GESTURE BASED CONTROLS FOR ONLINE VIDEO CONFERENCING APPLICATIONS

GOMATHI GANESAN*(112743868) - goganesan@cs.stonybrook.edu*

SUNY Stony Brook University

SRIRAM T K (*112670605) - stirupatturk@cs.stonybrook.edu*

SUNY Stony Brook University

ABSTRACT:

The current global pandemic has led to an increased dependency on online video conferencing applications in learning, corporate meetings, social and cultural gatherings. The features like 'mute', 'turning video on/off' require button clicks and hence, become less interactive with users. To enhance user interaction, we propose gesture based controls for these features. For instance, a user can put a finger on their lips to mute themselves. Similarly, in online classes a student can simply raise their hand to ask queries. This kind of gesture controlled application increases the user engagement, creating an interactive environment in these virtual spaces.

INTRODUCTION:

In the present world, people can acquire a lot of knowledge from texts, pictures and video materials from the internet by themselves, but still teaching in the classroom is the most important and effective way of delivering education. With the current global pandemic and a shift to online learning, there has been an increased use of video conferencing applications like Zoom. Teacher-student interactions make the classroom environment more proactive rather than just listening. Among them, raising hands, muting/unmuting, starting and stopping the classes are some of the most basic activities performed. If an intelligent system can identify hand-raising, it will help to analyze the teaching process to make it better. In this case, the most basic problem, hand-raising gesture recognition is required to be sorted out accurately in an intelligent classroom.

RELATED WORK:

The former 1 works on a two stage process for Hand raising gesture recognition in Classrooms. The first stage is pose estimation which is used to get the key points of students. Easy-to-detect body parts such as shoulders and neck help to locate difficult-to-detect parts such as elbows and knees.

The later 2 first detect faces in each frame of the video sequence in order to define the region of interest (ROI). Then the system locates arms in the region of interest by analyzing the geometric structure of edges on the arm instead of directly detecting the hand. The location and the orientation of a detected arm with respect to the location of the face is used to make a decision on whether or not a person is raising their hand. Finally, the frequency of a raised hand detected in previous frames is used to eliminate false positive detections and robustly detects persons who are raising a hand.

Pose Estimation

- **Single Person Pose Estimation.** In single person pose estimation, the pose estimation problem is simplified by only attempting to estimate the pose of a single person, and the person is assumed to dominate the image content.
- **Multi Person Pose Estimation**. Multi person pose estimation is now attracting more and more attention, because of the high demand for the real-life applications. There are two different types of approaches followed here,
    - **Bottom Up:** This kind of approach directly predicts all human parts first, then design algorithms to find out the relationship between the parts and assign the parts to different people.
    - **Top Down:** This kind of approach has achieved the best results on public datasets such as MPII and COCO. General methods use generic object detection algorithms such as RCNN to detect human bounding boxes first, and then perform single person pose estimation within each box independently.

HIGH LEVEL DESIGN:

Jitsi meet is a popular online video conferencing platform. We have added hand gesture controls for the features supported in it. The list of gestures and their uses are as follows:

- Index finger on lips: To mute audio
- Index finger on chin: To unmute audio
- Raise the hand: To raise your hand in the meeting and get host attention.
- Thumbs up: To start recording the meeting session
- Hand stop sign with all fingers facing vertically up: To stop recording the current session

All these gestures are inspired from American Sign Language(ASL).
Tensorflow js models were used to detect hand gestures from webcam video streams.

GESTURE DETECTION:

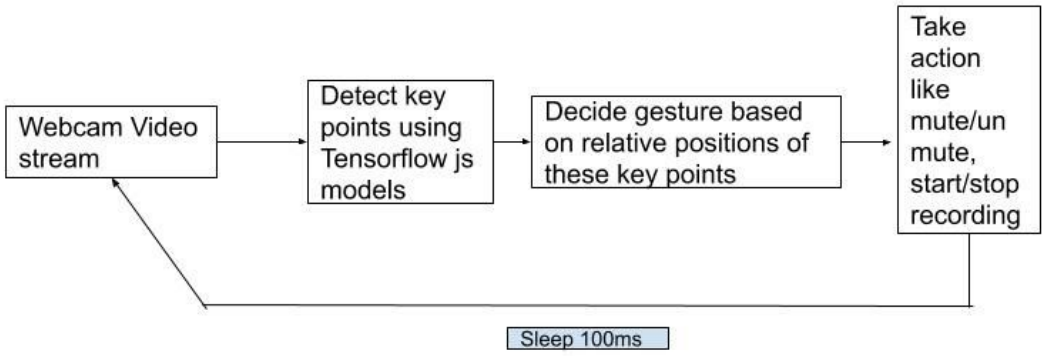


Figure 1: Gesture detection control flow diagram

IMPLEMENTATION:

There is a javascript routine that constantly runs detection on the input from the webcam video stream. Based on various keypoints, the gestures are determined as follows:

- **Gesture for Unmute audio:**
    - Using the blazeface tensorflow model, the key points in the face of the user is found. Then, by using the hand pose model we get the key points of all fingers. The top key point of the index finger's position relative to the lip and nose key points is found. Based on these key points' positions, the gesture of placing the index finger on the chin is detected and used for unmuting audio.

- **Gesture for mute audio:**
    - Using the blazeface tensorflow model, the key points in the face of the user is found. Then, by using the hand pose model we get the key points of all fingers. The top key point of the index finger's position relative to the lip and nose key points is found. Based on these key points' positions, the gesture of placing the index finger on the lip is detected and used for muting audio.

- **Gesture for "Start recording"**:

- The fingerpose tensorflow js model is used to find the key points of all the fingers and define gestures based on the curling and direction of the fingers. The ‘thumbs up’ gesture is defined using these attributes. During detection the confidence value for the thumbs up model is found and if it's greater than 0.85, then the gesture is concluded to the ‘thumbs up’ gesture. This gesture is used to start a recording.

- **Gesture for “Stop recording”:**
  - The fingerpose tensorflow js model is used to find the key points of all the fingers and define gestures based on the curling and direction of the fingers. The ‘stop’ hand gesture is defined using these attributes. During detection the confidence value for the ‘stop’ hand gesture is found and if it's greater than 0.85, then the gesture is concluded to the ‘stop’ hand gesture. This gesture is used to stop a recording.

- **Gesture for “Raise hand”:**
  - Whole body pose detection using the Pose estimation tensorflow js model is done. The relative positions of key points like elbow, wrist, shoulder and face is used to detect if the user is raising hand or not.

EVALUATION:

The design and implementation was evaluated based on the 10 usage heuristics:

1. **Visibility of System status:**
   - The UI components of the features controlled are updated as soon as the user signals a gesture.
   - For instance, the audio status changes when the user uses the gesture related to audio.

