Gesture-Based Control System for Presentation Slide Navigation

Enhancing Presentation Interaction through Natural Hand Gestures

Tirth Patel
Department of Electronics and communication
Nirma university
Ahmedabad,Gujarat
21bec094@nirmauni.ac.in

Dhairya Senghani
Department of Electronics and communication
Nirma university
Ahmedabad,Gujarat
21bec111@nirmauni.ac.in

***Abstract* — Our ML - powered system enables seamless control of PowerPoint presentations through hand gestures, enhancing user engagement and accessibility. The system uses computer vision techniques to detect gestures and trigger slide transitions, offering a novel and intuitive way to navigate presentations seamlessly.**

Keywords — ML, hand gesture recognition, PowerPoint control, computer vision, win32.com, cv zone, OS

I. Introduction

In recent years, technological advancements have revolutionized the way presentations are delivered, with traditional slide decks being replaced by interactive and immersive experiences. One such innovation is the use of hand gestures to control PowerPoint presentations, offering presenters a seamless and intuitive way to navigate slides without the need for conventional input devices.

The research paper delves into the novel concept of "PPT control using hand gestures" within the realm of presentation technology. By leveraging Python programming, specifically libraries like OpenCV, cvzone, and win32com, this paper investigates the development and implementation of a hand gesture recognition system for controlling presentations. Through computer vision and machine learning techniques, presenters can navigate slides seamlessly with intuitive hand movements, offering a dynamic and engaging presentation experience. The paper explores the design process, integration challenges, and potential applications of this technology across various domains, including education and business presentations. By empowering presenters to interact with their slides effortlessly, hand gesture recognition systems redefine traditional presentation methods, enhancing accessibility and audience engagement. This paper aims to shed light on the feasibility, effectiveness, and future prospects of incorporating hand gestures into presentation software, paving the way for more interactive and captivating presentations in the digital age.

II. Literature Survey

A. *Salonee Powar, Shweta Kadam, Sonali Malage, Priyanka Shingane*

The literature review provides an in-depth exploration of hand gesture recognition within the context of Human-Computer Interaction (HCI), focusing on the technological advancements and methodologies employed in the field. It elucidates the evolution of gesture recognition techniques, from wearable devices such as rings and armbands to computer vision-based approaches utilizing cameras and machine learning algorithms. The review underscores the significance of OpenCV for image processing and hand detection, alongside libraries like PyAutogui for simulating mouse and keyboard actions. Furthermore, it discusses the utilization of machine learning algorithms, such as Support Vector Machines (SVM), for accurate gesture classification. The review also examines the integration of frameworks like MediaPipe for robust hand and finger tracking. Overall, it synthesizes existing research to lay the foundation for the proposed AI-driven hand gesture detection system, leveraging a combination of computer vision, machine learning, and sophisticated algorithms to enhance digital presentation control through intuitive gestures.*[1]*

Algorithm used	AI-based Hand Gesture Detection
Accuracy achieved	98.73 %

B. *Ahmed Kadem Hamed AlSaedi1 and Abbas H. Hassin AlAsadi*

The research paper titled "A New Hand Gestures Recognition System" by Ahmed Kadem Hamed AlSaedi and Abbas H. Hassin AlAsadi provides a comprehensive overview of gesture recognition systems, focusing on the development of a low-cost system for real-time hand gesture recognition. The paper discusses the historical background of gesture communication and the significance of gestures in human interaction. It examines various approaches to hand gesture recognition, including vision-based methods and data glove approaches, highlighting their advantages and challenges. The proposed system, implemented using Python and OpenCV libraries, is detailed in five steps: image acquisition, pre-processing, hand region segmentation, feature extraction, and gesture recognition. The paper presents experimental results demonstrating the system's high recognition rate and flexibility in different conditions. Additionally, it discusses future enhancements, such as incorporating both hands for gesture recognition, to improve system accuracy and usability. Overall, this research paper contributes valuable insights into the field of gesture recognition and provides a framework for developing efficient and cost-effective systems for human-computer interaction.[2]

