Real-Time Conducting Feedback using Computer Vision and MediaPipe

By

Vivek Bharadwaj

CSC 498: Mentored Research
Date of Submission: May 15, 2023
Mentor: Dr. Salgian

1. ABSTRACT

In this paper, we present a program to detect a conductor's movement using MediaPipe, while providing feedback on form and correctness. This is a different approach from previous work that used the Myo armband and Microsoft Kinect to detect conducting movement. We use an iPhone camera to record conducting gestures from students in Dr. Eric Laprade's class to collect joint coordinates. By tracking the wrist, shoulder, and nose coordinates, a video is output to alert the user of their beats per minute, number of beats, swaying, mirroring, and the direction of their view. The program can provide feedback to help aspiring musicians practice on their conducting form and rhythm till perfection.

KEYWORDS
MediaPipe Pose, MediaPipe FaceMesh, Conducting Gestures, Human-Computer Interaction

2. MOTIVATION & BACKGROUND

Instrumentalists have numerous resources at their disposal to perfect their practice, including expert guidance, self-teaching methods, and the immediate feedback that comes from listening to their own performance. However, when musicians start learning the proper techniques to conduct an orchestra, they often face a unique challenge. Outside of formal lessons and critiques from their professor, aspiring conductors lack the necessary tools to effectively practice and evaluate their techniques independently. Watching oneself in the mirror may offer some insight, but self-assessment can often lead to biased judgments and leniency in evaluating one's own performance.

Our goal is to develop a system that provides immediate feedback on conducting performance, enabling aspiring conductors to access individual resources to improve their skills. In this research paper, we introduce a conducting feedback system using computer vision and the MediaPipe framework to detect and analyze a conductor's movements, providing valuable feedback on form and correctness. MediaPipe, an open-source, cross-platform framework created by Google, facilitates the development of real-time conducting feedback using computer vision. MediaPipe's proficiency in handling tasks such as human pose estimation, hand tracking, and facial landmark detection makes it particularly well-suited for detecting and analyzing conductor movements (See Figure 1).

Figure 1: MediaPipe Skeleton

Building upon previous work that utilized the Myo armband and the Microsoft Kinect, our approach leverages the advanced capabilities of MediaPipe to collect more reliable joint coordinates and evaluate crucial aspects of conducting, such as beats, mirroring, swaying, and gaze direction. Dr. Eric Laprade, the conductor of bands at The College of New Jersey, and his students, who are currently participating in an independent study, collaborated with us on this project. With their help, we recorded and processed videos of them conducting various beat patterns, such as 2/4, 3/4, and 4/4 (See Figure 2).

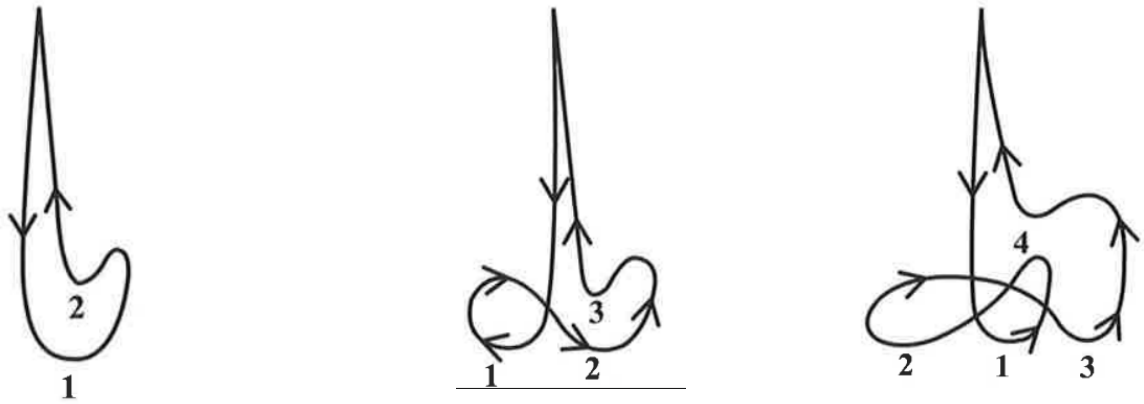

Figure 2: Conducting gestures for 2/4, 3/4, and 4/4 Beat Patterns

The proposed system offers accurate and correct feedback on key performance indicators, including Beats per Minute (BPM), number of beats, swaying, mirroring, and gaze detection.

