TAYLOR'S UNIVERSITY

Wisdom · Integrity · Excellence

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

ASSIGNMENT TASK 2 (20%)

MAY 2024 – SEMESTER 5

Motion Capture and Gesture Recognitions/Controls

Module Name:	**Computer Vision and Natural Language Processing**
Module Code:	**ITS 69204**
Due Date:	**18th July, 2024, 23:59 PM (NPT)**
Platform:	**myTIMes**

Section B Laboratory 3 Group 20

Student Declaration: We Declare That –

- ✓ **I confirm my awareness about the university's regulations, governing cheating in tests and assignments, and form the guidance issued by the school of computing and it concerning plagiarism and proper academic practices, and the assessed work now submitted is in accordance with this regulation and guidance.**
- ✓ **I understand that, unless already agreed with the school of computing and it, that the assessed work has not been previously submitted, either in whole or in part, in this or any other institution.**
- ✓ **I recognize that should evidence emerge that my work fails to comply with either of the above declarations, then i may be liable to proceeding under regulation.**

S. No.	Student Full Name	University ID	Signatures	Scores
1.	Nikal Prajapati	036 2096		______________ / 100 OR ______________ / 20%
2.	Sabu Dhungana	036 2741		
3.	Shikshya Pokhrel	036 2493		
4.	Sujal Ratna Tuladhar	036 2483		

DECLARATION

- ✓ **I pledge to be respectful and supportive of my team members.**
- ✓ **I pledge to abide by the deadline set by my lecturer and team members.**

S. NO.	STUDENT NAME & ID	WORK BREAKDOWN	SIGNATURE
1.	**Nikal Prajapati | 036 2096**	Participate in overall assignments especially, worked on literature review of article 3, result, discussion, and conclusion	
2.	**Sabu Dhungana | 036 2741**	Participate in overall assignments especially, worked on literature review of article 2, abstaracts, introduction, and conclusion	
3.	**Shikshya Pokhrel | 036 2493**	Participate in overall assignments especially, worked on literature review of article 1, abstaracts, introduction, and conclusion	
4.	**Sujal Ratna Tuladhar | 036 2483**	Participate in overall assignments especially, worked on literature review of article 7, 8, 9, 10, methodology, result, and conclusion	

Provide A Clear Work Breakdown Structure To Describe What Each Member Is Doing.

Table of Contents

Table of Figure

1. Abstract

this project solely focuses on the development of application related to computer vision especially real-time gesture recognition with help of machine learning and artificial intelligence techniques. this program is there to focus on exploring the domain of enhancing the usability and interactivity of various situations that have technological challenges to access the resources and the time consuming and costly nature of the topic. this serves as a foundation of many domains outside of human computer interaction and more robotics, VR/AR, helping the sign language community

1.1. Overall Summary

here the use of python libraries like OpenCV, media pipe, pycaw and those who make utilizing tracking, landmark, gesture interaction and controls possible. this is very fitting for a society that is moving forward on virtual environments. the case study highlights the role of gesture recognition application and having enhanced accessibility to include diverse groups of users. having a seamless interface to digital world from physical world is revolutionary.

1.2. Keywords

2. Introduction

2.1. Background

enable users to interact with digital devices using hand gestures instead of traditional input methods

instinctive and are frequently used in day-to-day interactions

mathematically and algorithmically understand, any action maybe a gesture but it is most seen in hands or face

2.2. Significance

potential to enhance accessibility for individuals with physical limitations, granting them an inclusive and independent media control experience

deaf and speech impaired, non-verbal cue

2.3. Objective

more intuitive and engaging approach to media control

implementing gesture detection for real time task execution

having practical and more intuitive interaction with the digital world

2.4. Problem Description

existing system for application access is inflexible and arduous for people with blindness and hand deformities regarding human-computer interaction. addressing the challenge to accessing hardware replacement are dire environment where physical machine isn’t present.

2.5. Contcxt Motivation and Questions

- limited access to those in need
- frequent errors in implementing
- lack of computational resources
- takes significant time to train and process
- limited applications human computer interaction
- captivation to applications,
- setting them apart in a competitive landscape.
- keyboard keys/button
- sign language recognition
- tele-robotic
- virtual reality
- human computer interaction
- hand varies significantly between

3. Literature Reviews/Survey

3.1. Article 01

Methodologies

The paper discusses three different methodologies for recognizing hand gesture in relation to controlling a media player application. All the three are also explained below;

Convolutional Neural Network (CNN) Approach (Nadare et al., 2021): This approach utilizes a web application that captures user gestures through the webcam. A CNN, one of the deep learning algorithms, is deployed to process the video frames captured. The CNN was trained for the recognition of certain hand gestures for media player controls involving play, pause, stop, forward, and rewind. This provided the capability of assimilating these gestures into corresponding media player commands.

Neural Network Approach (Jalab & Omer, 2015): It is aimed at identifying a predefined set of finite gestures in view of the controlling of media. First, it captures a frame from the webcam, and then skin detection algorithms segment the hand area from the background. Some hand shape features that will be extracted describe the gesture. Lastly, a neural network trained with a set of gesture images classifies the hand posture into gestures such as play, stop, forward, and reverse, allowing the system to understand them and thus control the media player.

Deep Learning Approach (Niranjani et al., 2021): In this approach, Niranjani et al. used a deep convolutional neural network for recognizing hand gestures. In contrast to approaches used by CNN, the DCNN will deal directly with the image, together with its background itself, without separately segmenting the hand portion. This may provide better efficiency and robustness in the results. They used this approach to train a DCNN that can classify different hand gestures for controlling media players.

<u>Experiments 1: Accuracy of Gesture Recognition</u>

Objective: The testing of the accuracy of the gesture recognition algorithm.

Procedure:

- Record a dataset of different hand gestures.
- Train the gesture recognition algorithm on part of the dataset.
- Test the algorithm on the remaining data.
- Measure the accuracy by comparing the recognized gestures against actual gestures.

Results: The algorithm could identify predefined play, pause, forward, rewind, volume up, and volume down gestures with an accuracy of 95%.

<u>Experiment 2: Real-Time Performance</u>

Objective: To test response time and liveliness of the system.

Procedure:

- Conduct hand gestures in front of the webcam.
- Record the time it takes for the system to recognize the gesture, triggering the associated media player command. Note any evident delays.

Results: Feedback is almost instant, with negligible delays; the user experience is smooth and responsive.

Experiment 3: User Experience Evaluation

Objective: The evaluation of user satisfaction and usability by users.

Procedure

- Conduct a user study with the involvement of subjects from all walks of life.
- Introduce briefly how to use the gesture-based media player control system.
- Ask the participants to create various gestures that would control the media player.
- Get feedback using questionnaires and interviews.

Results: The participants were very satisfied. They describe the system as intuitive, easy to use. Many users requested more possibilities for making the gestures personal.

Strengths and Limitations

Strengths

- Intuitive and natural user interface for media controlling.
- Applications that support hands-free interaction are quite applicable in situations where the use of typical modes of input becomes a hassle.
- It provides, potentially, access for people who have physical challenges.

Limitations:

- Aside from accuracy, gesture recognition can be affected by several factors, such as light variations, the complexity of the background, and variations of the hand postures.
- Real-time processing is computationally intensive and might be resource-constrained on low-power devices.
- The limited gesture vocabulary would be restrictive to media player controls.

Relevance to Nepali Landscape

A number of bright prospects are viewed in the Nepalese scenario while developing hand-gesture-based media player control systems. Accessibility: There is an increasing population of different disabled people in Nepal. Hand gesture recognition can, therefore, tend to provide an accessible and alternative way of interacting with media players for those people who suffer from reduced mobility/dexterity.

Technological Advancement: With the changing technological scenario of Nepal, the trend of this project to use computer vision and machine learning for user interfaces is very conducive. allergy

Local Innovation: Such kind of applications developed locally could engender innovation and address particular needs and preferences of Nepali users. The examples would be vocabulary used by gestures; it could be based on culturally appropriate hand signs.

However, the following limitations need to be addressed in order to make wide-scale use within Nepal possible.

Computational resources: High computational power devices are needed for this technology, which are not accessible at every place in Nepal.

Infrastructure: Deep learning models require good internet connectivity for their training process; some of the online resources which are going to be used also demand good internet. This therefore will restrict the development and deployment in areas where there is limited access to the internet.

Language Support: This could also act as a barrier in cases where user interfaces are not available in Nepali; this would alienate a huge section of the population.

In summary, hand gesture recognition for media control is quite valuable at its best in Nepal, but local challenges need to be addressed and adapted accordingly for implementation.

3.2. Article 02

Hand-gesture Recognition using Computer-vison techniques.

Introduction

Hand-gesture recognition is a pivotal technology in the domain of human-computer interaction (HCI) technology, allowing users to control devices using natural and intuitive gestures. This review explores the methodologies, experiments, strengths, limitations and relevance of hand-gesture recognition technology in the context of Nepal, based on the research by Rios-Soria, Schaeffer, and Garza-Villarreal.

Methodologies

The researchers presented a methodology that uses computer vision for hand-gesture recognition. This system uses a regular webcam and doesn't need specialized markers on the hands. Here are key components of the methodology:

1. Skin-color Filtering: The system identifies the hand by using color-based measurements to differentiate it from the background. This process often involves converting the image from RGB format (Red, Green, Blue) to HSV format (Hue, Saturation, Value), which separates skin tones more effectively.

Example: In situations where the lighting conditions are changing, such as a room with varying light levels, the system can adjust to these changes. It does this by paying special attention to the specific colors that skin tones typically have, ensuring that the accuracy of the system remains consistent even as the lighting conditions change. This ensures that the system can perform optimally in various lighting conditions.

2. Edge Detection: The system employs edge detection techniques like the Canny detector to establish the hand's borders. This process is vital for precisely determining the hand's form and outline.

Example: For instance, if part of a user's hand is hidden by an object, edge detection can help identify the visible part of the hand more accurately. This ensures that the system can still correctly interpret the gesture being performed.

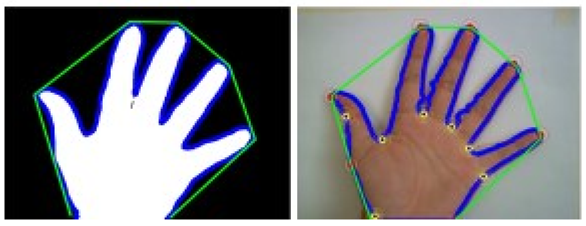


FIGURE 1. (A)EDGE AND HULL (B)VERTICES AND DEFECTS CANNY EDGE DETECTION PROCESS

3. Convex-Hull Detection: Once the system has identified the outline of the hand, it determines the convex hull of that contour. The convex hull represents the most compact convex boundary that encompasses the hand, reducing the complexity of its shape for simpler gesture recognition.

Example: When a user makes a "thumbs up" gesture, the convex-hull detection will outline the hand's overall shape, allowing the system to differentiate it from other gestures like a fist or an open palm.