Algorithm used	Image Acquisition, Pre-Processing, Hand Region Segmentation, Contour Extraction, Features Extraction and Recognition
Accuracy achieved	96.60 %

C. *Ishika Dhall, Shubham Vashisth and Garima Aggarwal*

The literature review incorporates a comprehensive examination of recent advancements in automated hand gesture recognition using deep convolutional neural networks (CNNs). The study highlights the transformative impact of deep learning, particularly CNNs, in computer vision applications. By leveraging CNNs, which excel at extracting relevant features from input images without manual tuning, researchers aim to revolutionize human-computer interaction by enabling intuitive gesture-based communication. The review underscores the potential of hand gesture recognition to replace traditional input devices Moreover, the study explores methodologies, algorithms, and libraries such as OpenCV, Keras, and TensorFlow employed in developing robust hand gesture recognition systems. Through a synthesis of existing literature, the review sets the stage for the proposed application of deep CNNs in building a hand gesture recognition system, offering insights into its significance and future prospects.[3]

like touchscreens and keyboards, offering enhanced security and efficiency. It discusses the architecture and functionality of CNNs, emphasizing their ability to learn intricate patterns and features crucial for accurate gesture classification.

Algorithm used	Deep Convolutional Neural Network
Dataset used	Collected dataset
Accuracy achieved	99.13 %

D. *Nelson Sinaga, Baharuddin, Hesti Fibriasari, Bakti Dwi Waluyo, Joni Syafrin Rambey*

The literature review conducted for this research paper encompasses a comprehensive analysis of existing studies and methodologies related to hand gesture recognition systems. The proposed research builds upon previous works by incorporating an Artificial Intelligence-based approach to hand gesture detection, aiming to enhance interaction with digital presentations. The methodology involves preprocessing the input image, utilizing the Anaconda framework along with libraries such as OpenCV, PyAutoGui, MediaPipe, and others. The research leverages techniques like image segmentation, distance transform, and finger counting to detect and recognize hand gestures accurately.[4]

Algorithm used	Boundary Selection, Color Masking, Blurring, Contour Detection, Finger Identification
Accuracy achieved	97.56 %

III. METHODOLOGY

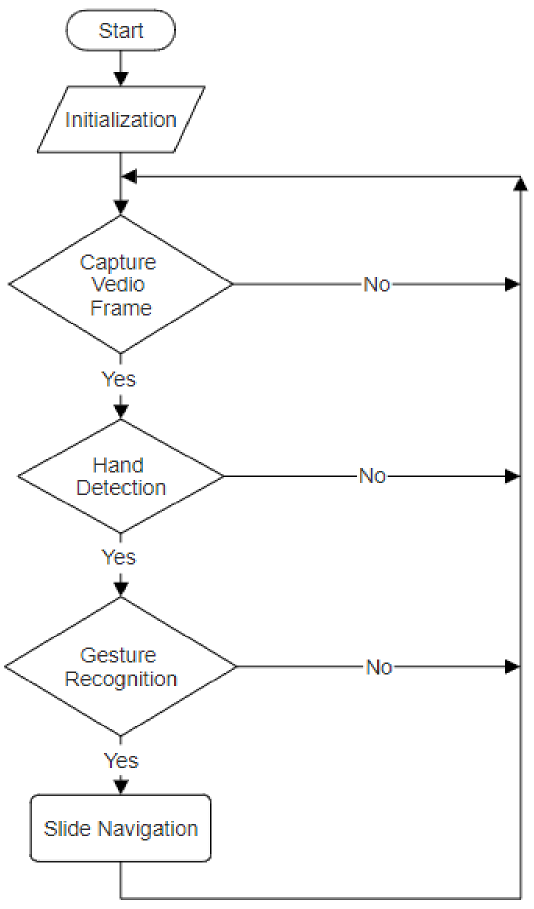


Figure 1 Flow of the System

A. *System Setup:*

Utilize webcam for gesture recognition and PowerPoint control via win32com.client.