3. DESIGN

By building upon the work done in past semesters, there was already an extensive understanding of the requirements and goals for a system to analyze conducting techniques. Many algorithms had been developed and tested with good results to detect conductor movement, but there was not much work done to utilize the MediaPipe framework using Python. The aim was to create a system that provides real-time feedback using computer vision and MediaPipe.

3.1 Determining Requirements

The primary goal of the system is to offer correct and accurate feedback that is easy for the user to observe. The system was originally supposed to provide feedback on BPM, number of beats, swaying, and mirroring. However, Dr. Eric Laprade and his students offered more insight into other crucial aspects of conducting that are more valued by conductors, such as gaze direction. Thereafter, we updated our system requirements to include gaze direction, with the intent to reduce the amount of time that a conductor may look at a section. The feedback output needed to be simple and well-organized, facilitating quick comprehension. In order to achieve these goals, it was necessary to adapt newer, emerging technology for collecting joint coordinates. As a result, we chose to employ Google MediaPipe as an alternative to previous approaches that used Myo Armband and Microsoft Kinect. Implementing MediaPipe involved understanding its capabilities and functionalities, which allowed us to provide a more effective and accurate tool for conductors.

3.2 MediaPipe

As stated previously, previous versions of the algorithms used for data analysis were developed using the Myo Armband and Microsoft Kinect. A proper understanding of the MediaPipe framework was required for the development of this system. Initially, the first task was to utilize the MediaPipe Pose model, rendering the user's skeletal structure. The MediaPipe Pose model is able to track the key joint coordinates of the body, including the wrists and shoulders (See Figure 3).

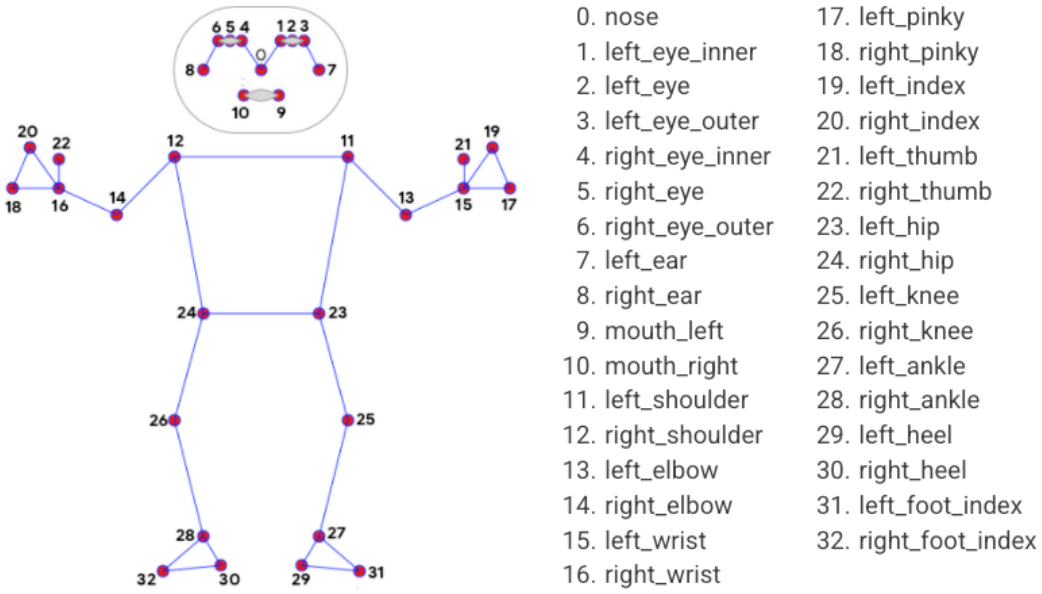

Figure 3: MediaPipe Pose Landmark Model with 33 pose landmarks

Using the MediaPipe FaceMesh model, we were tasked with rendering the FaceMesh contours on the user's face, allowing for tracking of the irises and nose landmarks (see Figure 4). Although our Pose model allows for tracking of lower body joints, the collection of those landmark coordinates is unnecessary since those joints are not used for conducting.

