Poster: Your Gesture Can Prevent Oops Moments in Online Meeting

Ashutosh Kr Sinha
ashutosh_2421cs13@iitp.ac.in
IIT Patna, India

Rahul Mishra
rahul_mishra@iitp.ac.in
IIT Patna, India

Hari Prabhat Gupta
hariprabhat.cse@iitbhu.ac.in
IIT (BHU) Varanasi, India

ABSTRACT

The rapid advancements in network and communication technologies have enabled numerous applications that enhance human interaction, with online meetings being one of the most significant. These virtual meetings proved invaluable during the COVID-19 pandemic, allowing people to stay connected despite physical distancing. However, such meetings also increase the likelihood of "Oops moments," particularly when participants are distant from their input devices. This work presents a smart application that recognises hand gestures to prevent embarrassing moments during online meetings. The system provides seamless control over key meeting functions, such as toggling the microphone and camera, taking screenshots, and starting/stopping recordings, all through simple hand gestures. In a survey of 20 participants, 95% found the application effective in improving their meeting experience.

CCS CONCEPTS

• **Computing methodologies** → **Image and video acquisition**.

KEYWORDS

Gesture, Online Meeting, Oops moments, Video

1 INTRODUCTION

Non-verbal communication (NVC) plays a crucial role in conveying over 65% of information, compared to just 35% transmitted through verbal communication. Among the various forms of NVC, Hand Gesture Recognition (HGR) has emerged as a prominent and practical method for interacting with smart devices, such as televisions, computers, and home appliances, air conditioners, enabling control over functions like fan speed and ambient temperature settings [2]. As a result, HGR has become a key research area that leverages sensor technology and computer vision. Although vision-based HGR systems exhibit superior accuracy, they require substantial computational resources [1]. Nonetheless, effective HGR techniques are essential for enhancing human-computer interaction (HCI), providing intuitive, non-traditional interfaces that bypass conventional input devices like keyboards and mice [2].

Online meetings and virtual collaboration have become essential for remote communication across sectors, from corporate settings to educational institutions. The global COVID-19 pandemic highlighted their critical role in enabling seamless collaboration without the need for physical presence. However, the convenience of online meetings also brings the risk of unintended interruptions or embarrassing “Oops moments,” such as forgetting to mute the microphone or inadvertently leaving the camera on (as illustrated in Figure 1 sourced from YouTube[1]). These issues are even more challenging when users are away from their input devices, making it difficult to control features like the microphone or camera quickly. As reliance on remote communication platforms continues to grow, there is a clear need for intuitive, hands-free solutions that allow participants to manage meetings effortlessly and prevent disruptions.

Figure 1: A snapshot of an "Oops moment" during an online meeting, sourced from YouTube.

This poster presents **Meet-Assist**, a smart application designed to reduce the risk of embarrassing "Oops moments" during online meetings by utilizing hand gesture recognition for controlling key functionalities. The system, built using OpenCV and MediaPipe, allows users to control essential features such as muting and un-muting the microphone, turning the camera on or off, taking screenshots, and managing recordings—all through simple hand gestures, like raising a certain number of fingers. This approach eliminates the need for direct interaction with input devices like the keyboard or mouse, ensuring a more seamless and intuitive experience. In addition to gesture control, the system incorporates a predictive mechanism that analyzes minimal user gestures to anticipate and prevent potential Oops moments before they occur.

2 METHODOLOGY

The architecture of Meet-Assist consists of three core modules: (a) Gesture Detection Module, (b) Control Interface, and (c) Oops Moment Prediction Module.

2.1 Gesture Detection Module

We utilize a Convolutional Neural Network (CNN) to recognize hand gestures based on video input. Let x_t represent the video frame at time t, and g_t denote the corresponding hand gesture in that frame. The CNN model is defined as a function $f(\cdot)$, which takes an input image x_t and outputs the probability distribution over the possible gestures:

$$P(g_t|x_t) = f(x_t). \tag{1}$$

[1]https://www.youtube.com/watch?v=NblRUTp60WI

The model is trained on a dataset $\mathcal{D} = \{(x_i, g_i)\}_{i=1}^{n}$, where each sample consists of a frame x_i and its corresponding gesture label g_i. The objective is to minimize the *cross-entropy loss*:

$$L = -\sum_{i=1}^{n}\sum_{k=1}^{K} \mathbf{1}(g_i = k)\log P(g_i = k|x_i), \tag{2}$$

where K is the number of possible gestures, and $\mathbf{1}$ is the indicator function. The trained CNN model is fine-tuned using real-time participant data to ensure high accuracy during live meetings.

2.2 Control Interface

Recognized gestures are translated into system commands through a *mapping function* $\phi(\cdot)$, where each gesture is mapped to a corresponding action. The interface communicates with online meeting platforms via their APIs. In cases where direct API integration is unavailable, the system simulates necessary keyboard or mouse actions $\psi(\cdot)$ to perform the mapped commands. Figure 2 illustrates several examples of operations carried out using hand gestures with the developed application. A sample video demonstrating these features is available at[2].