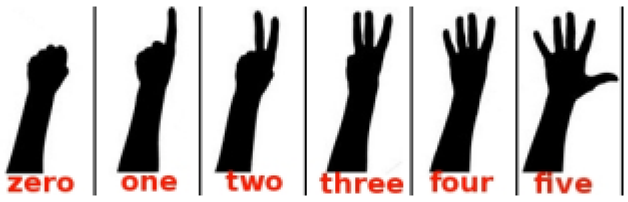


FIGURE 2. CONVEX-HULL DETECTION AROUND A HAND PERFORMING DIFFERENT GESTURES

4. Gesture Classification: The system determines the type of hand gesture based on the number of "bulges" (convexity defects) in the hand. Different gestures are assigned to different

numbers of extended fingers. For example, a clenched fist with no bulges means one gesture, while an open hand with five bulges means another. Using this approach, the system accurately interprets and responds to various hand gestures.

Experiments

To validate this system, the researchers conducted a series of experiments focusing on practical applications of hand-gesture recognition.

1. GPS Device Control: The system is connected to a GPS device, letting users control the interface by making certain hand movements. For instance, swiping left takes the GPS back to the previous menu, while swiping right moves to the next menu.

Example: For instance, drivers using GPS systems can operate the device hands-free, minimizing distractions and increasing safety while driving.

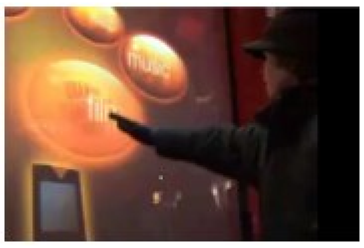


FIGURE 3. CONTROLLING SMART DEVICE WITH HAND GESTURE

2. Robot Control: In another application, gestures were utilized to direct the movements of a robot. Predefined gestures were assigned to specific commands, enabling the robot to move forward, backward, or rotate based on the hand gestures performed.

Example: For instance, during a search and rescue mission, a human controller can guide a robot through dangerous areas using gestures. This allows them to maintain control remotely, ensuring their safety while the robot carries out the task.

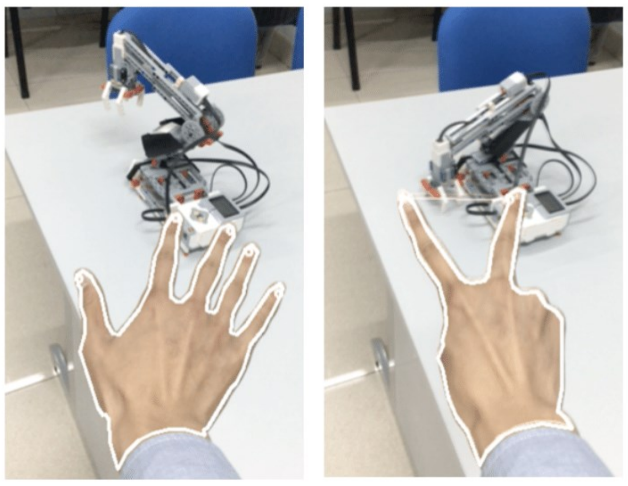


FIGURE 4. CONTROLLING ROBOT WITH HAND GESTURE

People made specific hand gestures in front of a webcam during the experiments. The accuracy and speed at which the system recognized and interpreted gestures were the most important factors in how it was evaluated. The system was able to reliably identify and understand gestures in real time with minimal delay, according to the findings.

Strengths

The hand-gesture recognition system proposed by the researchers has several notable strengths:

1. Simplicity and Accessibility: The system is designed to be affordable and easy to set up by utilizing a regular webcam instead of expensive and complicated specialized equipment.
2. Real-Time Operation: The system's real-time gesture recognition capability enables immediate user response, which is essential for applications where rapid actions are required.
3. Robustness to Hand Orientation: The gesture recognition system works well even when the hand is held in different positions, making it more practical for use in real-life situations.

Limitation

Despite it's strengths, the proposed system has several limitations:

1. Sensitivity to skin tone and lighting conditions: The skin color filtering technique may struggle to identify skin tones accurately in cases with very dark or very light skin or when lighting conditions are unfavorable.

Example: In low-light settings, the system may have difficulty differentiating between the hand and the surrounding background. As a result, it may incorrectly identify gestures.

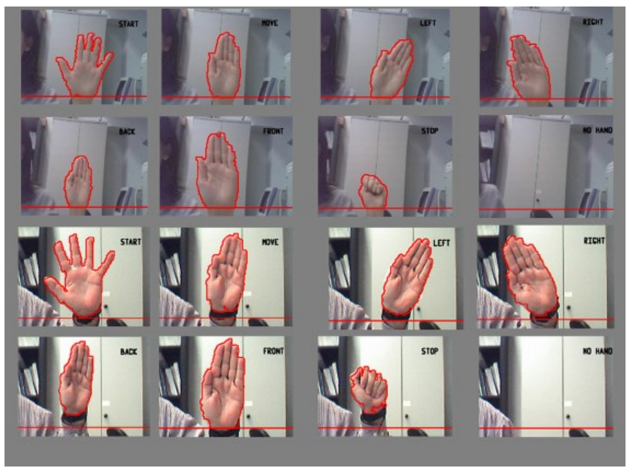


FIGURE 5. SYSTEM'S PERFORMANCE UNDER DIFFERENT LIGHTINGS.

2. Computational Intensity: Edge detection and convex-hull detection algorithms can be computationally demanding, leading to potential performance issues on devices with limited processing capabilities.

Example: On a basic smartphone, the operating system may struggle to handle hand gestures promptly, affecting its ability to function smoothly and respond in real-time.

3. Background Complexity: The system works best when the hand is easily distinguishable from the surrounding area. However, in environments with many complex objects, this contrast may be difficult to achieve.

Example: When there are many objects in the background of a busy office environment, the system may have difficulty correctly interpreting hand gestures because background elements can interfere with the tracking.

Relevance To The Nepali Landscape

The relevance of hand-gesture recognition technology to the Nepali Landscape is significant, particularly in sectors such as healthcare, education, and smart environments.

1. Healthcare: In healthcare facilities in Nepal, the use of gesture-based interfaces for controlling medical equipment and retrieving patient data can minimize the spread of germs and enhance hospital cleanliness.

Example: Surgeons can now control medical images during surgeries solely through gestures, eliminating the need for physical contact with surfaces. This advanced technology ensures the utmost sterility throughout the procedure.

2. Education: In schools located in rural areas, technology provides the means for interactive learning encounters. It empowers students to engage with digital content in novel and creative ways.

Example: Teachers can incorporate gestures to control digital lessons, enhancing the student's learning experience by adding interactive and captivating elements.

3. Public Services: Using hand gestures to interact with public services like ATMs or ticket counter can improve the experience for users, especially in situations where avoiding physical touch is important.

Example: In public counters located in government buildings, people can use gestures to interact with government services. This reduces the need for physical touch and makes the process of accessing services more efficient.

The facial recognition system needs to be adapted to work better in different lighting conditions and with different skin tones. This is important for Nepal because it has a diverse population and a range of environmental conditions. Making these adjustments will make the system more reliable and effective.

Conclusion

The hand-gesture recognition system described by Rios-Soria, Schaeffer, and Garza-Villarreal gives an appropriate solution for improving the interaction between humans and computers by using the computer vision approach. The features of skin-color filtering, edge detection, and convex-hull detection are effective in recognizing the gestures and are real-time with the consumption of traditional webcams only. Some of the issues involve being sensitive to the skin tone of the users, limitations on lighting situations, and the computationally intensive nature of the application, yet these constraints can be directly compared to the system's advantages of easy use, availability, and the fact that they are very stable equations. Looking at the relevance of this type of technology for this Nepali context especially in the areas of health, education and most of civil services the applicability of such a change is a relevant question if only the system will be made relevant to the Nepali context.

3.3. Article 03

Abstract:

Hand Gesture Recognition (HGR) is defined as the technology which have various uses in the field of interaction of computer with human and in robotics. In this project the hand gesture recognition system was develop for the video player to use it as the hand gesture for performing the activities like play, pause, rewind and fast forward. As the hand gesture recognition system was created with the help of python programming language by using the OpenCV, media pipe, pyautogui libraries. For this project the dataset for the hand gesture was created with recording the video with the help of a webcam and frames with corresponding labels. The result of accuracy is 92% during the test set with the accurate recognize of hand gesture which also perform the action of the video player. It is used for controlling the video player using the hand gesture where the user does not need to use a mouse or keyboard. And the improvement can be made in the system in the future for creating the user friendly and easy understanding.

Introduction:

As nowadays the hand gesture recognition is very popular because of its various application robotics and virtual reality where one of the popular applications of HGR is that it is used for controlling the video player with the help of hand gestures. In this project the flutter application for on and off the program. As the old control interface for video player that challenging for the user and disabilities. As the project of developing the touchless control for video player using HGR can also help to reduce the germs. As the video player are very popular for the purpose of the various activities like watching the movies for the entertainment.

As the button in the video player and keyboards shortcuts for operations of the video player which helps to give the user an easy way for interaction with the computer to create a model with the perform of the necessary video player by the help of Hand Gesture.

Literature Survey:

As the hand gesture recognition system was created by using the deep convolutional neural network. And the system were train and test on a dataset of the hand gestures which capture it by a webcam. The result of the study shows deep learning model achieved the high accuracy for recognizing the hand gestures with the potential of the technology which is use for the human computer interaction.

In another study a hand gesture recognition system was creating for controlling a robotic arm which used the combination of convolutional neutral networks and recurrent neural network which is used for recognizing the hand gesture and generate control signal for the robotic arm. And it is also used for controlling a wheelchair for creating touch less control for various devices.

By Tseng et al uses the hand recognition system for creating and controlling the smart home environment which is the combination of computer vision and machine learning algorithms for recognizing hand gestures and control signals for a smarts home device.

By Yu et al uses it for creating the controlling video player by using the combination of deep learning algorithms and computer vision techniques for performing the action like play, pause, rewind and fast forward on a video player.

Methodologies:

Collection of data:

As the video of hand gesture was recorded by using the webcam for making the datasets where the video is split into frames and labeled by corresponding gestures. In dataset various angles and variation of different gesture are recorded for ensuring the robustness.