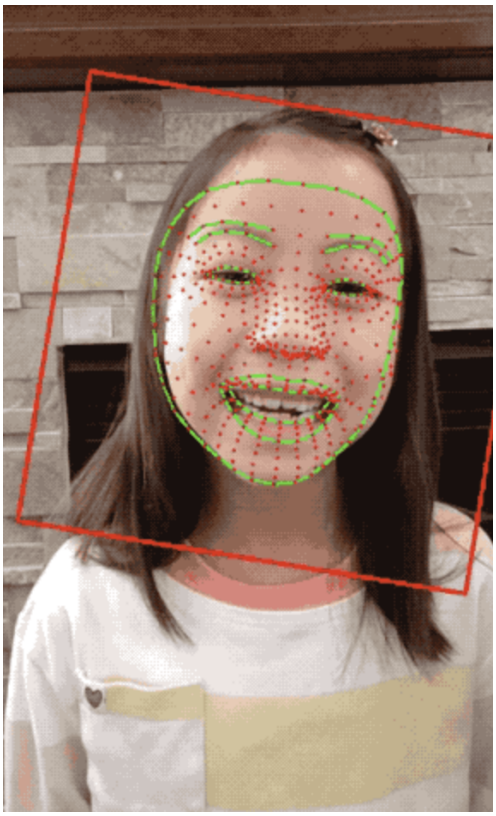

Figure 4: MediaPipe FaceMesh Landmark Model

4. METHODS

The system followed a series of steps during the analysis process. Initially, the video to be analyzed was input, and the joint coordinates of interest were collected in pickle files. Pickle files are a type of file format in Python that allows for efficient storage and retrieval of complex data structures, such as arrays holding wrist coordinate values. Next, the user had to provide the names of both the pickle files and the video for analysis. The system then generated the desired video output, complete with overlaid text offering feedback on form and tempo. While the script runs MediaPipe, the system checks for user input of 'q' to end the analysis if the user wants to use another video or start again.

4.1 Beat Detection

The beat detection module analyzes the video input to determine the number of beats and the beats per minute (BPM) in the conductor's performance. Detecting beats is useful for conductors for several reasons. It ensures conductors that they are maintaining a consistent tempo throughout the performance, which is essential for their ensemble to stay together and deliver a

cohesive musical experience. By accurately detecting beats, it will help conductors clearly communicate the intended tempo and rhythm of the musical piece with their musicians. Lastly, conductors can identify areas for improvement and work on refining their technique by monitoring their own beat patterns, ultimately leading to improved performance quality.

4.1.1 Number of Beats

Conducting patterns in music symbolize the rhythmic structure being conveyed by the conductor. For instance, in a 2/4 pattern, there are two beats per measure; in a 3/4 pattern, there are three beats; and in a 4/4 pattern, there are four beats.

The hand, and thus the wrist, often make motions on the beat, reaching its lowest point, or a 'low' in the y-coordinate data. As the y-coordinate value is 0 at the top of the image, these lows in the y-coordinate data are actually seen as peaks when plotted. In the 2/4, 3/4, or 4/4 pattern, the relative lows are noticeable at the points marked with numbers in Figure 2. Specifically, in a 2/4 pattern, the points marked with '1' and '2' are relatively low points that the wrist reach, which is shown in Figure 5. Similarly, in a 3/4 pattern, the points marked with '1', '2', and '3' are another set of relatively low points that the wrist reach, demonstrated in Figure 6. Lastly, in a 4/4 pattern, the points marked with '1', '2', '3', and '4' are a set of relatively low points that the wrist reaches, which is shown in Figure 7.

The function 'find_peaks' from the SciPy library [6] is used to detect these local maxima in the y-coordinate data. 'find_peaks' returns the indices of relative maxima in a 1-D array which allows for the plotting of indices over the left wrist y-coordinate data. By doing so, we can visually see the detected beats as they correspond to the peaks in the plot.

Once the script runs MediaPipe to estimate the pose landmarks, it checks whether the current frame is at one of these peak indices. If it is, this indicates that a beat was conducted, and the beat counter is incremented by 1. The beat number is overlaid on the video for visual reference while the process repeats for each frame.

