset of Western pop music may be fine-tuned on a dataset of Indian classical music to adapt to the different melodic patterns and scales.

- **Meta Learning based Domain Adaptation:** A method proposed in Deep Domain Adaptation for Polyphonic Melody Extraction [6] is based on adapting the model using some annotated data in the target domain. According to the experiments, meta-learning-based adaptation performs better than simple fine-tuning. Meta-learning-based technique, i.e. MAML(Model Agnostic Machine-Learning), focuses on learning suitable initialization parameters for a model trained on the source domain to adapt to the target domain with little training data quickly.

- **An approach from the image domain for Domain Adaptation:**
  In [7] author proposes a GAN-based model architecture for the Domain Adaptation task in images. The core idea of GANs is an adversarial game between two networks, called discriminator and generator, in an adversarial manner.
  The discriminator network is a binary classifier that tells if data is real (sampled from the true distribution) or synthetic (generated by the generator network). The generator network is tasked with generating data that resembles the real data.
  During the training phase, the generator network modifies its parameters based on

a signal (loss) from the discriminator so that the data generated at the current iteration is closer to the real samples than the previous iterations. Ideally, at the end of the training, the discriminator cannot discriminate between real and synthetic data.

Overall, domain adaptation techniques have the potential to improve the accuracy and robustness of melody extraction models in real-world scenarios, where the source and target domains may have significant differences. There are some approaches available in image domain which can be imported into audio domain for domain adaptation task. Like, GAN-based domain adaptation technique is used in the image domain widely. This technique might give promising results for domain adaptation in the field of melody extraction.

3. ABOUT TOOL

3.1 Use of External Libraries

Dash is a Python library that is used by the web application. It is well-known for the many different interactive applications. Also it is a package for the Python programming language that allows developers to create interactive web apps. It offers a straightforward and time-saving method for the development of interactive dashboards, data visualisation tools, and online apps.

In order to further enhance the application's capacity for user interaction, the javascript library known as chart.js has also been included in its deployment. Chart.js is a JavaScript toolkit that allows users to create customizable interactive charts and graphs on web pages. It is open-source, completely free, and offers a straightforward and adaptable method for the creation of a wide variety of chart kinds, such as line, bar, pie, radar, and more.

The application also makes heavy use of the Librosa library, which is primarily responsible for handling audio data. Flask, Scipy, Plotly, Soundfile, and Pydub are some of the other libraries that are utilised by the program.

3.2 Working of the Tool

From the corpus of the audio data of different duration the tool allows users to select audio of their desired length. Upon selection, the audio is divided into 5 second chunks and are available in a dropdown. Now, the user can choose a desired audio part from the dropdown. The first step is to generate a pitch contour. Users can create a melody plot by clicking the 'Get Melody Plot' button. The melody extraction model is used to estimate the pitch contour. The plot is overlapped with the spectrogram. Moreover, the tool highlights the points with low confidence values in 'red' frames. The user can magnify the plot accordingly.

After annotating data points, the user can click the 'Retrain-Model' button at the top. A meta-learning-based model updates the model's weight according to annotations made by the user, improving the model's performance. Now the user can regenerate the pitch contour using the 'Get Melody Plot' button, which will update the pitch values per the modifications in the model's weight. The plot also has highlighted blue frames representing the earlier annotations made by the user. The user can download the pitch contour data using the 'Download CSV' button. A CSV file with the same name as the annotated audio file will download.

3.2.1 Visual Error Detection

One of the methods by which one can mark the data point to its correct position is by visually recognizing inaccuracies through the use of the pitch contour plot and spectrogram. The pitch contour has to cross over into the fundamental frequency (f0) zone, the fundamental frequency band located at the very bottom of the spectrogram image. If this does not occur, the user will be presented with an interactive plot marker. These indicators are movable in the vertical direction and can be dragged. Using this, the user can drag them to the appropriate place and then save the modification by clicking the "Save" button, accessible directly below the tab in question.

3.2.2 Auditory Error Detection

Another method is to listen to the generated pitch contour to spot a mistake in the extracted tune. However, the user needs to be a music professional in this particular instance. Within the 'Auditory Error Detection' tab, the user can listen to sinusoid audio of the frequency that has just been recently changed. Users can also listen to the audio as it was originally recorded to locate the problem. The generated wave plot is interactive, and users can select different parts of the original audio by selecting a range of the plot. A "red" line is displayed on the plot to indicate the time frame with recently modified frequency. Once the user has finished analyzing the audio in this section, the user can update the data point in the "Visual Error Detection" tab according to their findings.
We are currently working on overlapping the '.wav' file generated for the pitch contour with the '.wav' file created for the original audio. When this occurs, audio error detection might become simpler for a non-expert as well.