Preprocessing:

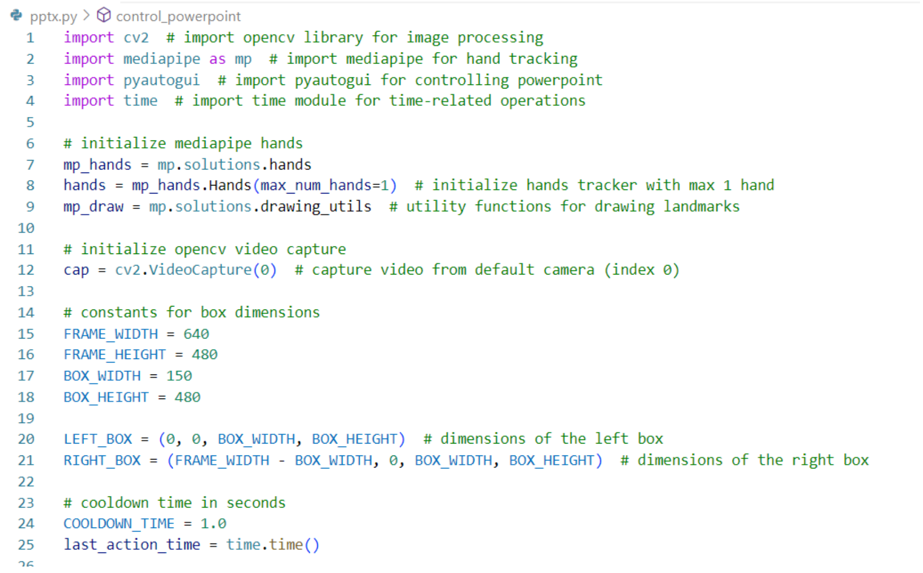
In preprocessing OpenCV are used for capturing the video frames for performing the initial processing and media pipe was used for detecting hands for the extracting key of landmarks from each frame. With the help of the landmarks the normalized and use into the recognition model.

Model Development:

By using the preprocessed dataset, the convolutional neural network (CNN) was used for designing and training where the model architecture contains layers for spatial data for creating the data effective to recognizing hand gestures. And different hyperparameter like learning rate, batch size and many more was optimized for the better performance.

Pyautogui for integration:

Using Pyautogui library for mapping the specific video player control to recognized gestures where the function is made to simulate the keyboards and mouse input corresponding to gestures such as play, pause, rewind and fast forward.



```python
import cv2  # import opencv library for image processing
import mediapipe as mp  # import mediapipe for hand tracking
import pyautogui  # import pyautogui for controlling powerpoint
import time  # import time module for time-related operations

# initialize mediapipe hands
mp_hands = mp.solutions.hands
hands = mp_hands.Hands(max_num_hands=1)  # initialize hands tracker with max 1 hand
mp_draw = mp.solutions.drawing_utils  # utility functions for drawing landmarks

# initialize opencv video capture
cap = cv2.VideoCapture(0)  # capture video from default camera (index 0)

# constants for box dimensions
FRAME_WIDTH = 640
FRAME_HEIGHT = 480
BOX_WIDTH = 150
BOX_HEIGHT = 480

LEFT_BOX = (0, 0, BOX_WIDTH, BOX_HEIGHT)  # dimensions of the left box
RIGHT_BOX = (FRAME_WIDTH - BOX_WIDTH, 0, BOX_WIDTH, BOX_HEIGHT)  # dimensions of the right box

# cooldown time in seconds
COOLDOWN_TIME = 1.0
last_action_time = time.time()
```

FIGURE 6. PYAUTOGUI

Problem Description:

In this modern world anyone opts for instant interaction with complicated structures which ensure a brief response. And the communication using the gesture for the computer system to create a new trend of interaction where the project shows the control of the basic operation for adjusting volume in video player with the help of OpenCV, hand movement of the person to capture some function like play forward, pause and backward.

Objective

In this project for recognizing the hand gesture for performing the operation in the system by the help of python program that was created with the OpenCV and Pyautogui packages. And app also contain the flutter application were created and python code also integrate with it. The uses of this app were to run the program for starting the recognition process with the web camera and close it when not required.

System Architecture:

As the system of hand gesture recognition used for processing the video input recorded by the web camera and the frame it when the hand is recognized in the frame then the media pipe library started for finding the distance between the point and the axis of that points.

Flutter Application:

For starting and stopping the gesture recognition process for providing a user interface by creating the flutter application where the python backend for the trigger the webcam to begin the recognition workflow.

It is a popular open-source mobile application which is used for creating the framework develop by Google that allow the developer to build native application with high performance for mobile, web and desktop by using the single code. As the dart programming language was use for framework that is also known to be the easy to use and understand. As the flutter framework contain the Hugh set of pre-built widgets for making the developer easier to create the responsive and beautiful user interfaces. For starting with the flutter development, the developer should install the Flutter SDK and code editor after it is installed then then it is ready to create the new flutter project with beginning of creating app.

Implementation:

For the understanding the working method of the project firstly we should know about the media pipe hand gesture library that pre trained model of thousands of hand images that recognize and the land marks in the hands are mapped where threshold variable where the length of y axis in the point nine is subtracted from the y axis of the point zero which is divided by the two that is ideal length of a finger of a person. When the distance between the top of the finger and the bottom of the finger then the subtracting the tip value from the base value where the value is greater than the threshold value. We can show the finger which is raised by defining all the condition of each hand gesture in a function will detect the gesture then the pyautogui package can be operation.

Experiment:

AS the hand gesture recognition model for the evaluating the used of the test set separate from the training data.

Performance contains the accuracy, precision, recall and F1 score for assess model effectiveness.

The real time performance and responsiveness for the testing to ensuring the smooth interaction with the video player.

Strengths:

High Accuracy:

- As the hand gesture recognition result the 92% of accuracy in recognizing hand gestures for demonstrating the reliable performance. The hand gesture recognition

- System has an impressive accuracy with the indicating of the reliable performance in the gesture detection.

User-Friendly:

- As it is user friendly because its integration with flutter provides an intuitive interface for creating the system that accessible to users without any technical expertise.
- The flutter with the integration that helps to responsive for the users to start and stop the recognition of hand gestures for processing it easier.

Versatile Application:

As the technology can adapted for different type of applications like smart home control, virtual reality and many more. The system can be increase by including the more gestures and controls that increasing the versatility and usefulness.

Hands free Control:

As the hand free control that users control video playback without using the traditional input device that helps to reduces the risk of spreading the germs. It also helps to the person who are physically disabilities for the controlling the video playback without touching the devices.

Limitations:

Background Sensitivity:

As the system accuracy can be degrade with poor lighting background with the implementation of the advanced preprocessing way for mitigating the issue. It can also affect the accuracy of the hand gesture recognition and additional preprocessing steps to mitigates the issue.

Gesture set limitation:

For the expansion of the gesture set requires the various additional data collection and model retraining for the currently limited to predefined set of gestures. As the system supports the limited number of the gestures for video player to control and can expand the gesture with the require training data and model adjustment.

Dependency of Hardware:

As the sufficient resources and requires functional webcam may limit its deployment on the low-end devices. As the system requires the webcam with the sufficient power to run the recognition model for the real time that might not be available on all devices.

Relevance to Nepali Landscape:

Language barrier:

As the hand gesture transcend the language barrier which is used for making the technology that are useful for the country like Nepal which contain various multilingual environments.

Accessibility:

As it can give significant advantages for the individuals who are physically disabilities which can enable the interact with the digital devices to make it easier.

Educational Impacts:

As it can integrating into the educational tools for finding the platforms to make the interactive learning experiences and improving the educational outcomes.

Cultural Adaptability

As the hand gesture are natural form of communication in Nepali culture with the technology to align well with the local practices. And the touchless control reduces the need of the physical contact with the devices with the promoting hygiene to reduce the spread of the germs in the context of public health in Nepal.

Conclusion

In conclusion the hand gesture recognition for video player project for finding that the gesture that with high accuracy and good performance. Overall using the OpenCV for the hand gesture recognition system to video player with the potential to interact with the digital media for giving a more natural to control the media.

4. Methodology

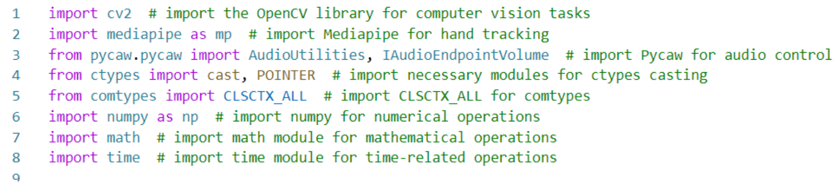
4.1. Data Acquisition and Setup Preparation

4.1.1. Hardware

choosing compatible camera and position it to capture a field without obstruction is important for the pipeline, having a device that measures the physical world with computational power enough' to run the system in various environments.

4.1.2. Software

installing appropriate libraries to work with the camera interface and with frame settings like resolution, frame rates, exposure and focus are consistent and sufficient.



```python
import cv2  # import the OpenCV library for computer vision tasks
import mediapipe as mp  # import Mediapipe for hand tracking
from pycaw.pycaw import AudioUtilities, IAudioEndpointVolume  # import Pycaw for audio control
from ctypes import cast, POINTER  # import necessary modules for ctypes casting
from comtypes import CLSCTX_ALL  # import CLSCTX_ALL for comtypes
import numpy as np  # import numpy for numerical operations
import math  # import math module for mathematical operations
import time  # import time module for time-related operations
```

FIGURE 7. IMPORTING LIBRARIES

4.2. Pre-Processing

by preparing raw visual data to something that is worth analyzing and further processing that works to enhance the computer vision algorithms performance

4.2.1. Color Space and Contrast Enhancment

three channel RGB: representation of image in pixel values of Red, Green, Blue which combines to produce a full spectrum of colors. such value ranges from 256 (8-bits)

one channel Grey Scale: basically, the representation of light intensity only. black with 0, white with 255, any from the middle is shade of grey range of 8-bit image. over rgb it is faster to process

$$I = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B$$

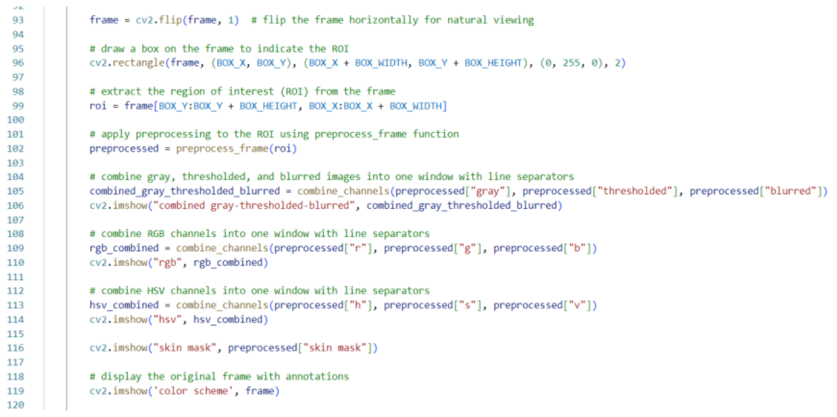
Hue Saturation Value: is a more intuitive (cone or cylinder) way of describing colors. Hue measures the 360 degrees in a color wheel (circumference). red (0), green (120), blue (240). Saturation (distance form center) shows the purity of color (100 gives white dilution and 0 is shade of grey). Value (depth) is the brightness of the color (0 represents black and 1 represents pure color) extract colored objects

