

**Project Title:**

Real-Time Hand Gesture Recognition Using MediaPipe and Computer Vision Techniques

**Project Description:**

This project focuses on leveraging computer vision techniques to automatically detect and classify human hand gestures from images and real-time video streams. Using MediaPipe's hand landmark detection and OpenCV-based image processing, the system identifies hand poses and movements to interpret distinct gestures. We aim to extend existing implementations by adding and training the model on new gesture classes beyond those in the original dataset, improving recognition accuracy and flexibility for diverse interaction scenarios such as sign language, robotics, and AR/VR control.

**Computer Vision libraries/Prior code used:**

We will build upon the open-source repository [kinivi/hand-gesture-recognition-mediapipe](#), which integrates MediaPipe for hand landmark tracking and OpenCV for video processing. Our modification will include collecting or labeling new gesture samples and training the classifier on the extended dataset derived from the large-scale Hagrid dataset ([hukenovs/hagrid](#)). Additional preprocessing, data augmentation, and visualization will be implemented to improve model generalization and robustness.

**Literature Review:**

1. Zhang et al., "MediaPipe Hands: On-device Real-time Hand Tracking," *arXiv:2006.10214*, 2020.
2. Molchanov et al., "Hand Gesture Recognition with 3D Convolutional Neural Networks," *CVPR Workshops*, 2015.
3. Li et al., "A Review on Hand Gesture Recognition Using Deep Learning," *IEEE Access*, 2023.
4. Ahmed et al., "Vision-Based Dynamic Hand Gesture Recognition: A Survey," *Pattern Recognition Letters*, 2021.

**Example Code/ Tutorials used:**

- Example code: [kinivi/hand-gesture-recognition-mediapipe](#)
- Dataset: [hukenovs/hagrid](#)
- Tutorials and references:
  - Google *MediaPipe Hands* documentation and sample notebooks
  - OpenCV tutorials for preprocessing and visualization
  - CS231n project archives for baseline evaluation and reporting