B. *Hand Gesture Detection:*

Employ HandDetector module from cvzone for real-time hand gesture tracking, configured for single-hand detection.

C. *Presentation Control Logic:*

Define gestures (thumbs up, whole hand pointing up) for previous and next slide navigation based on hand center position relative to a gesture threshold.

D. *Interaction Flow:*

Continuously capture video frames, process hand detection and gesture recognition, and trigger PowerPoint actions accordingly.

E. *Implementation Details:*

Utilize OpenCV (cv2) for webcam access, cvzone for gesture detection, win32com.client for PowerPoint interaction, numpy for computations, and aspose.slides for PowerPoint functionalities.

F. User Feedback and Visual Annotations:

Display feedback messages ("Next" or "Previous") in console and support visual annotations on the video stream.

G. Iterative Development and Testing:

Iterate development cycles, testing different gestures, and refining recognition logic for accuracy and responsiveness. Continuous validation ensures reliability across various conditions.

IV. Libraries Implementation

A. win32com.client

The win32com.client library provides a powerful interface for interacting with Windows applications through COM (Component Object Model) interfaces, enabling programmatic control and automation of Windows-based software.

This library is specifically utilized in the project to programmatically control the PowerPoint application, allowing seamless integration of hand gesture-based control functionalities.

```
Application=
win32com.client.Dispatch("PowerPoint.Applica
tion")
```

B. cv2 (OpenCV):

OpenCV (Open-Source Computer Vision Library) is a widely used open-source computer vision and image processing library, offering a plethora of tools and algorithms for various vision-related tasks.

OpenCV is instrumental in capturing video frames from the webcam, processing images, and displaying visuals. In this project, it's crucial for real-time hand gesture detection and tracking

```
cap = cv2.VideoCapture(0)

success, img = cap.read()

cv2.imshow("Image", img)
```

C. cvzone.HandTrackingModule:

The cvzone.HandTrackingModule, part of the cvzone library, provides advanced hand tracking and gesture recognition functionalities, offering a streamlined approach for integrating hand gestures into applications.

This module provides advanced hand tracking and gesture recognition functionalities. In your code, the HandDetector class from cvzone.HandTrackingModule is used to detect and track hand movements for gesture-based control.

```
    From cvzone.HandTrackingModule import
HandDetector

    detectorHand = HandDetector(detectionCon=0.8,
maxHands=1)

    hands, img = detectorHand.findHands(img)
```

D. os:

The os module in Python provides a platform-independent interface for interacting with the operating system, facilitating file and directory operations, environment management, and system-related tasks.

In this project, the os module is employed to specify and access the file path for opening the PowerPoint presentation within the system environment.

```
Presentation =
Application.Presentations.Open("C:\\Users\\u
sername\\Desktop\\presentation.pptx")
```

E. numpy:

NumPy (Numerical Python) is a fundamental numerical computing library in Python, offering support for multidimensional arrays, mathematical functions, and linear algebra operations.

NumPy is utilized for array manipulations, mathematical computations, and data processing tasks, enhancing the efficiency of data handling in the project.

```
import numpy as np
imgList = []
```

F. aspose.slides:

Aspose.Slides is a comprehensive library for working with Microsoft PowerPoint presentations programmatically, providing extensive functionalities for creating, editing, and manipulating slides and slide elements.

In this project, the aspose.slides library is used for controlling slide transitions within the PowerPoint presentation environment, contributing to the core functionality of gesture-controlled navigation.

```
import aspose.slides as slides
Presentation.SlideShowSettings.Run()
```

V. Results

A. Photo 1: Left Hand Detection

The *Figure 2* serves as a reference for detecting the user's left hand. This reference image is crucial for the system to accurately identify and differentiate the left hand from the right hand. By analyzing key features and characteristics specific to the left hand in this photo, such as finger positions or palm orientation, the system establishes a baseline for left hand detection during gesture recognition tasks.