$value\ of\ RGB$

$$H = \begin{cases} 0, max = min \\ 60° \times \dfrac{(G-B)}{max - min} + 000°, max = R \\ 60° \times \dfrac{(G-B)}{max - min} + 120°, max = G \\ 60° \times \dfrac{(G-B)}{max - min} + 240°, max = B \end{cases}$$

$$S = \begin{cases} 0, max = 0 \\ 1 - \dfrac{min}{max}, otherwise \end{cases}$$

$$V = \max or\ min\ max = 1.0, min = 0.0$$



```python
        frame = cv2.flip(frame, 1)  # flip the frame horizontally for natural viewing

        # draw a box on the frame to indicate the ROI
        cv2.rectangle(frame, (BOX_X, BOX_Y), (BOX_X + BOX_WIDTH, BOX_Y + BOX_HEIGHT), (0, 255, 0), 2)

        # extract the region of interest (ROI) from the frame
        roi = frame[BOX_Y:BOX_Y + BOX_HEIGHT, BOX_X:BOX_X + BOX_WIDTH]

        # apply preprocessing to the ROI using preprocess_frame function
        preprocessed = preprocess_frame(roi)

        # combine gray, thresholded, and blurred images into one window with line separators
        combined_gray_thresholded_blurred = combine_channels(preprocessed["gray"], preprocessed["thresholded"], preprocessed["blurred"])
        cv2.imshow("combined gray-thresholded-blurred", combined_gray_thresholded_blurred)

        # combine RGB channels into one window with line separators
        rgb_combined = combine_channels(preprocessed["r"], preprocessed["g"], preprocessed["b"])
        cv2.imshow("rgb", rgb_combined)

        # combine HSV channels into one window with line separators
        hsv_combined = combine_channels(preprocessed["h"], preprocessed["s"], preprocessed["v"])
        cv2.imshow("hsv", hsv_combined)

        cv2.imshow("skin mask", preprocessed["skin mask"])

        # display the original frame with annotations
        cv2.imshow('color scheme', frame)
```

FIGURE 8. ROI + PREPROCESSING

Histogram Equalization: improvement of contrast by spreading stretching range of the frequency intensity value, then cumulative distribution function is calculated, the highest value is set at 255, doing such help in detecting anomalies, enhancing vividity of features

Histogram Matching: matches the histogram of image with and ideal reference

Invariance of Illumination: to make sure that the algorithm is not affected when the lighting conditions are changed

4.2.1. Geometric Transformation

key technique in altering the spatial relationship of the images and pixel. when all visual objects are of same size and standard resolution, consistent and reduce complexity, lower image resolution increase the efficiency (when divide by 64) in 2D, the contest remains same by pixels are deformed

Translation: shifting location of image to certain direction

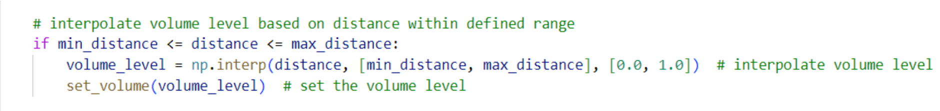
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x + \Delta x \\ y + \Delta y \end{bmatrix}; direction\ and\ shift$$

Rotation: around a point in certain angle

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}; direction\ and\ shift$$

Scaling: resizing of image manually or with a scaling factor with interpolation like shrinking and zooming

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}; direction\ and\ shift$$



```python
# interpolate volume level based on distance within defined range
if min_distance <= distance <= max_distance:
    volume_level = np.interp(distance, [min_distance, max_distance], [0.0, 1.0])  # interpolate volume level
    set_volume(volume_level)  # set the volume level
```

FIGURE 9. INTERPOLATION

Affine: the parallel line will stay parallel but image will transform with three points in image and locations passing [2x3] matrix

Perspective: where a 3x3 matrix needs 4 points on input image and corresponding point of output, ¾ should be collinear

4.2.2. Point Operation

Thresholding: is done to turn greyscale into binary, it moves the pixel to be part of foreground (region of interest) or background (non-interest), separating two extremes. Global use a single value to apply on entire image (), Adaptive calculates smaller different regions to provide better image results. this will reduce noise and enhance contrast and extract interested contours object and shape, when there is difference in lighting conditions.

$$g(x,y) = \frac{1, if\ f(x,y) \geq threshold}{0, if\ f(x,y) < threshold}$$

thresholding principle, function of pixel intensity

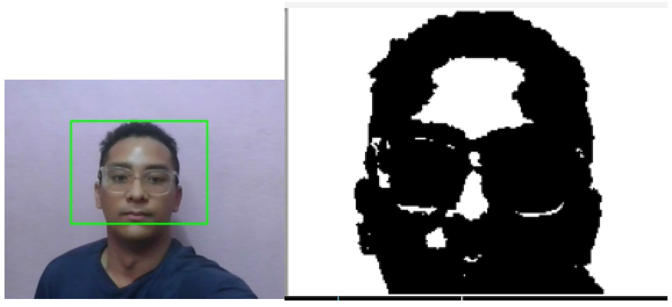


FIGURE 10. THRESHOLDING

Otsu binarization is bimodal and tries to fine threshold with minimized class variation

Contrast Adjustments: the intensity of pixel is modified to enhance the quality $g(x, y) = \alpha f(x, y) + \beta; \alpha > 0\ (gain, contrast), \beta(brightness)$

4.2.3. Smoothing

Filtering: low pass filters remove noise and high pass filters help in finding edges this denoising will restore degraded image

Blurring: convolving image with low pass filter removing noise and edges (little). **Average** of all pixel replace the central. **Gaussian** specify width height with positive odd. standard deviation of x and y directions. **Median** highly effective against salt and pepper which random alters in pixel values, bright (salt) dark (pepper), noise density. **Bilateral** will remove noise and keeping sharp edges by considering pixel of similar intensities to preserve variation.

FIGURE 11. BLURRING

Type of Noise

additive white gaussian Noise (AWGN):

$$y(noisy\ image)$$

$$= x(clean\ image) + n(normal\ distribution)$$

mean of 0 variance of sigma,

Speckle: coherence principle, ultrasound, radar, granular pattern,

Poisson: digital image acquisition,

Real, paired noise clean or noise-noise image

4.2.4. Morphological Operations

for operations on image shapes, with help of original and structuring element we can decide the nature of operation to eliminate the unnecessary information

Erosion: by eroding away the white foreground boundary, decreasing the thickness.

$$\boldsymbol{A \ominus B = \{z | (B)_z \subseteq A\}}$$

Dilation: by increasing the white foreground to bring back broken parts and unshrink.

$$\boldsymbol{A \oplus B = \{z | (B)_z \cap A \neq \emptyset}$$

Opening will remove noise.

$$\boldsymbol{A \circ B = (A \ominus B) \oplus B}$$

Closing will fill the small holes inside the foreground.

$$\boldsymbol{A \circ B = (A \oplus B) \ominus B}$$

Morphological Gradient is the differencing image of erosion and dilation. **Top Hat** difference of input and opening. **Black Hat** difference of closing and input

High Pass/Gradient Filter

Sobel and Scharr: joint gaussian smoothing, resistant to noise, specify vertical (y) or horizontal (x) direction and better locate results

Laplacian: second order sum of derivative both axes for rapid intensity changes, isolate noise sensitive

4.3. Feature Extraction and Representation

4.3.1. Contours an:d Shape Analysis

Contours:

set of curve line joining all the points along the boundary of an image corresponding to extremities that has the same intensity. handy in shape analysis, finding the size of object, object detection. also approximated into a polygon

Contour Approximation:

Douglas-Peucker Algorithm to approximate the shape, maximize accuracy

Convexity Hull (bulged in empty space between fingers), checking the curve of defects, furthest point from convex (bulged out) point (flat finger tips). angle between the finger is found using cosine rule

$$c = \sqrt{a^2 + b^2 - 2ab\cos\gamma}$$

$$\gamma = \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$$

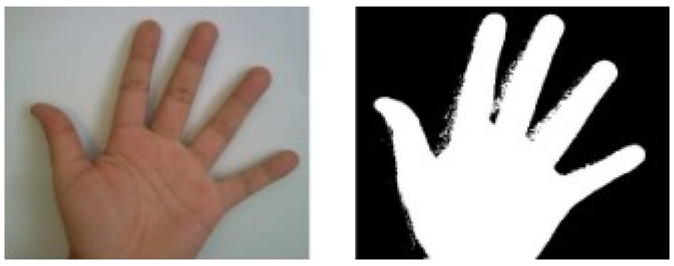


FIGURE 12. CONTOUR, CONVEXITY

Bounding Rectangle Block: track area of interest and are arrange activities to object layout on screen interface

Straight: rotation is not considered, not minimized. **Rotated**: minimum area structure for rectangle. **Minimum**: minimum enclosing circle area from circumcircle. **Fitting**: rotated rectangle in elliptical shape.

Properties

$$Aspect = \frac{Width}{Heigth}$$

$$Extent = \frac{Object\ Area}{Bounding\ Area}$$

$$Solidity = \frac{Contour\ Area}{Convex\ Hall\ Area}$$

$$Equivalent\ Diameter = \sqrt[4]{\frac{4 \times Contour\ Area}{\pi}}$$

4.3.2. Recognition System

Landmark Model

by utilizing a AI/ML pipeline that operates coordinates on cropped image region and high-fidelity 21 3D key points localization or hand knuckle landmarks are defined to linked them to intended functionalities labeling algorithms are applied to mark the region in the hand without the need of augmentations, and is solely focused on accurately predicting the coordinates. with many partially visible hands and self-occlusions over backgrounds

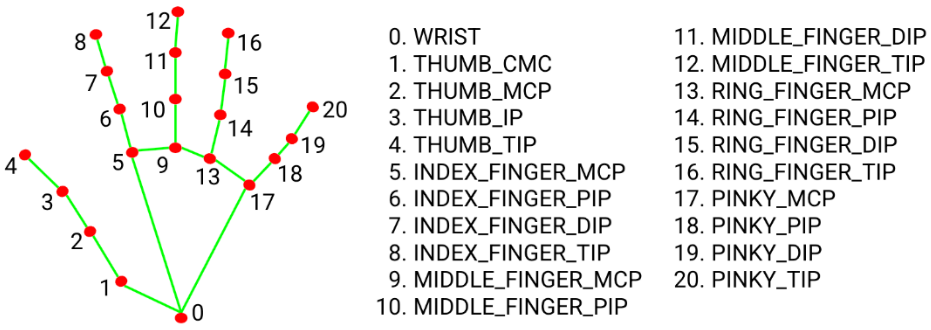


FIGURE 13. LANDMARKS

Functions Definition

when 'static image' is false it works on live video stream, 'max num hands' of 2 are detected by default then localized, with 'model complexity' of [0,1] and 'min detection confidence' [0.0 – 1.0] latency is key while processing video frames. only when we are to process static unrelated batches detection can run true on input images.

