Real-Time Conducting Analysis Using Computer Vision & Machine Learning

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

Abstract

This work presents a real-time conducting tutor that combines pose estimation (MediaPipe) with a supervised beat-detection model trained on labeled wrist motion. The model runs during live sessions to estimate when beats occur (and time signature), enabling immediate feedback and tempo display in an interactive setting. The system is intended as a practice tool for objective feedback and as a testbed for vision-based, learning-assisted music pedagogy.

Background

Challenge: Conducting is a complex skill that relies on precise timing, spatial awareness, and consistent technique.

Barrier: Objective, real-time feedback on these gestures is difficult for students to obtain without a direct mentor present.

Approach: This system uses MediaPipe for landmark-based tracking and a Keras model trained on wrist motion to estimate beats-per-minute during live practice, providing feedback on both style and movement.

Technical Stack: The application is built in Python, utilizing OpenCV for video handling, NumPy for numerical data, TensorFlow/Keras for the beat detection model, pydub for the audio metronome, and Tkinter for the graphical interface.

Methodology

Learning Pipeline (Model Training)

- **Collect:** Extracted conducting-hand wrist data frame-by-frame from YouTube and classroom videos using MediaPipe.
- **Preprocess:** Normalize wrist (x, y) with a bounding box so scale is stable across distance and framing. Built 11-frame sliding windows for training.
- **Train:** Label each window beat vs no beat. Train / validation split (80/20). Train an LSTM to output beat detection(~5,000+ windows).

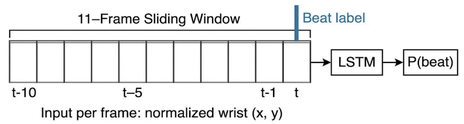

Figure 2: 11-frame window of normalized wrist motion is input to an LSTM.

Live Beat Integration: The trained LSTM model is loaded into the live application to predict beats in real-time, driving on-screen markers so users can receive their current beats per-minute.

Pose-Based Technique Cues:

- **Swaying:** Monitors core stability and flags excessive upper-body motion.
- **Mirroring:** Flags when the non-conducting hand mirrors the pattern (symmetry cue).
- **Elbow Tracking:** Uses shoulder–arm geometry to identify excessive arm movement.

Display Customization:

Toggles for the hand marker, metronome, and beat visuals allow for a highly personalized training session.

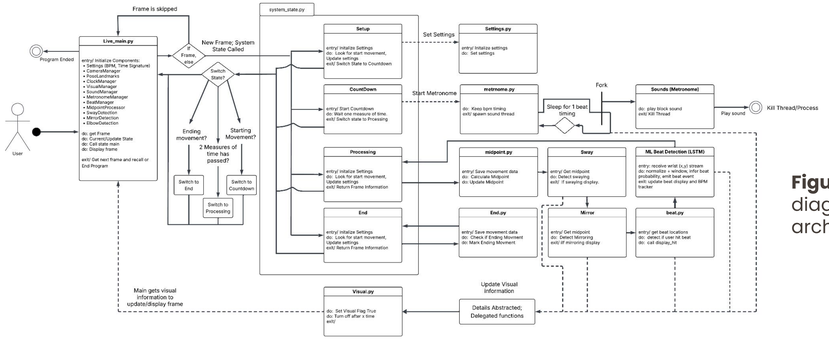

Figure 1: State diagram of live architecture.

System Architecture & Evolution

The application supports both offline video review and live practice, building upon previous work by completely overhauling the live processing pipeline this semester.

Past Work:

Offline video analysis and foundational live tracking existed, but real-time beat detection was impossible because the legacy algorithms required the entire video's data at once.

Current Contribution:

- **Live ML Integration:** Replaced the legacy beat-detection algorithms by training a new machine learning model, enabling instant beat detection for live practice.
- **Finalized Live Pipeline:** Streamlined the user experience by enabling customizable practice settings and integrating "ending movement" logic for touchless workflows.
- **Standalone Deployment:** Packaged the Tkinter based application into a fully distributable build that requires no external setup.

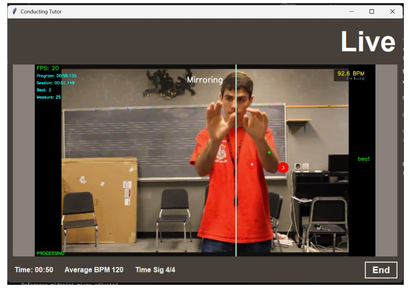

Figure 3: Live processing interface with objective feedback overlays (rolling BPM, beat-detection indicator, and mirroring cue).

Future Work

- 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

Beat detection (learned model)

The LSTM beat detector was evaluated against held-out labeled sessions. Matching predicted beat times to ground-truth beats within ±2 video frames, the system achieved ~94% correct beat detection.

Live integration & feedback

The model runs inside the live pipeline and drives on-screen cues: instant beat indication and rolling tempo (BPM) so users get continuous timing feedback during practice.

Settings & usability

Session settings (time signature, tempo, metronome, beat markers, conducting-hand marker) let users tailor what they see and hear. Visual feedback stays readable by optional

Distribution

Packaged build that can download, install, and run without setting up of tooling.

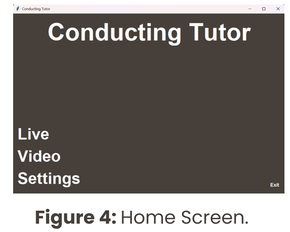

Figure 4: Home Screen.

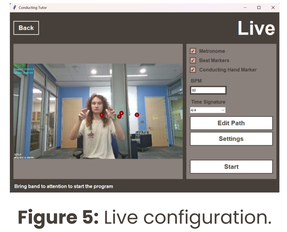

Figure 5: Live configuration.

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.
[5] Google. (Nov. 2025) Gemini. gemini.google.com.
[6] Shneiderman, B., et al. (2016). Designing the User Interface (6th ed.). Pearson.
[7] Chollet, F., et al. Keras. https://keras.io/
[8] TensorFlow Developers. TensorFlow. (2015)
[9] GeeksforGeeks. What is LSTM?

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