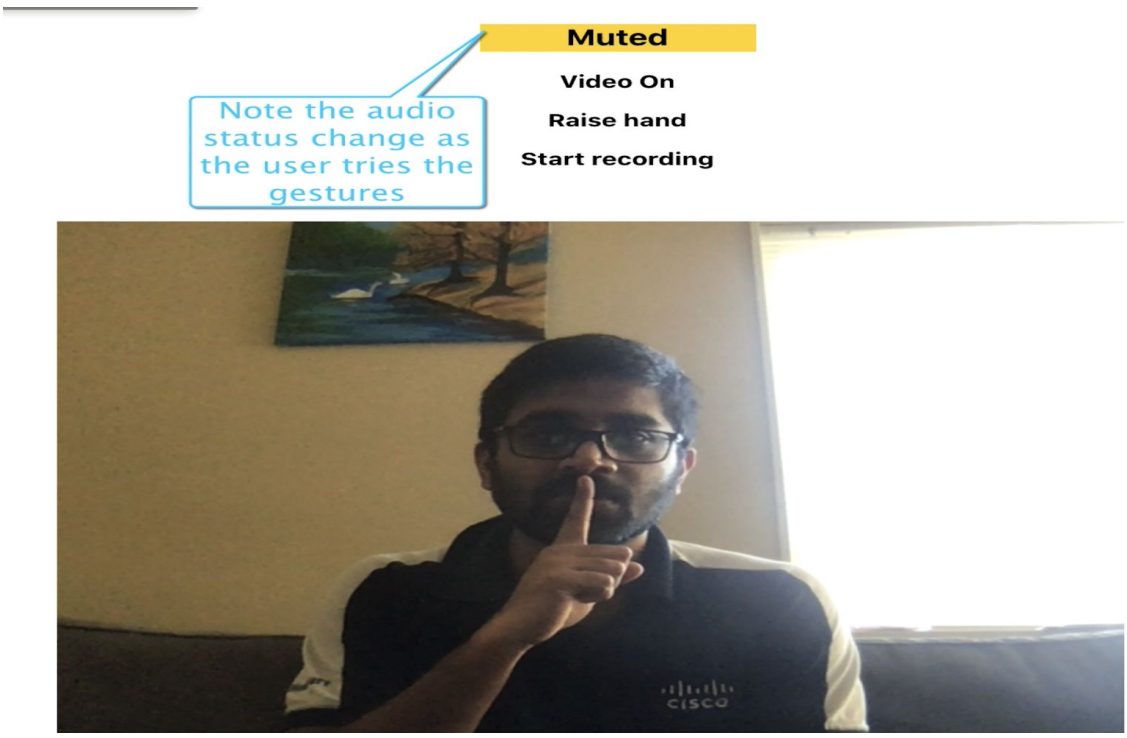
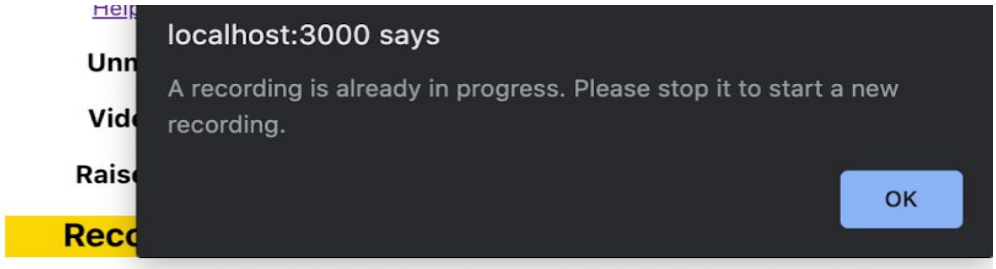


Figure 2: Visibility of System status for muting audio

2. **Match between system and real world:**
    - The hand gestures used here are inspired from the American sign language(ASL). Hence, the design matches with the real world gestures.

3. **User control and freedom:**

    For a mistaken choice like starting a recording when it has already started or stopping a recording which isn't started in the first place, we've handled it. "Exits" for mistaken choices, undo, redo are provided.



4. **Error prevention:**

- Before taking action, each gesture is validated if it's performed with the right context.
- For example, if a user tries to signal “start recording” while a recording is already in progress, the user will be notified about this via an alert message.