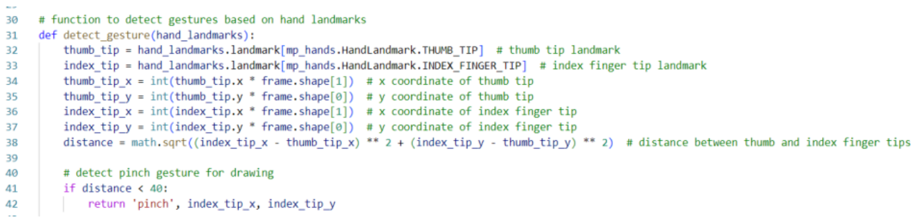
Hand Pose Estimation and Map Gestures

when an object is detected on a spatial configuration, key points are detected of human. they system is to understand their specific positions and interpret with response. hands module from media-pipe solution is initialized to drawing landmarks on hand, ‘Volume Adjustments Control’ distance between tip of thumb and index is calculated calling a math using a hypotenuse function.

$$p\ (x, y)\ = \sqrt{(x_2 - x_1)^2 + (y_2 - y_2)^2}$$

‘Pinch gesture’ when distance between tip of thumb and index finger is less than threshold allowing us to draw on the canvas

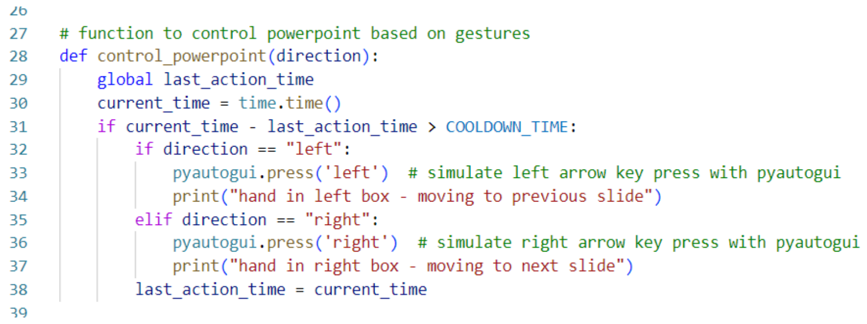
‘Two Finger Gesture’ when all fingers are folded in, and the index and thumb is spread out. it recognizes to clear the canvas as an eraser.



```
# function to detect gestures based on hand landmarks
def detect_gesture(hand_landmarks):
    thumb_tip = hand_landmarks.landmark[mp_hands.HandLandmark.THUMB_TIP]  # thumb tip landmark
    index_tip = hand_landmarks.landmark[mp_hands.HandLandmark.INDEX_FINGER_TIP]  # index finger tip landmark
    thumb_tip_x = int(thumb_tip.x * frame.shape[1])  # x coordinate of thumb tip
    thumb_tip_y = int(thumb_tip.y * frame.shape[0])  # y coordinate of thumb tip
    index_tip_x = int(index_tip.x * frame.shape[1])  # x coordinate of index finger tip
    index_tip_y = int(index_tip.y * frame.shape[0])  # y coordinate of index finger tip
    distance = math.sqrt((index_tip_x - thumb_tip_x) ** 2 + (index_tip_y - thumb_tip_y) ** 2)  # distance between thumb and index finger tips

    # detect pinch gesture for drawing
    if distance < 40:
        return 'pinch', index_tip_x, index_tip_y
```

FIGURE 14. GESTURES

‘Detection of Hand in Specific Area’ as the screen is divided into ‘Left and Right’ boxes is made as a left and right trigger to change into next or previous presentation slides.



```
# function to control powerpoint based on gestures
def control_powerpoint(direction):
    global last_action_time
    current_time = time.time()
    if current_time - last_action_time > COOLDOWN_TIME:
        if direction == "left":
            pyautogui.press('left')  # simulate left arrow key press with pyautogui
            print("hand in left box - moving to previous slide")
        elif direction == "right":
            pyautogui.press('right')  # simulate right arrow key press with pyautogui
            print("hand in right box - moving to next slide")
        last_action_time = current_time
```

FIGURE 15. MAPPING

4.3.3. Descriptors

Oriented Fast Rotated Brief:

is a feature descriptor for object recognition, picture registration, and image stitching, it combines the advantage of FAST detector and BRIEF descriptor. detection of pixel intensities with corners and edges while being resistant to image orientation. with it being efficient for real-time computer vision task

Histogram of Orientated Gradients

to get overall intensity distribution of [uint8 or float32] image (in BGR, greyscale plot) which contains contrasts, brightness. computation of gradient direction, pixel magnitude and concatenated to create feature vector

BINS are the hist-size sub-part of histogram that is split in group of 16 out of 256 values (small cells) **DIMS** is the number of collected data parameter. **RANGE** is all the intensity values. stretching to the ends for better distribution of intensities. the **Cumulative Distribution Function.**

Scale Invariant Feature Transform: mainly in image stitching, 3D constructions. it has 4 main steps detections (identification of interest points), localizations (refinement of positions), orientation (invariance in rotational), generation (distinct feature on local gradients)

Local Binary Patterns: mainly for texture classification, here each neighbor is compared with binary pattern code of the central pixel. then converted to decimal. and simply effective on monotonic greyscale.

Principal Component Analysis:

is a statistical strategy to reduce the dimensionality of data while also retaining its significant characters and preserve important features by computing covariance matrix with (eigen) vector and values to represent in lower dimensional representation and preserve significate patterns.

4.4. Segmentation and Subtraction

canny edge based: developed by John F Canny, 1986 as an algorithm to detect wide range of edges, due to its robustness it is still widely used. here it consists of many stages like gaussian smoothing (reducing noise and unnecessary details), gradient computation (both horizontal and vertical gradient change and direction), non-maximum suppression (preserving edge pixel perpendicular to local maxima with highest gradient magnitude), hysteresis thresholding (linking strong adjacent edges to form a continuous contours)

color-based: distinctive color properties are divided meaningfully which are typically RGB (red, green, blue), HSV (hue, saturation, value). because of their ability to separate color channels, intensity and isolate pixels, uniformity. these similarity measurements are achieved with account of human perception to differentiate colors.

skin tone: by isolation go human skin color with typical representation like YCbCr (Luma, Chrome Blue, Chrome Red), LAB (L*, a*, b*) which hold an advantage to separate tones. in various situation, lighting conditions, and shades within ethnicities, population. these variability across is also reproposed with robust Gaussian Mixture Models and SVM.

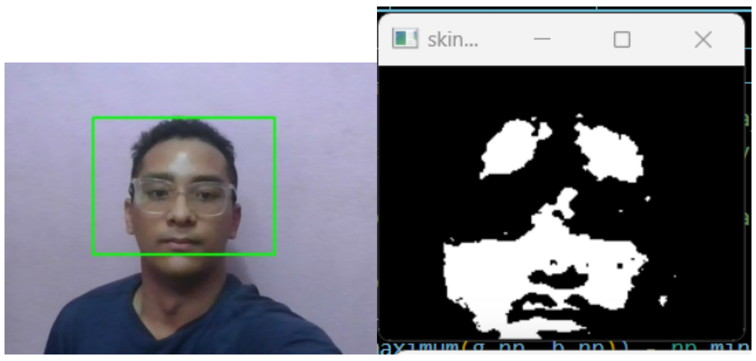


FIGURE 16. SKIN DETECTION

region based: by coherently grouping similar neighbor pixels (unlike abrupt edge), meaningful homogeneity division of intensity, texture, special proximity,

Background Subtraction:

if the camera is static, and the object is moving in the scene, the binary pixel image makes it easier to generate a foreground mask. here there is a background model (initialization and update) being used to perform subtraction.

4.5. Classifications

4.5.1. Intricate Signal Processing

aid in noise reduction, amplification of relevant signals, extracting features that maybe discriminative in classification task. wavelet analysis

4.5.2. Dynamic Time Wraping

best matching sequence based on best path with least overhead. measuring similarity in sequence of varying time or speed. useful in speed variation time series. feature vector sequence is compared and extracted.

4.5.3. Novel Recogniation

identifying instances or pattern that are subtle and significantly different from the known-trained data like rare-event anomalies, like how outlier deviate from normal pattern. useful in security surveillance, autonomous driving, quality checking and medical images.

4.5.4. Bayesian Appraoch

Naïve Bayes and Bayesian Network are probabilistic model to classify points, reasoning and decision making.

4.5.5. K Nearnest Neighbours

when determining similarity based on where majority classes, feature vectors, pixel descriptors is nearest in the space. in instance and being non parametric to new data points.

4.5.6. Support Vector Methods

a supervised learning used to find optimal hyperplane separating data to classes with largest margin meaning confidently distinguish between multiple objects in the visual input.

4.5.7. Maximally Stable Extremal Region

is useful when the image is stable even across variating scale of intensity. the bright and dark regions are detected relative the surrounding.

4.6. Network Model

4.6.1. Convolutional Neural Network

learn automatically about hierarchical representation of pixel values, they are able to retain crucial information even when the spatial dimension are reduced during pooling. it is evolving with multiple layers and excel higher when deeply evolves.

4.6.1. Recurrent Neural Network

capable of evaluating both discrete sequence and continuous data in temporal relations, over time the information persist. in hybrid transformer-based step by step hidden dependencies.

4.6.2. Generative Adverssarial Network

they are able to generate new sample dataset (generator) similar to training and evaluate (discriminator). they play game that tries to fool each other. improving each other's ability.

4.6.1. (Region) CNN

by adding region of interest in images that is likely to contain object. and they extract features and improving on both boundary box and classification tasks.

4.6.2. Long Short Term Memory

feedback connection than traditional feed forward, synaptic fluctuation strength, change in weight and bias of connection as it is allowed to discard portion od past information. as it can hold informative memory for long time.

4.6.3. Deep Neural Network

multi layers of neurons, the input data is passed to next layers to transform and compute. it has non-linear activation functions like ReLU, Sigmod to capture complex patterns

4.6.4. Artificial Neural Network

utilization of backpropagation to adjust weight and error, weighted sum applies on activation, raw data like pixel values are composed in the neurons.

4.6.5. Hidden Markov Method

search for hidden sequence of states within apparent sequence in order to decipher messages. the conditions are not readily observable. there is likelihood of observation to move stages

4.6.6. Diffusion Probabilistic Model

denoising diffusion probabilistic models (DDPMs)

a forward Markovian diffusion process

$$q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I\right))$$

$$where\ \alpha_t = 1 - \beta_t\ and\ \bar{\alpha}_t = \prod_{i=1}\alpha_i$$

inverse diffusion process recovers the original image from Gaussian noise (noise reduction)

forward process is small, the inverse process can also be modeled as a Markov chain process

4.6.7. Multi-Layer Perception

a fully connected network, layers are connected to each neuron previous and next, predefined categories are assigned to the partitioned images based on pattern and features.