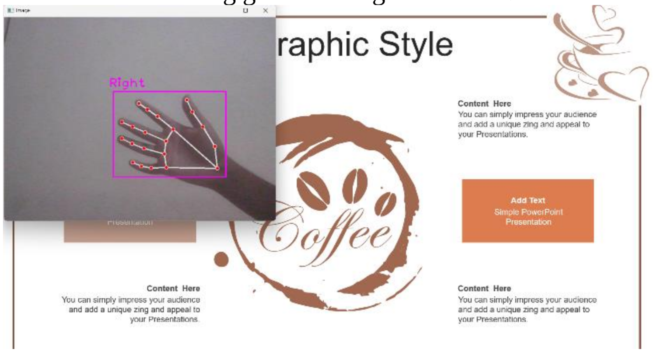


Image
Right
raphic Style
Content Here
You can simply impress your audience and add a unique zing and appeal to your Presentations.
Add Text
Simple PowerPoint Presentation
Coffee
Content Here
You can simply impress your audience and add a unique zing and appeal to your Presentations.
Content Here
You can simply impress your audience and add a unique zing and appeal to your Presentations.

Figure 2 Left Hand Detection

B. Photo 2: Right Hand Detection

Similarly, the *Figure 3* acts as a reference for detecting the user's right hand. This reference image provides essential information for the system to distinguish the right hand from the left-hand during hand gesture recognition. The photo's features, such as finger arrangement or hand shape, enable the system to identify and track the right hand accurately, facilitating precise gesture recognition and control of slide navigation based on right hand gestures.

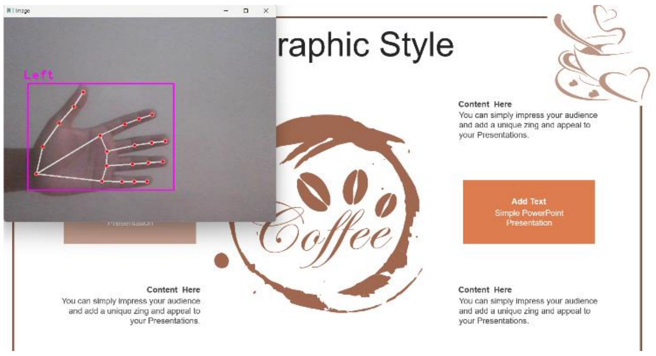

Left
raphic Style
Content Here
You can simply impress your audience and add a unique zing and appeal to your Presentations.
Add Text
Simple PowerPoint Presentation
Coffee
Content Here
You can simply impress your audience and add a unique zing and appeal to your Presentations.
Content Here
You can simply impress your audience and add a unique zing and appeal to your Presentations.

Figure 3 Right Hand Detection

C. Photo 3: Gesture Recognition

The *Figure 4* and *Figure 5* capture the specific hand gestures used for controlling slide navigation in the system. This photo includes examples of Gesture 1 (thumbs up) and Gesture 2 (whole hand pointing up), which correspond to navigating to the previous slide and next slide, respectively. By training the system to recognize these gestures based on visual cues captured in the photo, users can seamlessly interact with presentations using natural hand movements, enhancing user experience and engagement during presentations.

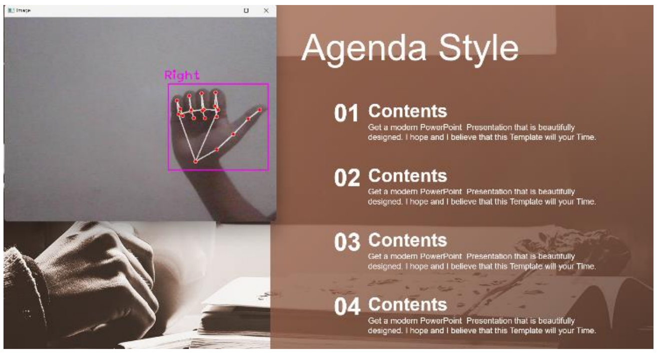

Right
Agenda Style
01 Contents
Get a modern PowerPoint Presentation that is beautifully designed. I hope and I believe that this Template will your Time.
02 Contents
Get a modern PowerPoint Presentation that is beautifully designed. I hope and I believe that this Template will your Time.
03 Contents
Get a modern PowerPoint Presentation that is beautifully designed. I hope and I believe that this Template will your Time.
04 Contents
Get a modern PowerPoint Presentation that is beautifully designed. I hope and I believe that this Template will your Time.

