



VIGNANA BHARATHI
Institute of Technology



(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

Real-Time Hand Gesture Recognition For Media control

PROJECT GUIDE:

Mrs.N.Sudha Rani
Assistant Professor,
CSE

Batch Number:21PA14

SEC: CSE-A

IV-I

TEAM MEMBERS:

B. Naveen	21P61A0517
B. Venkatesh	21P61A0523
B. Raju	21P61A0537

ABSTRACT

- This project implements a real-time hand gesture recognition system using Mediapipe and OpenCV. The system captures webcam input, detects hands, and recognizes predefined gestures. It utilizes the Mediapipe library to track hand landmarks and analyzes the relative positions of fingers for gesture recognition.
- Based on the recognized gesture, corresponding actions such as "play/pause," "volume up/down," or "skip forward/backward" are triggered using PyAutoGUI to simulate keyboard inputs. The application is intended for media control, offering a hands-free way to interact with media players.

KEYWORDS

- Real-time gesture recognition, Hand tracking, Media control, Mediapipe, OpenCV, PyAutoGUI, Webcam input, ,Landmark, detection, Python.

INTRODUCTION

- Real-time hand gesture recognition involves detecting and interpreting hand movements from live video input, allowing computers to recognize specific gestures in real-time.
- The technology enables human-computer interaction without the need for physical devices, using hand gestures to control systems or perform actions, such as media control or gaming.
- It is widely used in virtual reality (VR), robotics, smart devices, and assistive technologies to enhance user experience and improve accessibility.
- It allows users to interact with computers and devices in a natural, intuitive way using hand movements, reducing reliance on physical input devices like keyboards or remotes.
- Different hand movements or poses can be mapped to various functions (e.g., play/pause, volume control), allowing for customizable user interactions.

LITERATURE SURVEY

Topic	Key Contribution	Summary	Pros	Cons
CNN-based Hand Gesture Recognition (Zhang et al., 2018)	Introduced deep learning for gesture recognition using CNN.	This study utilized CNNs to recognize hand gestures from images, achieving high accuracy by learning from large datasets.	High accuracy in gesture recognition, suitable for complex gestures.	Requires large datasets for training and high computational power.
OpenCV for Real-time Hand Tracking (Kupyn et al., 2020)	Applied OpenCV for gesture recognition using hand contours and finger detection.	Demonstrated real-time hand gesture recognition using OpenCV by detecting contours and key finger points from video streams.	Lightweight and fast; OpenCV is widely available and well-documented.	Accuracy can drop with hand occlusion, varying lighting, and complex backgrounds.

Topic	Key Contribution	Summary	Pros	Cons
Mediapipe for Hand Landmark Detection (Zhang et al., 2020)	Used the Mediapipe framework for real-time hand tracking and gesture recognition.	This paper showcased the use of the Mediapipe framework for detecting hand landmarks in real-time.	High precision and efficiency, easy integration with real-time applications.	Limited flexibility in customizing gesture definitions for specific use cases.
Gesture-controlled Media Applications with PyAutoGUI (Yang et al., 2019)	Utilized PyAutoGUI for gesture-based media control.	The study explored gesture-controlled media applications using PyAutoGUI.	Easy to implement and use for simple media control tasks.	Limited to predefined media control actions; not suitable for more complex applications.
Landmark-based Gesture Recognition (Alrashdi et al., 2020)	Improved accuracy in gesture recognition using hand landmark detection.	Showed how using hand landmarks, such as finger joints, enhances the precision of real-time gesture	High accuracy in detecting fine and complex hand gestures.	Hand occlusion and complex backgrounds may affect recognition.

PROBLEM STATEMENT

- The current methods of media control, such as using physical buttons or remotes, limit hands-free interaction and can be inconvenient in certain situations. There is a need for a real-time, intuitive, and non-invasive system that can recognize hand gestures and translate them into media control commands, enabling users to interact with devices more naturally and efficiently.

OBJECTIVE

- To develop a real-time hand gesture recognition system using OpenCV, Mediapipe, and PyAutoGUI that enables users to control media playback (e.g., play/pause, volume control, track navigation) through intuitive hand gestures, offering a hands-free and seamless user experience.
- Additionally, to ensure accurate gesture recognition under various lighting conditions and hand positions, and to create an easily customizable system for different gesture sets and media controls.

