Real-Time Conducting Tutor: Computer Vision-Based Analysis

Jeffrey Ernest
Faculty Mentor: Dr. Andrea Salgian
The College of New Jersey, Computer Science Department

Abstract

This work presents a real-time conducting tutor utilizing Google's MediaPipe framework for precise pose estimation. The system leverages computer vision to analyze conducting gestures and provides instantaneous pedagogical feedback. An interactive Graphical User Interface (GUI) facilitates independent practice by visualizing performance metrics in real-time.

Background

Conducting is a complex skill that relies on precise timing, acute spatial awareness, and consistent technique to successfully lead a performance. To provide real-time, objective feedback on these intricate gestures, this system utilizes MediaPipe, Google's advanced framework for precise pose estimation and landmark detection. The application is built using the Python ecosystem and seamlessly integrates OpenCV for computer vision, NumPy for numerical computing, pydub for audio processing, and Tkinter to power the graphical user interface.

Methodology

Beat Configuration
Allows users to fully customize their practice session by selecting specific time signatures (e.g., 4/4, 3/4) and tempo (BPM) settings.

Beat Accuracy
Validates precision by tracking the conductor's hand trajectory in real-time to ensure it hits the correct beat targets.

Figure 2: 4/4 as displayed in program

Figure 3: 4/4 time signature path

Metronome
Provides precise, synchronized metronome clicks to help users internalize the tempo and maintain consistent timing throughout the session.

Swaying Detection
Alerts the user to excessive upper-body movement by monitoring core stability to promote a stable conducting style for learning the basics.

Mirroring Detection
Provides instant visual feedback when the expressive hand mirrors the conducting hand. This increases awareness of hand symmetry, supporting deliberate expressive choices.

Elbow Tracking
Uses vector calculations to determine the shoulder joint angle to control excessive arm movement.

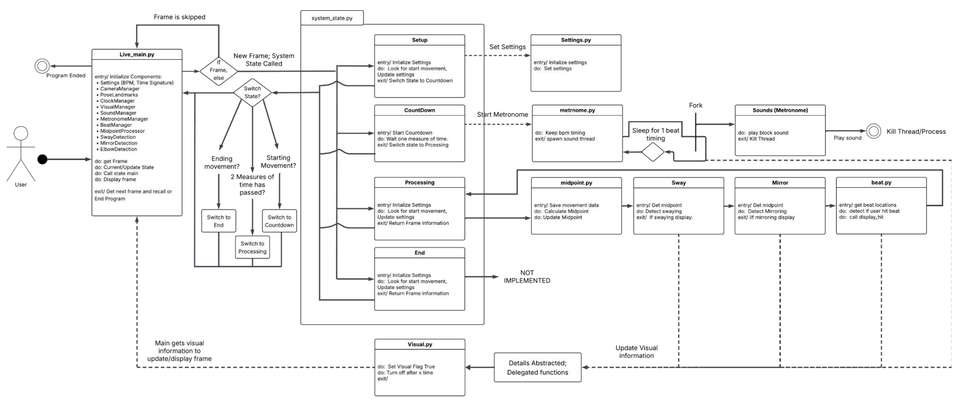

Figure 1: State diagram of live architecture

System Architecture

The system builds upon existing video processing work and extends it for live real-time analysis:

Video (Past Work)
Analyzes pre-recorded video files to provide summative feedback on conducting technique. Implemented using a PyGame based graphical user interface.

Graphical User Interface (GUI)
Tkinter-based GUI featuring an intuitive workflow and adaptive video resizing that maintains aspect ratios. The interface includes settings panel to ensure a customizable user experience.

Live
Processes real-time video to provide immediate formative visual feedback during conducting practice. To maintain system stability and tempo consistency, the metronome function is managed on a separate thread, ensuring audio timing is decoupled from the main visual processing loop. The session workflow is organized in four phases (setup, countdown, processing, ending).

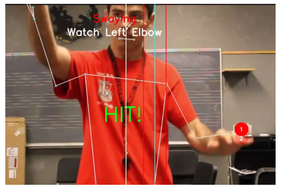

Figure 4: Feedback display

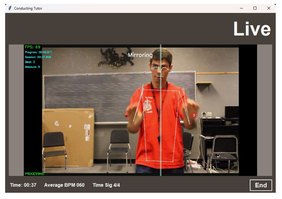

Figure 5: Live processing frame

Future Work

- Implement machine learning to automatically track and calculate average BPM.
- Develop real-time detection for conducting dynamics (loudness/intensity).
- Add features to customize conducting paths and specific beat patterns.
- Generate detailed end-of-session reports showing metrics like body sway and hand mirroring.

Results

Performance:
Achieved consistent 30 FPS processing for both live camera feeds and video files that are simulated as live input.

Stability:
Implemented robust state management for seamless session transitions.

Visual Feedback:
Generated dynamic overlays for landmarks and feedback without performance lag.

Validation & Tracking:
Verified system accuracy and specific patterns (e.g., 4/4) using diverse datasets from classroom and online sources.

Interface:
Designed an intuitive GUI complying with Shneiderman's Eight Golden Rules.

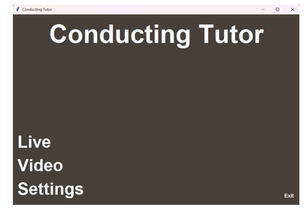

Figure 6: Home frame

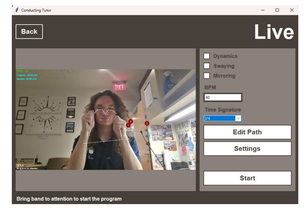

Figure 7: Live configuration frame

References

[1] Google. (Aug. 2024). MediaPipe. ai.google.dev/edge/mediapipe.
[2] Salgian, A., Burke L., Ernest J. (2025). Visual Analysis of Conducting Gestures. ICMC 25.
[3] Chin-Shyurng, F., et al. (2019). Real-Time Musical Conducting Gesture Recognition. Applied Sciences, 9(3), 528.
[4] Carthen, C. D., Dascalu S. M. (2015). MUSE: A Music Conducting Recognition System. University of Nevada, Reno.
[5] Shneiderman, B., et al. (2016). Designing the User Interface (6th ed.). Pearson.
[6] Google. (Nov. 2025) Gemini. gemini.google.com.

Acknowledgments: The author thanks Dr. David Vickerman (San José State University) for his expertise and assistance with video data collection.