Figure 4 Navigation to the Previous Slide

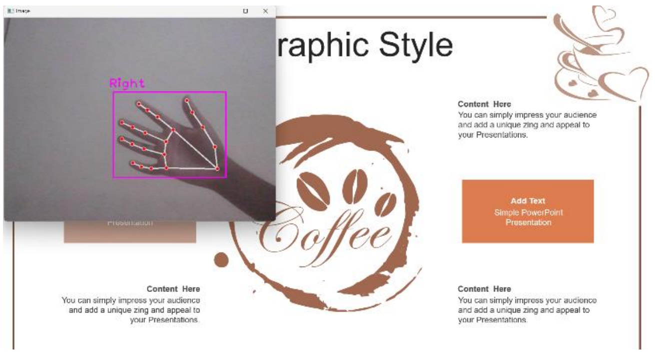
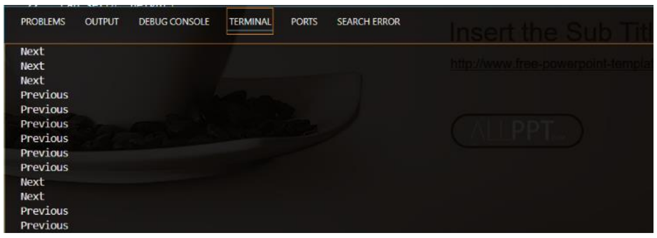

Right
raphic Style
Content Here
You can simply impress your audience and add a unique zing and appeal to your Presentations.
Add Text
Simple PowerPoint Presentation
Coffee
Content Here
You can simply impress your audience and add a unique zing and appeal to your Presentations.
Content Here
You can simply impress your audience and add a unique zing and appeal to your Presentations.

Figure 5 Navigation to the Next Slide

D. Real-Time Gesture Feedback in Slide Navigation

The *Figure 6* shows the console logging mechanism provides real-time feedback to the user based on the system's accurate detection of specific hand gestures. When Gesture 1 (thumbs up) or Gesture 2 (whole hand pointing up) is correctly recognized, the system generates corresponding feedback messages such as "Previous" or "Next," respectively.



```
Next
Next
Next
Previous
Previous
Previous
Previous
Previous
Previous
Next
Next
Previous
Previous
```

Figure 6 Real-Time Gesture Feedback System

Photo capture condition	Accuracy
Dark background	20 - 30 %
Light background	80 - 90 %
With proper lightning	80 - 95 %
Without proper lightning	30 – 50 %

VI. Discussion

The proposed system leverages the naturalness and convenience of hand gestures in interaction, which are integral to body language and require no additional devices. By employing an AI-based hand gesture detection methodology, the system aims to enhance the presenter's ability to navigate slides seamlessly during presentations. While the project demonstrated promising accuracy in recognizing gestures like "thumbs up" and "pointing up," it encountered challenges in real-world settings such as varying lighting and background conditions. To address this, integrating machine learning techniques for diverse gesture training and considering depth-sensing technologies could improve robustness and adaptability.

VII. Conclusion

In conclusion, the project successfully explored the use of hand gestures as a natural and intuitive method for controlling presentations. Despite challenges in real-world scenarios, such as lighting variations impacting gesture recognition accuracy, the system's AI-based approach showed promise with an overall accuracy of 80-90 %. Integrating machine learning and depth-sensing technologies could further enhance its robustness. Overall, the project lays a foundation for future developments in gesture-based presentation control, offering a convenient and device-free interaction option for presenters.