SOFTWARE AND HARDWARE REQUIREMENTS

Software Requirements:

- **Python:** Version 3.6 or higher for running the script and managing dependencies.
- **OpenCV:** Version 4.x or higher for capturing and processing video frames from the webcam.
- **Mediapipe:** Version 0.8.x or higher for hand tracking and landmark detection.
- **PyAutoGUI:** Version 0.9.x or higher for automating media control commands.

Hardware Requirements:

- **Webcam:** A standard webcam with a resolution of at least 640x480 pixels for capturing live video feed.
- **Computer:** A system with a modern processor (e.g., Intel Core i5 or equivalent) and at least 4 GB of RAM to handle real-time video processing and gesture recognition.
- **Display Monitor:** To display the video feed and any user interface elements.

PROPOSED METHODOLOGY

- **Gesture Recognition with Mediapipe:** Utilize Mediapipe's Hands module to detect and track hand landmarks in real-time from the webcam feed. Mediapipe provides a comprehensive set of hand landmarks that represent key points on the hand, which are used to identify different gestures.
- **Gesture Interpretation:** Develop an algorithm to interpret specific hand gestures based on the relative positions of detected landmarks. The algorithm will map predefined gestures to corresponding media control commands such as play/pause, volume up, and skip forward.
- **Media Control Integration:** Integrate the gesture recognition output with PyAutoGUI to execute media control commands based on recognized gestures. This allows for direct interaction with media players or applications using the identified gestures.

MODULES

1. **Video Capture Module:** Uses OpenCV to capture real-time video from the webcam. Preprocesses frames to prepare them for hand detection.
2. **Hand Detection Module:** Employs Mediapipe to detect and track hands in each video frame. Extracts key hand landmarks, providing coordinates for gesture recognition.
3. **Gesture Recognition Module:** Analyzes hand landmarks to identify specific gestures. Uses conditions to map gestures to media control actions (e.g., play, pause, volume adjustment).
4. **Media Control Module:** Uses PyAutoGUI to perform actions like play/pause, volume up/down. Executes commands based on recognized gestures, allowing real-time media control.
5. **User Interface Module:** Displays feedback or detected gesture information to the user. Shows current gesture or media action, improving user interaction.
6. **Testing and Calibration Module:** Adjusts landmark thresholds and gesture mappings for accuracy. Ensures reliable performance across different lighting and hand positions.

IMPLEMENTATION

Import Libraries and Dependencies

OpenCV: For capturing and processing video frames.

Mediapipe: For hand landmark detection.

PyAutoGUI: To control media (play, pause, volume) based on gestures.

Capture and Process Video Frames

1. Use OpenCV to access the webcam and capture video frames in real-time.
2. Convert frames to RGB format for compatibility with Mediapipe.

Hand Detection and Landmark Extraction

1. Load Mediapipe's hand detection model.
2. Detect hands in each frame and extract key landmarks (e.g., fingertips and knuckles).

Gesture Recognition

1. Define gestures based on the position of landmarks (e.g., a specific finger up for play/pause).
2. Use conditions or algorithms to map landmark positions to predefined gestures.