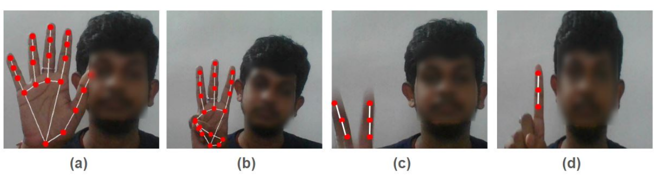

Figure 2: Illustration of the developed application showcasing various fingers performing different actions: a) launching the application, b) displaying captured images, c) initiating video recording, d) taking a screenshot.

2.3 Oops Moment Prediction Module

Based on recent user actions, the *Oops Moment Prediction Module* anticipates potential mishaps (like leaving the microphone unmuted). Let $\mathcal{A}_t = \{a_1, a_2, \ldots, a_m\}$ be the set of actions the user performs at time t. A risk score $R(\mathcal{A}_t)$ estimates the probability of an Oops moment occurring:

$$R(\mathcal{A}_t) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}(a_i \in O), \tag{3}$$

where O represents risky actions (e.g., an open mic at an inappropriate time). If $R(\mathcal{A}_t)$ surpasses a predefined threshold R_{th}, the system alerts the user to avoid potential disruptions. Algorithm 1 outlines the flow of real-time hand gesture recognition, command execution, and Oops moment prediction.

Algorithm 1: Meet-Assist: Gesture-Based Control and Oops Moment Prediction

Input: Video feed V, Pretrained gesture recognition model $f(\cdot)$, Set of recognized gestures $G = \{g_1, g_2, \ldots, g_k\}$, Meeting platform API $\psi(\cdot)$, Risky actions set O, Action history $\mathcal{A}_t$;
Output: Gesture-based control in online meetings, Oops moment warning;

```
Initialize system: Load gesture model f(·), define
  gesture-to-command mapping φ(g_t);
while meeting is ongoing do
    Capture frame x_t from video feed V;
    Perform gesture recognition: P(g_t | x_t) = f(x_t);
    if g_t exceeds confidence threshold τ then
        Map gesture to command: command ← φ(g_t);
        Execute command: ψ(command);
        Log action a_t in 𝒜_t;
    Calculate Oops risk score using (3);
    if R(𝒜_t) > R_th then
        Issue Oops moment warning to user;
```

3 RESULTS AND DISCUSSION

To verify the efficacy of the proposed system, we surveyed with 20 participants, focusing on various aspects of user experience. As shown in Table 1, the participants provided overwhelmingly positive feedback across key categories such as navigation, loading speed, suitability, overall design, interface, and overall experience. Specifically, 95.7% of the responses were positive, indicating a high level of user satisfaction, with the system scoring particularly well in navigation, suitability, and overall experience. A small portion (4.3%) of the feedback indicated a need for improvement, mainly in loading speed and the visual appeal of the interface. These results highlight the system's robustness in providing a smooth and intuitive user experience while leaving room for minor enhancements.

Table 1: Responses from 20 participants. Nav. = Navigation, LoadS. = Loading Speed, Suitab. = Suitability, OD = Overall Design, Inter. = Interface, and OE = Overall Experience.

Response	**Nav.**	**LoadS.**	**Suitab.**	**OD**
Postive	19/20	18/20	20/20	19/20
Need Impv.	1/20	2/20	0	1/20
Response	**Inter.**	**Appealing**	**OE**	**Total**
Postive	20/20	18/20	20/20	**134/140=95.7%**
Need Impv.	0	2/20	0	**6/140=4.3%**

To evaluate the performance of the proposed hand gesture recognition system, we measured the accuracy of the CNN model in recognizing different finger counts. As illustrated in Table 2, the model achieved high accuracy across all finger counts. Specifically, the system performed best in recognizing five fingers, with an accuracy of 98%, followed by four and three fingers, each at 96%. The accuracy slightly decreased to 94% for two fingers and 91% for a single finger. These results demonstrate the robustness of the CNN model in accurately detecting hand gestures, confirming its effectiveness in real-time gesture-based controls.

Table 2: Illustration of the accuracy percentage in recognizing different finger counts using the CNN model.

No. of fingers	**5**	**4**	**3**	**2**	**1**
Accuracy of CNN	**98%**	**96%**	**96%**	**94%**	**91%**

REFERENCES

[1] Haoming Liu and Zhenyu Liu. 2023. A Multimodal Dynamic Hand Gesture Recognition Based on Radar–Vision Fusion. *IEEE Transactions on Instrumentation and Measurement* 72 (2023), 1–15.

[2] Jungpil Shin, Abu Saleh Musa Miah, Md. Humaun Kabir, Md. Abdur Rahim, and Abdullah Al Shiam. 2024. A Methodological and Structural Review of Hand Gesture Recognition Across Diverse Data Modalities. *IEEE Access* (2024), 1–34.

[2]https://drive.google.com/file/d/1KUoiaJIzCGEc1H_LStq2txX394-ugXsq/view?usp=sharing