3.3 Quick review of the tool

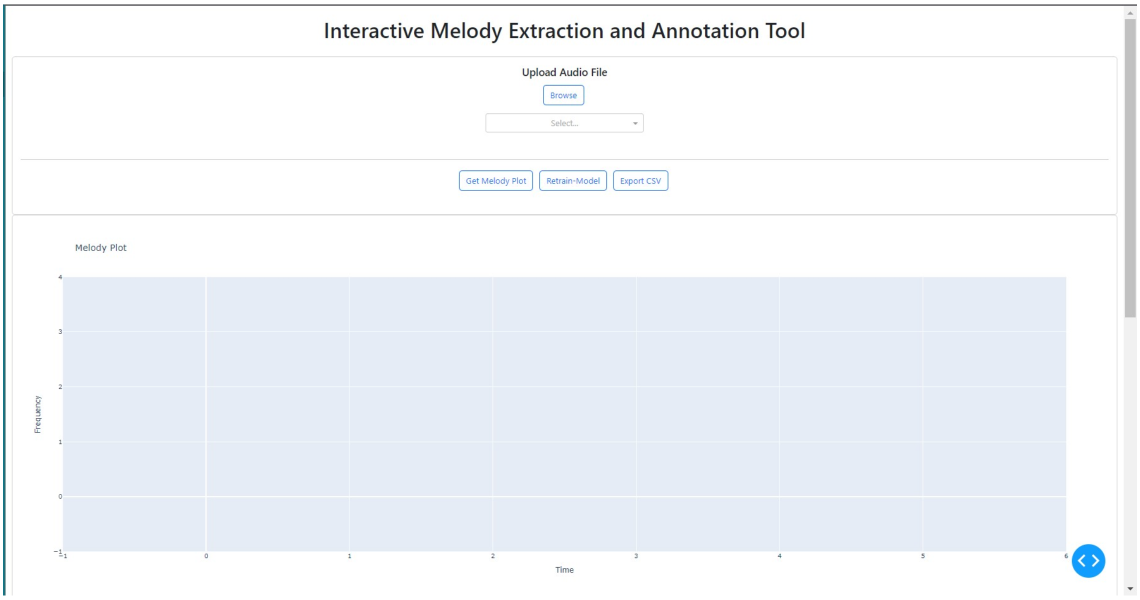
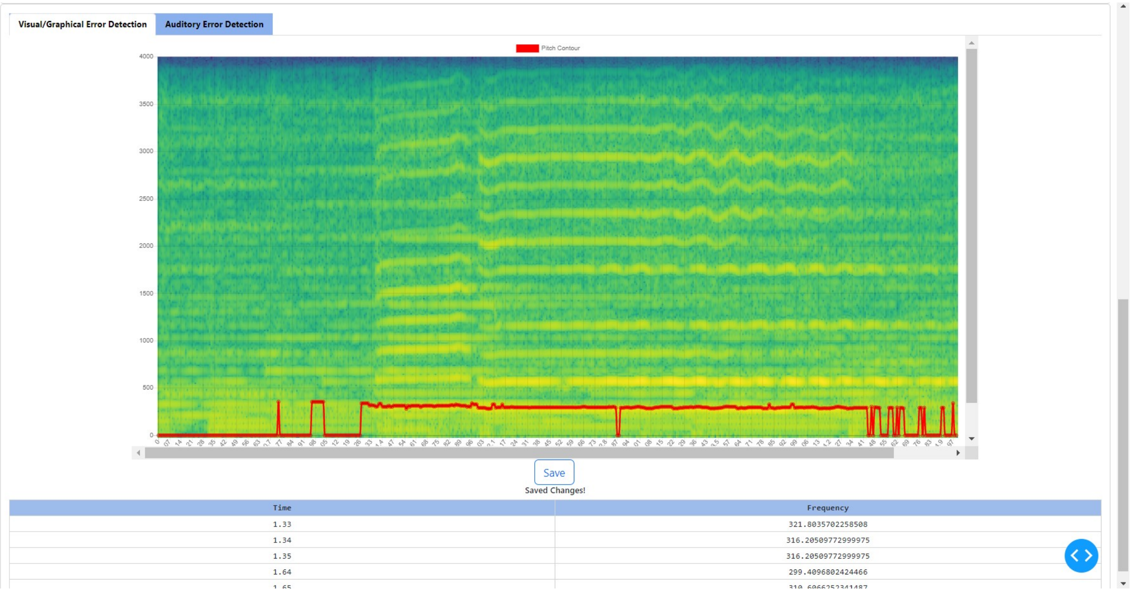


Figure 3.1: Initial encounter with the application's web page



Saved Changes!