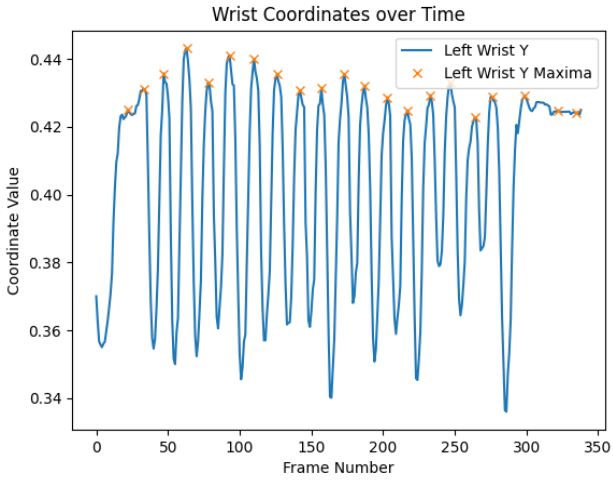

Figure 5: Left Wrist Y Coordinates along with their local maxima for a 2/4 beat pattern

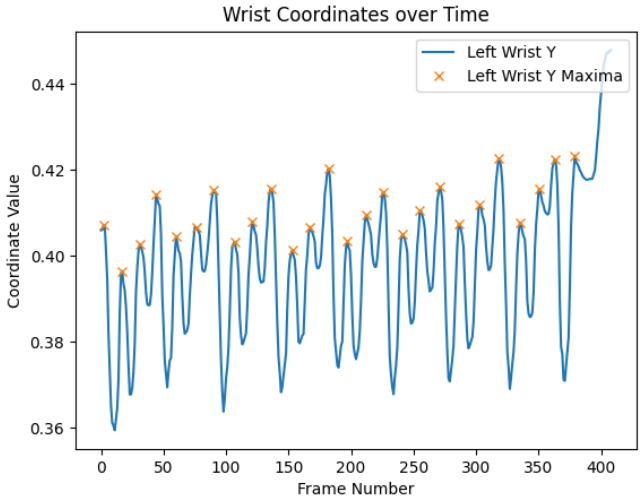

Figure 6: Left Wrist Y Coordinates along with their local maxima for a 3/4 beat pattern

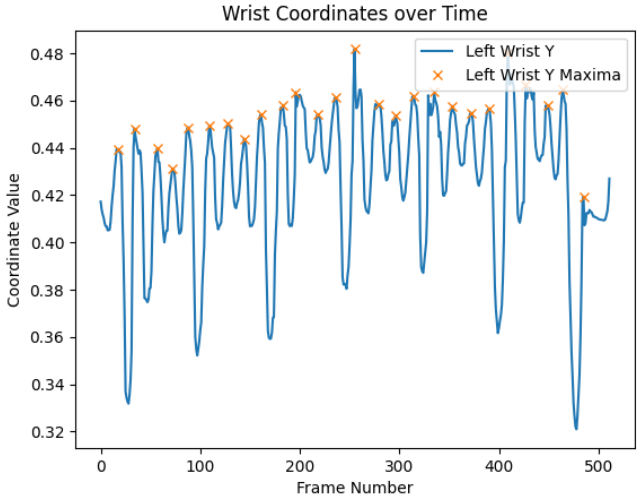

Figure 7: Left Wrist Y Coordinates along with their local maxima for a 4/4 beat pattern

4.1.2 BPM

While calculating the Beats per Minute (BPM), our initial method involved computing the average BPM for the entire video. However, this approach was not helpful for users since they could not ascertain their tempo at a specific point in the video. As an alternative, we opted to calculate the difference between the current beat frame and the previous beat frame. By dividing this difference by the frame rate (29.98 FPS when using an iPhone camera), we determined the time duration between the current and previous beats. Finally, dividing 60 by this duration gave

us the updated BPM, which was displayed as overlaid text. If no beat was detected, the overlaid text would display the previous BPM.

$$BPM = 60 / ((Beat\_frame - prev\_beat\_frame) / frame\_rate)$$

4.2 Swaying

Excessive swaying could potentially distract or confuse members of the ensemble. A conductor may want to minimize swaying to make their movements more efficient and their cues clearer to musicians. To minimize swaying, we closely monitored the position of the user's shoulders during the video because the shoulders should be stationary if the user is not swaying. By calculating the midpoint between the left and right shoulder x and z coordinates, we established an initial reference point. We then set a threshold for swaying by multiplying the initial midpoint x coordinate by 0.08. As the video progressed, we compared the current shoulder midpoint with the initial reference point. If the difference exceeded the threshold in either x or z direction, a "swaying alert" was displayed on the screen. Although integrating the z-coordinate will provide a more comprehensive assessment of the user's swaying behavior, the current MediaPipe framework did not present the most optimal results when plotting the z-coordinate for us to develop an algorithm that detects swaying forward and backward.

