

HandGesture2Symbol

Project Proposal: Hand Gesture-to-Symbol Recognition System Using Computer Vision

a. Application and AI Algorithm

We aim to develop a system that recognizes hand gestures via computer vision (CV), maps them to predefined symbols (e.g., letters A, B, C), and simulates keyboard input based on the recognized gestures. The AI component involves training a supervised learning model to classify hand joint positions (captured using **OpenPose**) into encoded symbols. This system could serve as an alternative input method for users with mobility challenges or enhance human-computer interaction in scenarios where traditional keyboards are impractical.

b. Task or Problem

The primary challenge is to accurately detect hand gestures in real-time, translate joint coordinates into symbols, and seamlessly integrate this output as keyboard input. Key sub-tasks include:

1. **Gesture Design:** Define intuitive gestures (e.g., thumb touching specific finger joints) to represent symbols.
2. **Robust Detection:** Ensure reliable hand joint tracking under varying lighting conditions and hand orientations.
3. **Low-Latency Processing:** Achieve real-time performance to enable practical use.

c. Work Distribution

- **LU, Zihan:** Deploy OpenPose library for real-time hand joint detection and coordinate extraction.
- **ZHANG, Shuhan:** Collect and preprocess a dataset of labeled hand gesture images/videos (500+ samples per symbol).

- **SHAO, Chenhang:** Design and train a classification model (e.g., SVM, CNN) using the joint position data.
- **XING, Wanli:** Develop a Windows-compatible keyboard simulator that maps model predictions to keystrokes.

d. Schedule Plan

Week	Tasks
Mar 23–29	<div>- Setup OpenPose environment (A).</div> <div>- Design gesture symbols (All).</div> <div>- Begin data collection (B).</div>
Mar 30–Apr 5	<div>- Finalize dataset with labeled gestures (B).</div> <div>- Draft model architecture (C).</div> <div>- Research keyboard simulation tools (D).</div>
Apr 6–12	<div>- Train and validate the model (C).</div> <div>- Optimize OpenPose for real-time use (A).</div>
Apr 13–19	<div>- Integrate model with OpenPose output (A+C).</div> <div>- Develop keyboard simulator prototype (D).</div>
Apr 20–27	<div>- System testing and debugging (All).</div> <div>- Prepare presentation video and final report (All).</div>

Creativity and Practicality

- **Creativity:** Customizable gesture encoding allows users to define personalized shortcuts, broadening application scope (e.g., gaming, accessibility).
- **Practicality:** A functional demo will showcase real-time gesture recognition and keystroke simulation, validated through accuracy metrics (>90% target) and latency tests (<0.5s delay).

Risks and Mitigation

- **Data Variability:** Augment dataset with synthetic hand poses and varying backgrounds.

- **Model Accuracy:** Experiment with multiple architectures (e.g., Random Forest vs. CNN) and hyperparameter tuning.
- **Real-Time Performance:** Optimize OpenPose parameters and use lightweight models.