Time	Frequency
1.33	321.8035702258508
1.34	316.20509772999975
1.35	316.20509772999975
1.64	299.4096802424466
1.65	310.6066252341487

Figure 3.2: Tab for Visual Error Detection

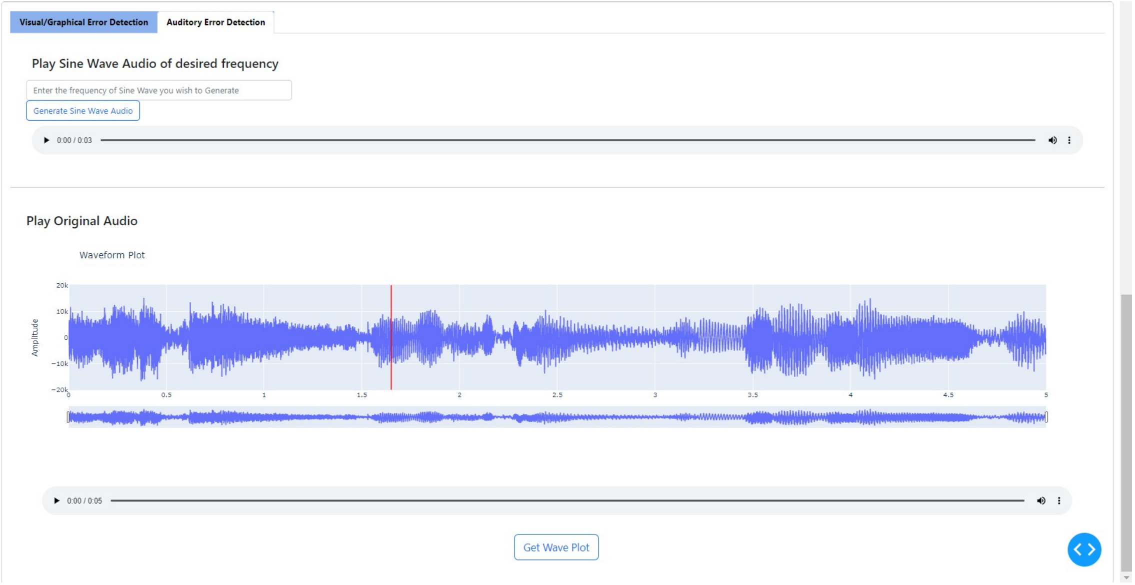


Figure 3.3: Tab for Auditory Error Detection

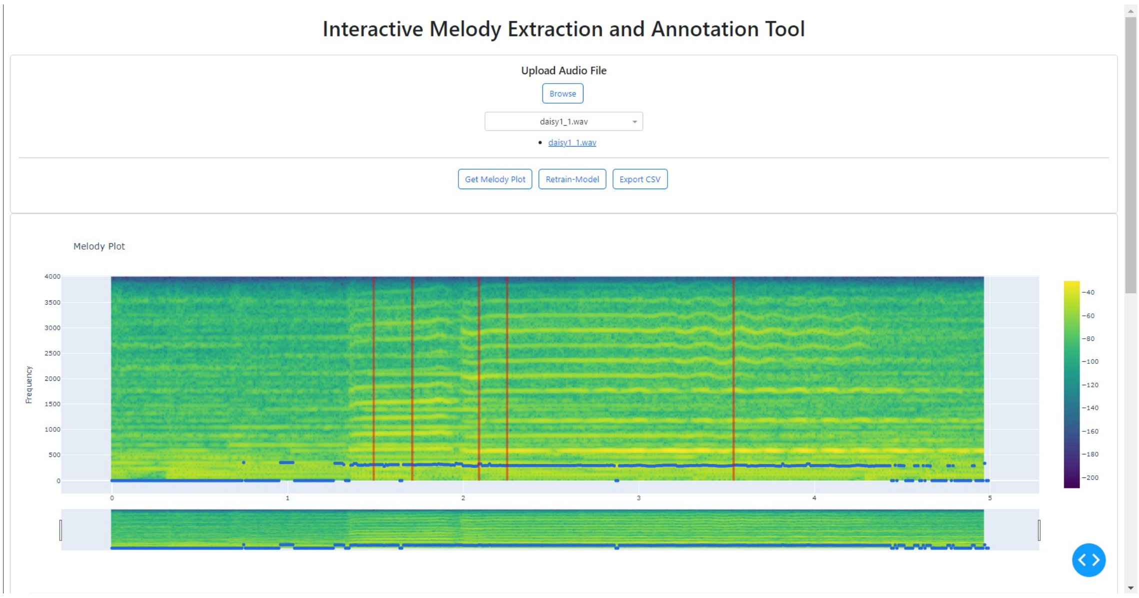
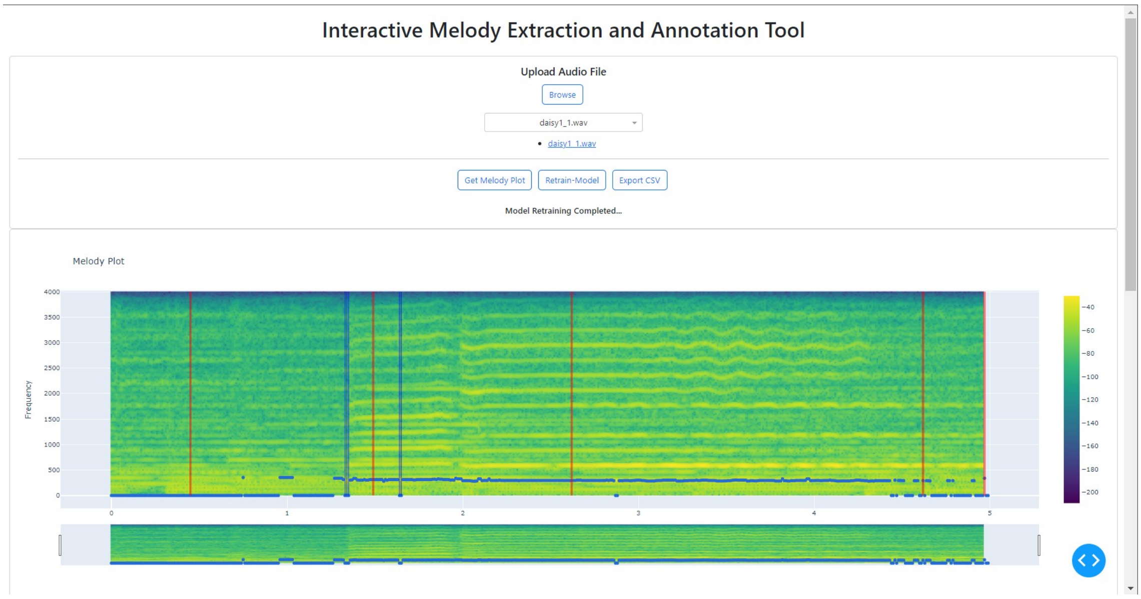