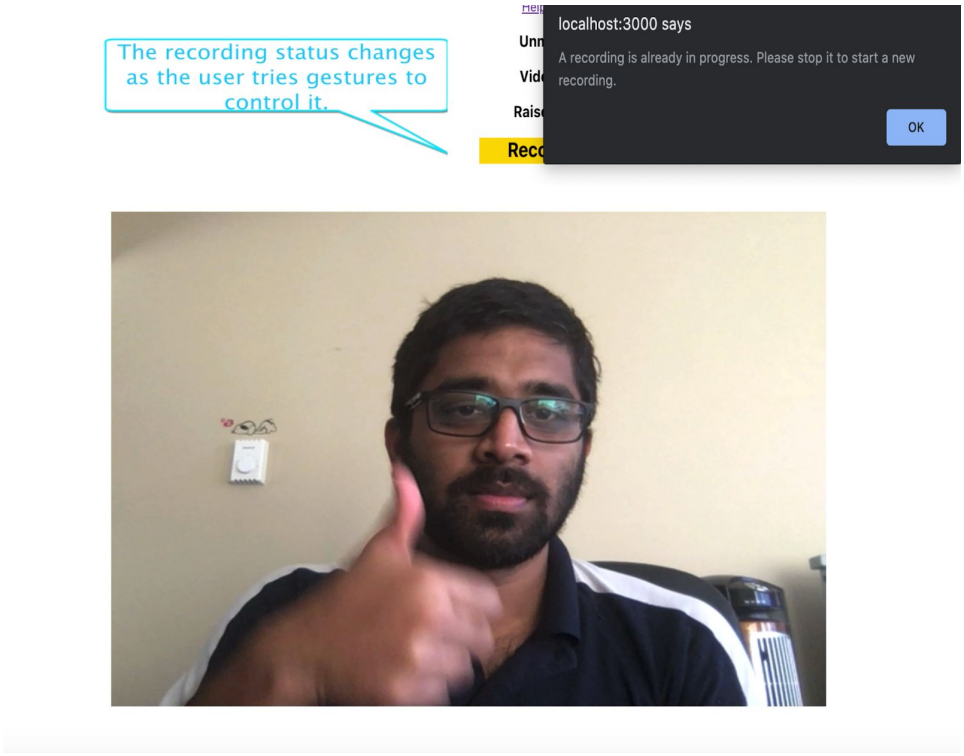


Figure 3: Error prevention for start recording gesture

5. **Recognition rather than recall:**
   - The gestures are based on sign language and hence are more intuitive to use. The user is not forced to memorize gestures.

6. **Aesthetic and minimalist design:**
   - Messages are shown only in case of an error or inappropriate gesture.
   - Irrelevant information isn’t presented at any point.

7. **Flexibility and ease of use:**
   - The selected gestures are easy to use for people of all age groups from children to old age people.
   - Hence, it is highly flexible.

8. **Help users with errors:**
   - The error messages are presented in plain language.
   - No technical jargons are used.

- The messages also convey a solution if needed. Example, if a user tries to signal "start recording" while a recording is already in progress, the user will be notified with the following message:

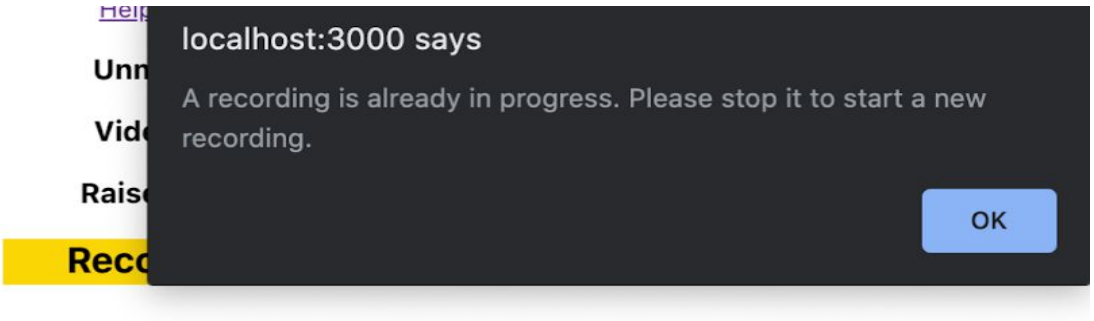


Figure 4: An example of error message shown to user

9. **Help and documentation:**
    - The users are provided with a Help page where they can find the usages of supported hand gesture controls.

FUTURE WORK:

We have designed gestures for some of the controls and built a prototype application to demonstrate all the functionality. We have done the heuristic evaluation as a performance tester. We have tried integrating it with Jitsi and currently facing issues with it. As future work, we want to integrate our functionalities with Jitsi and evaluate it further.

CONCLUSION:

In this paper, we have performed the single person pose estimation by extracting various points on the body, face, palms by employing libraries like HandPose, PoseNet, Blazeface. These points are used to point out *the inclination and the location of the arm, the coordinates of fingers with respect to face.* These are used to identify the various gestures defined which play a significant role in automating various controls like raise hand, mute/unmute, start/stop recording. The performance has been evaluated by heuristic evaluation.

REFERENCES:

- https://dl.acm.org/doi/10.1145/3318396.3318437
- https://ieeexplore.ieee.org/document/6181414

- https://github.com/andypotato/fingerpose/tree/master/src/gestures
- https://github.com/tensorflow/tfjs-models/tree/master/handpose
- https://github.com/tensorflow/tfjs-models/tree/master/blazeface
- https://www.youtube.com/watch?v=f7uBsb-0sGQ&t=1121s&ab_channel=NicholasRenotte
- http://proceedings.mlr.press/v95/zhou18a/zhou18a.pdf
- Web builder tool used - https://nicepage.com/
- https://github.com/nicknochnack/HandPoseDetection