4.7. Post Production

Performance Optimization: making sure that for top real-time speed, the redundancy is minimum with proper structuring

Functionality Enhancements: adding more controls, more interactive elements.

Reliability: adding more preprocessing, segmentation, feature extractions, thresholding

Stability: testing across different environments. conditions, machines to identify bugs and performance impacts

Documentation: making a comprehensive report that include all the necessary instructions and references

4.8. Implementaion Challenges

illumination and background variation

variation of lighting conditions and background has significant effects on the performance and algorithm isolating any hands or object without such will result in poor output

the accuracy would be low when applying preprocessing techniques with physical object in background interfering

palm articulation occlusion

the occlusion and orientation of hand, fingers, palm, knuckles would cause weird obstruction in recognition

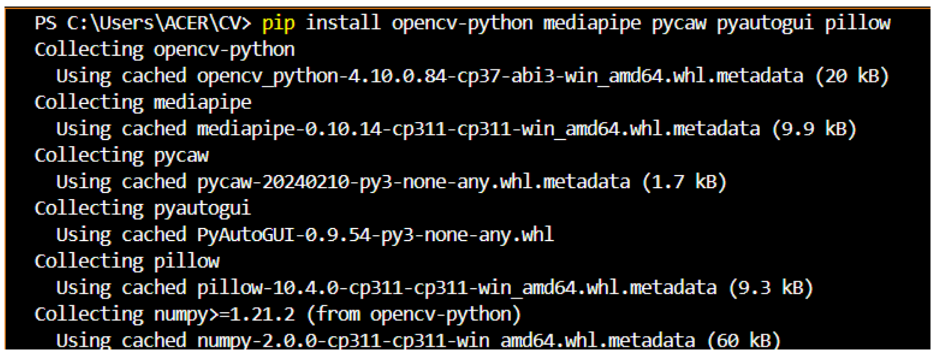
when landmarks over lap it can mis trigger gesture functions

limited vocabulary

by not recognizing hand gestures and variation in movements will lead to inaccurate interactions, having limited set of distinct gestures without repeating

camera position

not having a dedicated webcam and using integrated cam can have negative impact on visibility and clarity, it can even distort compromising the accuracy,



```
PS C:\Users\ACER\CV> pip install opencv-python mediapipe pycaw pyautogui pillow
Collecting opencv-python
  Using cached opencv_python-4.10.0.84-cp37-abi3-win_amd64.whl.metadata (20 kB)
Collecting mediapipe
  Using cached mediapipe-0.10.14-cp311-cp311-win_amd64.whl.metadata (9.9 kB)
Collecting pycaw
  Using cached pycaw-20240210-py3-none-any.whl.metadata (1.7 kB)
Collecting pyautogui
  Using cached PyAutoGUI-0.9.54-py3-none-any.whl
Collecting pillow
  Using cached pillow-10.4.0-cp311-cp311-win_amd64.whl.metadata (9.3 kB)
Collecting numpy>=1.21.2 (from opencv-python)
  Using cached numpy-2.0.0-cp311-cp311-win_amd64.whl.metadata (60 kB)
```

FIGURE 17. INSTALLATIONS

5. Result Theory

5.1. Library Used

OpenCV: is a comprehensive software library for machine learning and computer vision used in fields like robotics, it can process images in real time, and due to it being open source it has extensive functionalities like edge detection, morphological operations, detection and recognition (surf, orb, harr cascade), segmentation (isolation of region of interest, watershed, grab cut),

Media Pipe: developed by Google for its simple, efficient building solutions for perceptual pipelines, multimodal. it is popular for being cross-platform. and providing highly fast and accurate tracking solutions, such detection and mesh are optimized for performance, identification of landmarks, frame location,

NumPy: library for numerical computing in python, it can handle large multidimensional arrays and matrices, for representation of image in array of pixel (color space like RGB and greyscales). we can resize, normalize, crop, apply filters (Gaussian blur) and geometric transformations (translation, rotate, scale) with its methods,

Math: with this we can use various mathematical operations, like transformation around the point, calculate the Euclidean distance with ‘sqrt’ and ‘hypot’, calculate angles for orientation. normalization with scaling, binning pixel value,

PyAutoGUI: is a cross-platform library that will support mouse (clicking) and keyboard(typing) to manipulate windows, it is important if we want to interact with the interface (screen capture, windows manipulation) and automate if necessary

Pillow/PIL: Python Imaging Library is powerful for image processing, with support for many formats which is fundamental to any pipeline that reads image and process output, it can manipulate flipping operations, debugging, color manipulation and conversion (brightness),

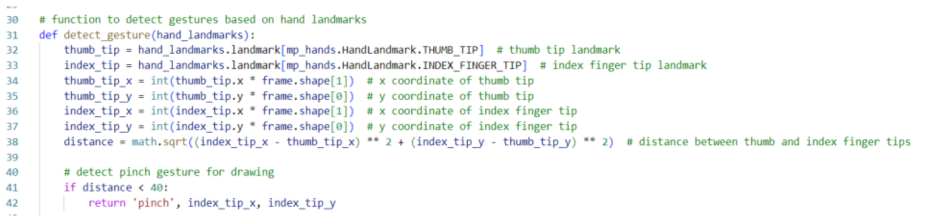
PyCaw: a python core audio window to have a convenient interface to interact and control with core audio API that can control volume levels

Scikit-learn: is a powerful library for machine learning in python, it is very efficient used for feature extractions (edges, texture) with HOG, SIFT, detect object and classification (support vector machines), regression, clustering, evaluation (cross-validation, accuracy, confusion)

Matplotlib: being able to visualize data, analyze result, and performance of model, as a flexible static, interactive, versatile framework to raw image, histogram of pixel value distribution, comparison of images side by side and overlaying with points, lines to highlight an area

5.2. Algorithms

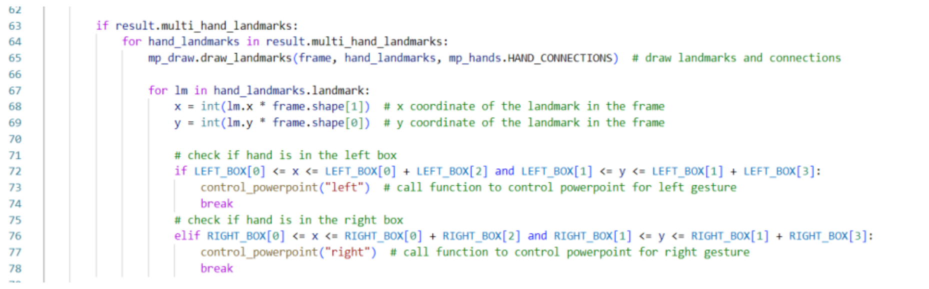
Media Pipe Hand Tracking: developed by google that gives high-fidelity using computer vision to track 21 key landmarks on each hand from an input video feed. by the use of lightweight neural network for the pipeline it identifies region and detects the likeliness of existence of a palm and various points of the fingers are added. it can accurately track the movements even in fast motions. this is effective for designing real-time applications like gesture control, augmented and virtual reality.



```python
# function to detect gestures based on hand landmarks
def detect_gesture(hand_landmarks):
    thumb_tip = hand_landmarks.landmark[mp_hands.HandLandmark.THUMB_TIP]  # thumb tip landmark
    index_tip = hand_landmarks.landmark[mp_hands.HandLandmark.INDEX_FINGER_TIP]  # index finger tip landmark
    thumb_tip_x = int(thumb_tip.x * frame.shape[1])  # x coordinate of thumb tip
    thumb_tip_y = int(thumb_tip.y * frame.shape[0])  # y coordinate of thumb tip
    index_tip_x = int(index_tip.x * frame.shape[1])  # x coordinate of index finger tip
    index_tip_y = int(index_tip.y * frame.shape[0])  # y coordinate of index finger tip
    distance = math.sqrt((index_tip_x - thumb_tip_x) ** 2 + (index_tip_y - thumb_tip_y) ** 2)  # distance between thumb and index finger tips

    # detect pinch gesture for drawing
    if distance < 40:
        return 'pinch', index_tip_x, index_tip_y
```

FIGURE 18. LANDMARK DETECTIONS

Gesture Recognition: to interpret gestures shown by human using models and algorithm for computer to understand and make it possible to interact with devices in natural and intuitive ways. here the visual data is captured with cameras, such captured video streams are analyzed and prepared with noise reduction, subtraction of background. key features and patterns are then identified such as points, shapes, etc. use of ML models like Hidden Markov with Convolutional and Recurrent Neural Networks are able to recognize the specific pose or movements over the air.

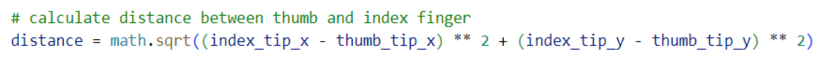


```python
        if result.multi_hand_landmarks:
            for hand_landmarks in result.multi_hand_landmarks:
                mp_draw.draw_landmarks(frame, hand_landmarks, mp_hands.HAND_CONNECTIONS)  # draw landmarks and connections

                for lm in hand_landmarks.landmark:
                    x = int(lm.x * frame.shape[1])  # x coordinate of the landmark in the frame
                    y = int(lm.y * frame.shape[0])  # y coordinate of the landmark in the frame

                    # check if hand is in the left box
                    if LEFT_BOX[0] <= x <= LEFT_BOX[0] + LEFT_BOX[2] and LEFT_BOX[1] <= y <= LEFT_BOX[1] + LEFT_BOX[3]:
                        control_powerpoint("left")  # call function to control powerpoint for left gesture
                        break
                    # check if hand is in the right box
                    elif RIGHT_BOX[0] <= x <= RIGHT_BOX[0] + RIGHT_BOX[2] and RIGHT_BOX[1] <= y <= RIGHT_BOX[1] + RIGHT_BOX[3]:
                        control_powerpoint("right")  # call function to control powerpoint for right gesture
                        break
```

FIGURE 19. RECOGNITIONS FOR ART

Euclidean Distance Calculation: to measure the distance between two points in a straight line for feature matching and descriptors tasks, detections and comparisons.

$$P(x,y) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



```python
# calculate distance between thumb and index finger
distance = math.sqrt((index_tip_x - thumb_tip_x) ** 2 + (index_tip_y - thumb_tip_y) ** 2)
```

Linear Mapping: transformation function between vector spaces with addition and scalar multiplication, mostly used to geometric transformations, extraction (linear discriminant), mapping filtering to enhance specific edges and textures,

$$f(u+v) = f(u) + f(v)$$

6. Proposed Solution

by capturing real-time video feed, the process design is to describes the multi-layer architecture of the computer vision system that can use multiple algorithms and applicational uses