4.3 Mirroring

Mirroring in conducting is where both hands of the conductor move in sync or are in the same position relative to the body's midline. This practice is often discouraged by conducting experts because the non-baton hand can be used to signal crucial moments in a composition, providing additional information beyond the tempo and beat patterns conveyed by the baton hand. To detect this practice, we assessed mirroring by comparing the y coordinates of both wrists and their respective distances from the shoulder midpoint. If the difference between the y coordinates of the wrists was less than 0.05, and the difference between each wrist's x coordinate distance from the shoulder midpoint was also less than 0.05, a "mirroring alert" was displayed on the screen.

In our analysis, we observed symmetry when mirroring was employed. When the right or left wrist coordinates were plotted over time during periods of mirroring, the trajectories of the two wrists exhibited symmetry. To illustrate this, we drew a line of symmetry along the horizontal axis (See Figure 8). The plots of the wrist coordinates appeared as mirror images on either side of this line, providing a clear visual representation of the mirroring behavior. This graphical representation not only helped us in identifying the mirroring technique but also in demonstrating and quantifying its occurrence.

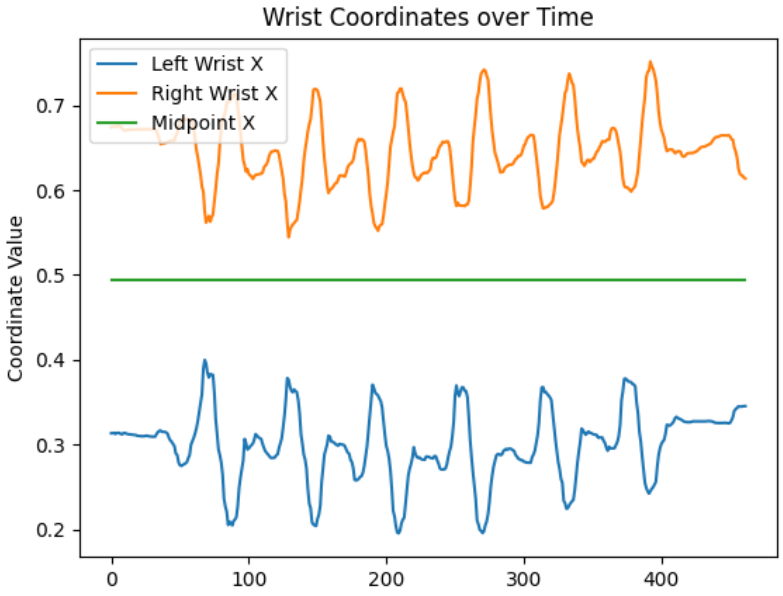

Figure 8: Wrist Coordinates with a line of symmetry

4.4 Gaze Detection

In addition to detecting swaying and mirroring, we also implemented a method for determining the user's gaze direction. Analyzing gaze direction is essential as it enables conductors to effectively communicate with their ensemble and maintain proper focus. Due to challenges in collecting iris coordinates, we chose to prioritize the direction that the conductor is facing by using nose coordinates rather than relying on gaze detection. We compared the current x coordinate of the user's nose to its initial x coordinate position, utilizing a threshold of 6% of the initial midpoint. If the difference in the nose position exceeded this threshold, the gaze direction was determined, and the corresponding feedback, such as "Looking straight," or "Looking to the side," was displayed on the screen (See Figure 7). This real-time feedback allows conductors to adjust their gaze direction as needed for better communication and control during performances.

Figure 9: Gaze direction using the FaceMesh Model

5. RESULTS

Our results demonstrate that the algorithm effectively detected beats and accurately calculated the BPM, providing users with valuable feedback on their conducting performance. The system monitored the user's conducting form and rhythm by analyzing their shoulder and wrist positions. The output video displayed an overlay of feedback analysis that alerted the user of their current BPM, the number of beats, as well as any swaying, gaze direction, and mirroring issues (See Figure 8).