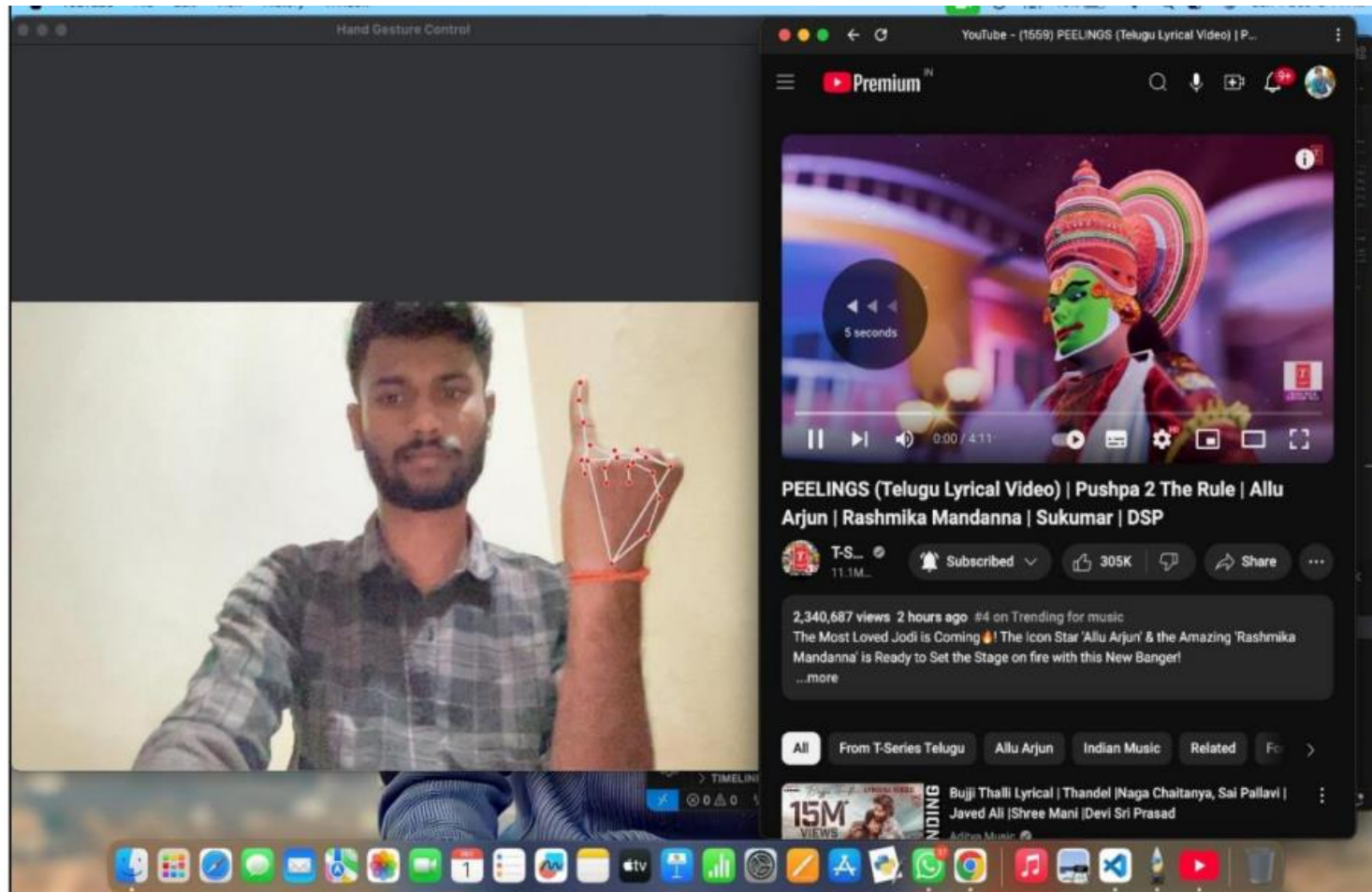
Media Control with PyAutoGUI

1. Assign gestures to media control functions:
 1. **Play/Pause:** Gesture A (e.g., one finger up).
 2. **Volume Up/Down:** Gesture B and C (e.g., thumbs up/down).
2. Use PyAutoGUI to execute corresponding media actions.

Testing and Optimization

1. Test with different lighting and hand positions for reliability.
2. Adjust landmark thresholds to improve gesture recognition accuracy.

RESULTS



REFERENCES

- <https://www.sciencedirect.com/science/article/pii/S1877056818303032>
- <https://arxiv.org/abs/2006.10560>
- <https://arxiv.org/abs/2003.10423>
- <https://www.sciencedirect.com/science/article/pii/S1877056819302224>
- <https://ieeexplore.ieee.org/document/9350218>
- <https://ieeexplore.ieee.org/document/9145207>
- Project Tunnel



**THANK
YOU**

A hand in a dark suit sleeve holds a glowing, translucent sphere. Inside the sphere, the words "THANK YOU" are written in a bold, white, sans-serif font. The sphere is surrounded by a complex, glowing blue network of lines, resembling a neural network or a digital web. The background is dark and textured with similar blue glowing lines.