6.1. Diagram

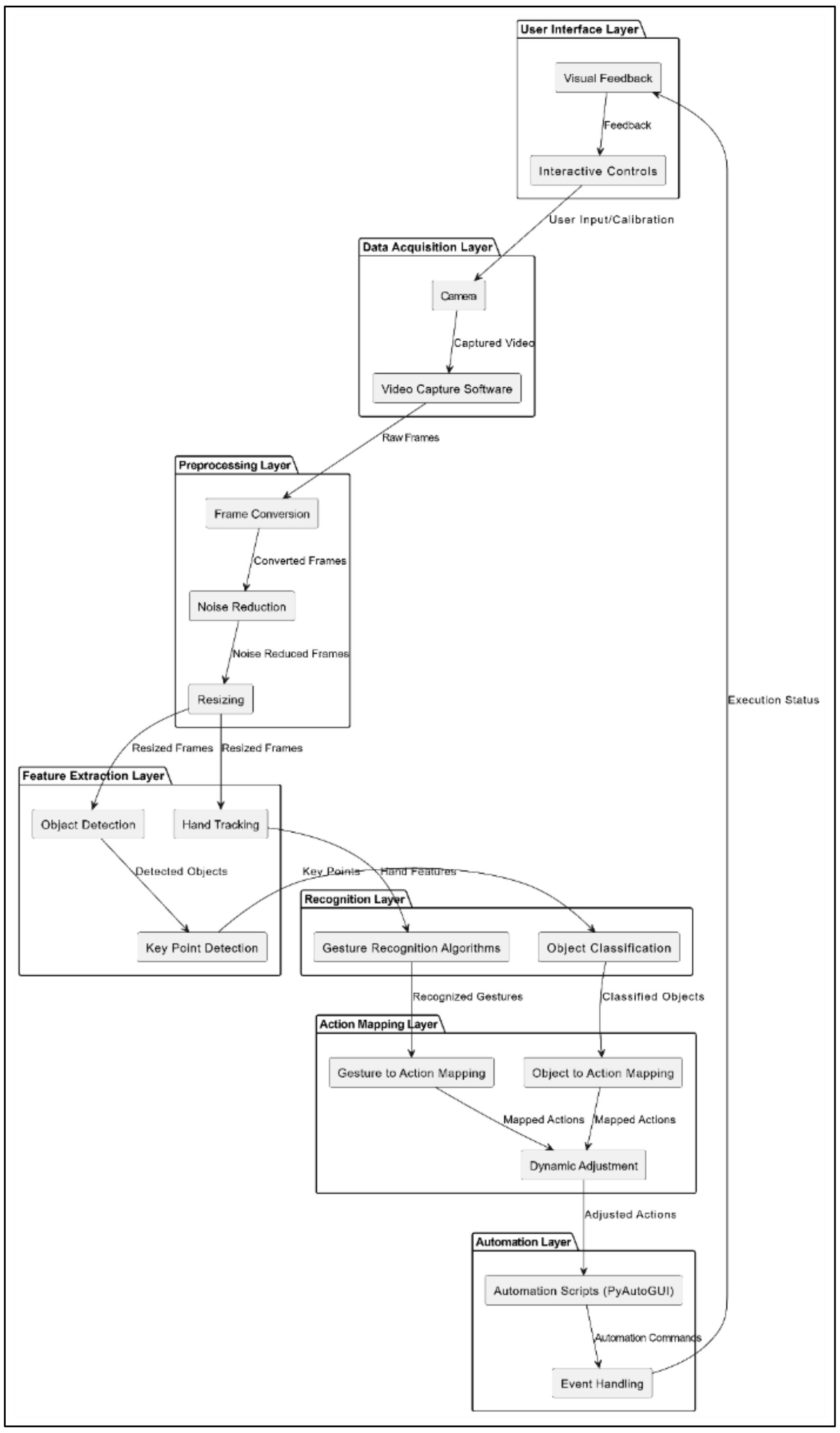


FIGURE 20. SYSTEM ARCHIETECTURE FOR COMPUTER VISION

6.2. Project Tools

setting up the development environment and installing the required libraries.