The comprehensive analysis offered by our system allowed users to gain insights into their conducting technique, helping them identify areas for improvement. By addressing swaying, gaze direction, and mirroring, conductors can refine their skills to achieve better control, communication, and focus during performances. The success of our method in detecting beats and providing real-time feedback on various aspects of conducting form highlights its potential as a valuable training tool for conductors at all experience levels.

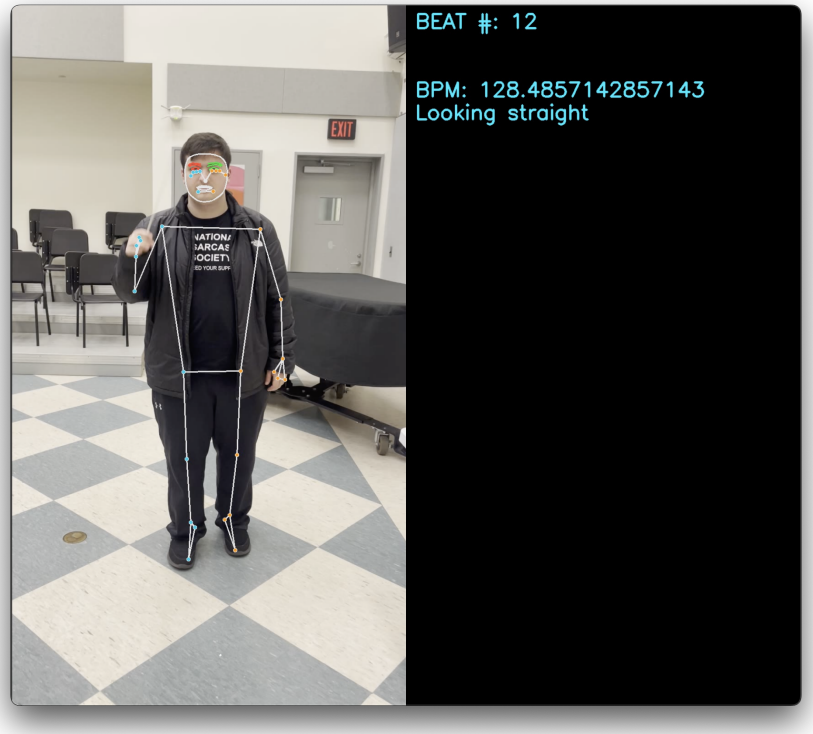

Figure 10: Overlay of Feedback Analysis on Output Video

6. FUTURE WORK

In the continued development of our Conducting Feedback system, we have identified several key areas for future work and potential enhancements to further improve the accuracy and applicability of the tool.

To ensure the robustness and accuracy of our algorithms, we plan to conduct more testing using a wide range of conducting beat patterns and a larger dataset of videos. By doing so, tests will verify the algorithm's ability to effectively adapt to various conducting styles and cater to a broader audience of users.

Lastly, we envision creating a real-time application that offers users immediate feedback on their conducting performance. Currently, our script outputs a video including the overlaid text feedback. By developing a real-time feedback application, it would allow users to make adjustments and improvements in real-time, enhancing the overall user experience and expediting the learning process for conductors seeking to refine their skills.

By pursuing these future developments, we hope to establish a more versatile and effective tool that enables conductors to optimize their technique and ultimately enhance their musical performances.

7. REFERENCES

[1] Andrea Salgian, David Vickerman “Computer-Based Tutoring for Conducting Students”, The 42nd International Computer Music Conference (ICMC 2016), Utrecht, The Netherlands, September 2016.

[2] Andrea Salgian, David Vickerman, David Vassallo, “A Smart Mirror for Conducting Exercises”, The 25th ACM Multimedia Conference – Thematic Workshops, Mountain View, California, October 2017.

[3] Anon. Pose. Retrieved May 1, 2023 from

https://google.github.io/mediapipe/solutions/pose.html

[4] Anon. FaceMesh. Retrieved May 1, 2023 from

https://google.github.io/mediapipe/solutions/face_mesh.html

[5] Nowak, J., & Nowak, H. (2002). Conducting the music, not the musicians. Carl Fischer.

[6] SciPy Community. (2023). find_peaks - Peak finding. Retrieved May 1, 2023, from https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html