VIII. Future Scope

The project's future prospects are expansive, encompassing the integration of augmented reality (AR) and virtual reality (VR) technologies to create immersive presentation experiences with intuitive hand gesture controls. This advancement not only enhances engagement during presentations but also sets the stage for innovative applications in interactive learning and training environments. Moreover, the project's focus on expanding gesture recognition capabilities to include a broader range of hand signs and gestures caters to individuals unable to speak,

such as mute individuals, empowering them to communicate effectively and inclusively. This project is allowing for seamless interaction through voice commands alongside hand gestures, thereby catering to diverse user preferences and accessibility needs. This holistic approach positions the project at the forefront of human-computer interaction advancements, promoting accessibility, engagement, and user empowerment in immersive interaction platforms.

IX. References

[1] ITM Web of Conferences, vol. 44, p. 03031, 2022. DOI: 10.1051/itmconf/20224403031. Conference Name: ICACC-2022.

[2] Indonesian Journal of Electrical Engineering and Computer Science, vol. 18, no. 1, pp. 49-55, Apr. 2020. DOI: 10.11591/ijeecs.v18.i1.pp49-55. ISSN: 2502-4752.

[3] IEEE, ISBN: 978-1-7281-2791-0, ©2020.

[4] International Journal of Computer Applications Technology and Research, vol. 11, no. 12, pp. 464-469, 2022. ISSN: 2319-8656. DOI: 10.7753/IJCATR1112.1012.

[5] IEEE, Digital Object Identifier: 10.1109/ACCESS.2020.3032140.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[6] Hindawi, "Complexity," vol. 2021, article ID 8812542, pp. 1-18, 2021. doi: 10.1155/2021/8812542.

Appendix :

```
import win32com.client
from cvzone.HandTrackingModule import HandDetector
import cv2
import os
import numpy as np
import aspose.slides as slides
Application = win32com.client.Dispatch("PowerPoint.Application" )
Presentation =
Application.Presentations.Open("C:\\Users\\91942\\Desktop\\CURRENT\\ML\\ML_preoject\\PPT-
Presentation-controlled-by-hand-gesture-main\\zani.pptx")
# print(Presentation.Name)
Presentation.SlideShowSettings.Run()
# Parameters
width, height = 900, 720
gestureThreshold = 300
# Camera Setup
cap = cv2.VideoCapture(0)
cap.set(3, width)
cap.set(4, height)
# Hand Detector
detectorHand = HandDetector(detectionCon=0.8, maxHands=1)
# Variables
imgList = []
delay = 30
buttonPressed = False
counter = 0
drawMode = False
imgNumber = 20
delayCounter = 0
annotations = [[]]
annotationNumber = -1
annotationStart = False
while True:
# Get image frame
success, img = cap.read()
# Find the hand and its landmarks
hands, img = detectorHand.findHands(img)  # with draw
if hands and buttonPressed is False:  # If hand is detected
hand = hands[0]
cx, cy = hand["center"]
lmList = hand["lmList"]  # List of 21 Landmark points
fingers = detectorHand.fingersUp(hand)  # List of which fingers are up
if cy <= gestureThreshold:  # If hand is at the height of the face
if fingers == [1, 1, 1, 1, 1]:
print("Next")
buttonPressed = True
if imgNumber > 0:
Presentation.SlideShowWindow.View.Next()
imgNumber -= 1
annotations = [[]]
```

```
annotationNumber = -1
annotationStart = False
if fingers == [1, 0, 0, 0, 0]:
print("Previous")
buttonPressed = True
if imgNumber >0 :
Presentation.SlideShowWindow.View.Previous()
imgNumber += 1
annotations = [[]]
annotationNumber = -1
annotationStart = False

else:
annotationStart = False

if buttonPressed:
counter += 1
if counter > delay:
counter = 0
buttonPressed = False

for i, annotation in enumerate(annotations):
for j in range(len(annotation)):
if j != 0:
cv2.line(imgCurrent, annotation[j - 1], annotation[j], (0, 0, 200), 12)

cv2.imshow("Image", img)

key = cv2.waitKey(1)
if key == ord('q'):
break
```