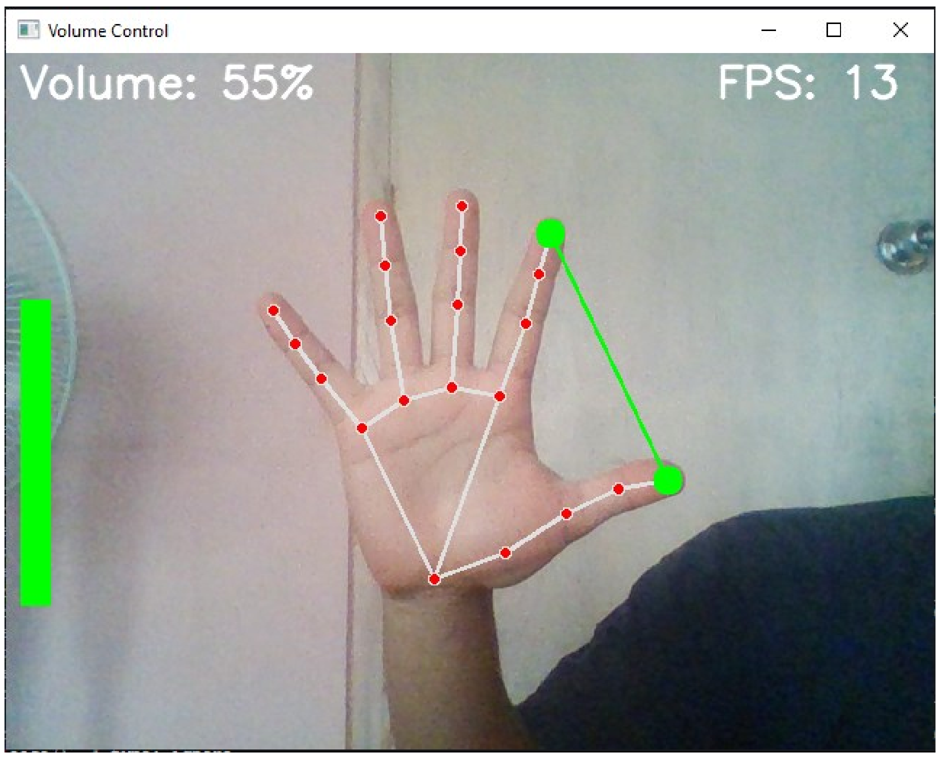


FIGURE 21. VOLUME CONTROL

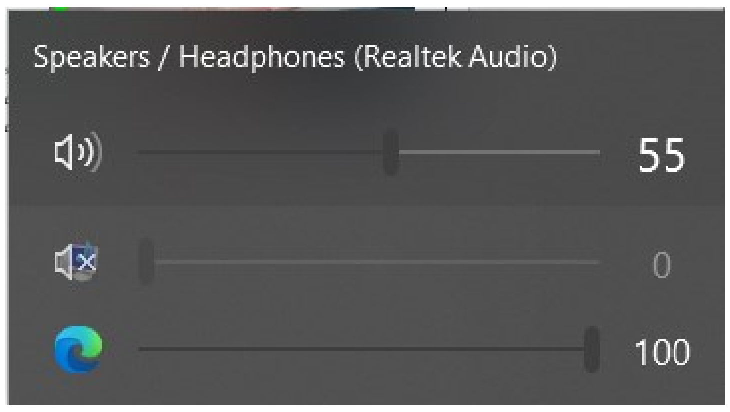


FIGURE 22. EFFECT OF VOLUME

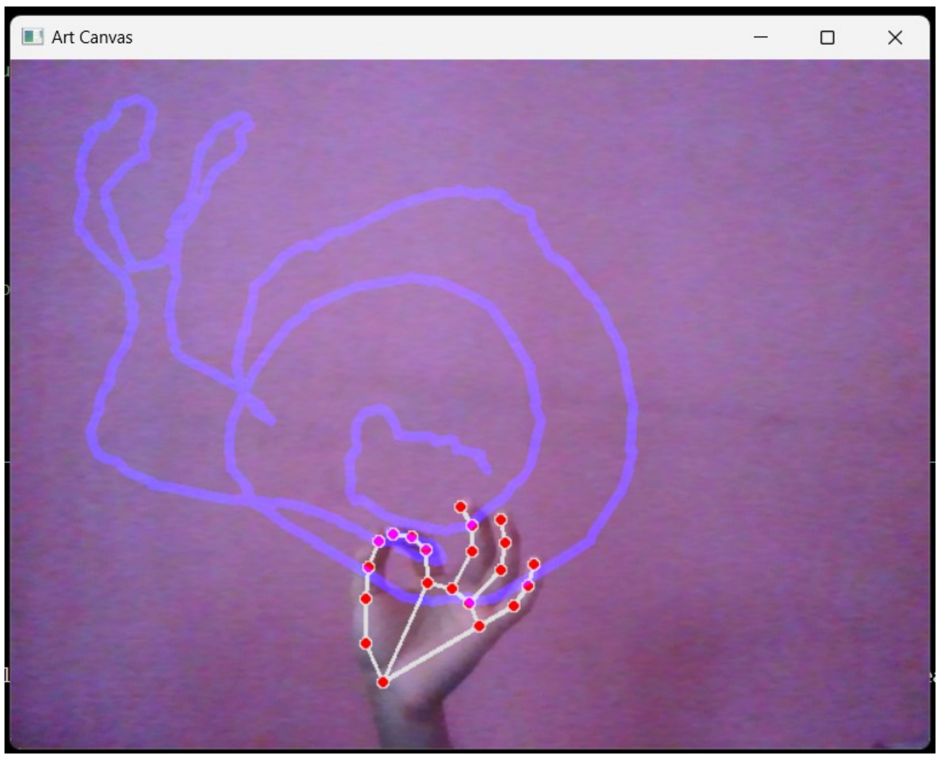


FIGURE 23. ART CANVAS

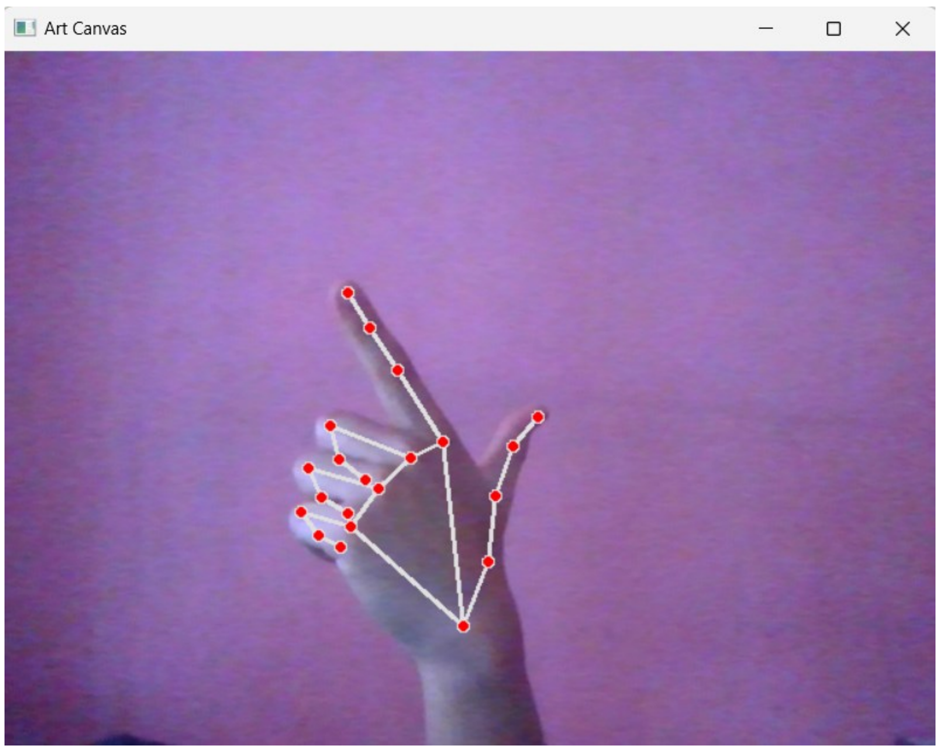


FIGURE 24. ERASE ART

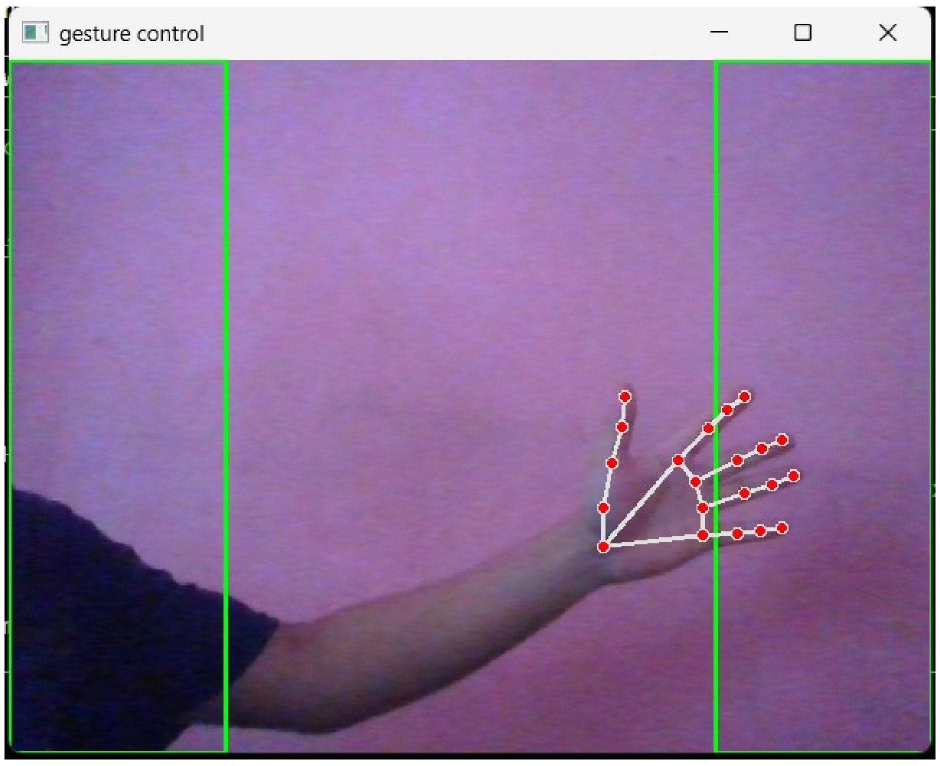
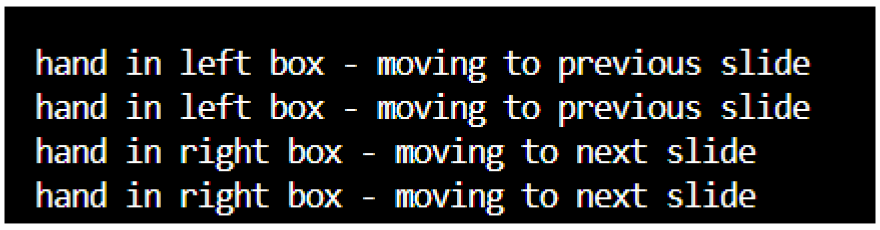


FIGURE 25. PRESENTATION INTERFACE



```
hand in left box - moving to previous slide
hand in left box - moving to previous slide
hand in right box - moving to next slide
hand in right box - moving to next slide
```

FIGURE 26. TERMINAL OUTPUT FOR PPTX

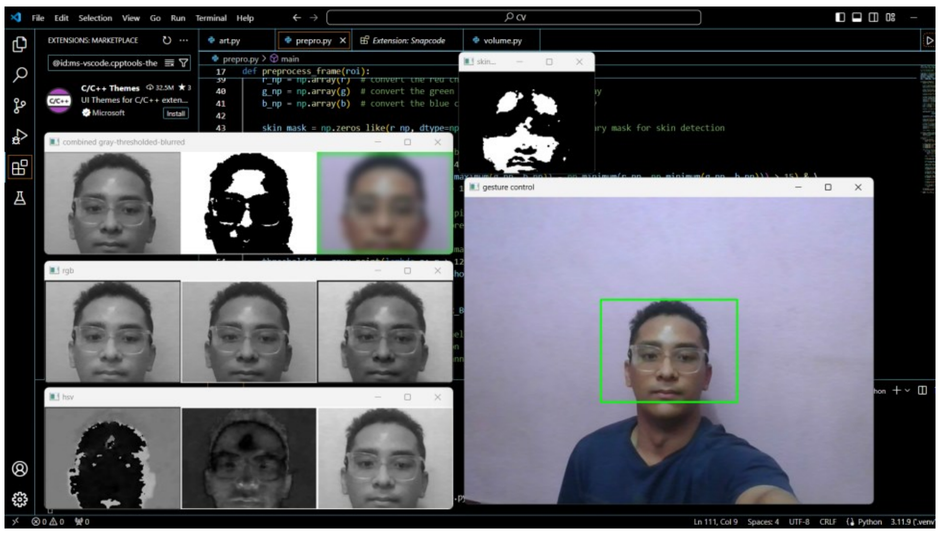


FIGURE 27. DIFFERENT IMAGE PROCESSING

7. Discussion

7.1. Justification

this approach is chosen as it can be scaled up and adapted to recognize a lot more than just few gestures, in reality it can make versatile application that will bridge gap between the deaf community and other

deep learning can adapt to multiple variation and environment, which is suitable for task dynamic nature. further more it can demonstrate to be more efficient and accurate

7.2. Implications and Emprovements

this module is highly accurate so it can be potentially effective in real world application. some gestures were misinterpreted due to occlusion and lighting conditions didn’t help in registering right gesture outputs. so increasing the robustness to work on difficult lighting would be key.

7.3. Comparision

We can see improvement in the later iterations of the systems. Compared with those existing bench marks it was able to match and sometimes be better than shown.

Although responsive, the system managed to produce results. The interaction and accessibility also worked well in providing defined ranges of functionality in real time. In reflecting on the developmental process, one can acknowledge the strengths and limitations noted by the area of future improvements expanding and refining the methods. Much more detailed assessment of dexterity is required. It improves the outcome's effectiveness, usability, and performance, and works out directions on research and its impact on users. acknowledge the limitation as it requires extensive computational resources and needs generalization of different data sets, hence would want to try making a lightweight model. Recognizing stylized and cursive is suggested as future exploration.

8. Conclusion

without the need of physical contact. the system helps users to have a more intuitive hands-free controller function. these are very natural and practical in current world where technology is made with inclusivity in mind. such open broader audience that are not cared for. The problems are getting more complex with its solutions needing effective machine learning techniques leveraging the strengths, providing robustness and accuracy for problem at hand.

more people seem to work out their old devices and lack the newer technologies, although they are more comprehensive. the developer feel that language support exceeds the leveraging hard and advanced technology to help those with limited access to healthcare. on empirical evidence. the pipelines work with wide application. multiple -modalities is also in consideration. hybrid model and combination of image processing and machine learning will improve the accuracy and address the limitations.

8.1. Future Works/Scopes/Development

the successful implementation and development of real-time hand gesture recognition system by combining computer vision technologies. as the system effectively detects the hand gesture and adjust the volume, and multimedia application in high accuracy. is intuitive and accessibility reiterates the huge potential in various settings. the system's ability to be as accurate as traditional methods and more valuable in rehabilitative settings. performance is effectiveness and enhancement with approach of advanced machine learning techniques models. not only computer vision, natural language process is also suitable in facilitating the individuals with some utility

9. Appendixes

9.1. References

Akbari Sekehravani, Ehsan & Babulak, Eduard & Masoodi, Mehdi. (2020). Implementing canny edge detection algorithm for noisy image. Bulletin of Electrical Engineering and Informatics. 9. 1404-1410. 10.11591/eei.v9i4.1837.

Anjaneya, L & U., Dr. (2023). MACHINE LEARNING APPROACH TO THE CLASSIFICATION AND IDENTIFICATION OF HAND GESTURE RECOGNITION USING PYTHON. International Journal of Recent Scientific Research. 14. 4372-4377. 10.24327/ijrsr.20231411.0821.

Biswas, Soumya. (2023). Hand Gesture Recognition Model for Task Execution. 10.13140/RG.2.2.29877.86243.

Bouraya, Sara & Belangour, Abdessamad. (2024). Dissecting of the two-stages object detection models architecture and performance. Bulletin of Electrical Engineering and Informatics. 13. 1694-1706. 10.11591/eei.v13i3.6424.

Camgoz, Necati & Koller, Oscar & Hadfield, Simon & Bowden, Richard. (2020). Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation. 10.1109/CVPR42600.2020.01004.

Hussain, Noor & Abdul Kareem, Emad. (2023). Detecting Hand Gestures Using Machine Learning Techniques. Ingénierie des systèmes d information. 27. 10.18280/isi.270612.

Jadhav, Utkarsh & Mishra, Rishabh & Mishra, Shivam. (2024). Text Detection in images & video processing.

Loresco, Pocholo & Bandala, Argel. (2018). Human Gesture Recognition Using Computer Vision for Robot Navigation.

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, D., & Gebru, T. (2019). *Model Cards for Model Reporting*. https://arxiv.org/pdf/1810.03993

Sarma, Debajit & Bhuyan, M.. (2021). Methods, Databases and Recent Advancement of Vision-Based Hand Gesture Recognition for HCI Systems: A Review. SN Computer Science. 2. 10.1007/s42979-021-00827-x.

Wu, J., Xie, R., Wu, H., & Yuan, G. (2024). Improving the generalization of image denoising via structure-preserved MLP-based denoiser and generative diffusion prior. *IET Image Processing*. https://doi.org/10.1049/ipr2.13122

Zheng Yu, Tan & Basah, Shafriza & Yazid, Haniza & Safar, M. Juhairi Aziz. (2021). Performance analysis of Otsu thresholding for sign language segmentation. Multimedia Tools and Applications. 80. 1-22. 10.1007/s11042-021-10688-4.