Figure 3.4: ”Get Melody Plot” button yields plot. Red shows top 5 least confident regions.



Interactive Melody Extraction and Annotation Tool

Upload Audio File

daisy1_1.wav

- daisy1_1.wav

Model Retraining Completed...

Figure 3.5: Plot after model retraining. Previously annotated regions are highlighted blue.

3.4 Advantage over previously available tools

It is possible to annotate melody pitch using several different tools and software program. However, each one of them is best suited for a particular purpose. One cannot annotate a specific data point in the known application.

This tool does not have any limitations in terms of data point annotation. The application is simple to use. Before the model is retrained, the tool records the annotations made to the points. It also highlights the frames with low confidence, making it easier for the user to rapidly identify the locations requiring annotation and more apparent where those locations are. In addition, the machine learning model can be easily deployed in the back-end so that a visual examination of the model's performance can be carried out. Aside from these specific aspects, the front-end operations of the various software and program that are currently accessible do not differ significantly from one another.

4. FUTURE SCOPE

4.1 Potential Improvements

The following are some of the possible enhancements that can be made to the tool:

- In the Auditory Error Detection section, overlapping the melody audio file with the original audio will make it easier for users to spot errors in the predictions provided by machine learning models. This will allow for faster error identification.
- For a spectrogram of a particular audio, we aim to depict the least confident time frames by a separate color bar, so that visually it is easier to annotate those points.
- The user experience can be improved by eliminating some of the possible redundant elements. This will make tool more user friendly.
- Currently, the web application is being executed on a client-side server, meaning it is a client-side application. Hosting the application requires multiple adjustments to the back-end handling of the application.

Bibliography

[1] Melody Extraction On Vocal Segments Using Multi Column Deep Neural Network
http://m.mr-pc.org/ismir16/website/articles/119_Paper.pdf

[2] Frequency Temporal Attention Network For Singing Melody Extraction
https://arxiv.org/pdf/2102.09763.pdf

[3] Deep Salience Representations For F0 Estimation In Polyphonic Music
https://brianmcfee.net/papers/ismir2017_salience.pdf

[4] Vocal Melody Extraction Using Patch-Based CNN
https://arxiv.org/pdf/1804.09202.pdf

[5] A Streamlined Encoder/Decoder Architecture For Melody Extraction
https://arxiv.org/pdf/1810.12947.pdf

[6] Deep Domain Adaptation For Polyphonic Melody Extraction
https://www.researchgate.net/publication/364689954_Deep_domain_adaptation_for_polyphonic_melody_extraction

[7] GAN-based Domain Adaptation For Object Classification
https://ieeexplore.ieee.org/document/8518649

[8] Domain Adaptation In Images By Hung Yi Lee
speech.ee.ntu.edu.tw/~hylee/ml/ml2021-course-data/da_v6.pdf