Gesture-Enhanced Presentation Control for Education

Saumya Y M
Dept. of CSE
St Joseph Engineering College
Vamanjoor, India
saumyam@sjec.ac.in

JaishmaKumari B
Dept. of CSE
St Joseph Engineering College
Vamanjoor, India
jaishmab@sjec.ac.in

Colin Christon DCruz
Dept. of CSE
St Joseph Engineering College
Vamanjoor, India
colinchriston@gmail.com

Austin Dsouza
Dept. of CSE
St Joseph Engineering College
Vamanjoor, India
austindsz21@gmail.com

Daniel Loy Braggs
Dept. of CSE
St Joseph Engineering College
Vamanjoor, India
danielloy675@gmail.com

Elwin Jason Pereira
Dept. of CSE
St Joseph Engineering College
Vamanjoor, India
elwinjpereira02@gmail.com

***Abstract*—Presentation skills are vital in many areas of life. Giving presentations is probably a common experience for anyone, whether they are a worker, student, business owner, or employee of an organisation. The requirement to manage and manipulate the slides with a keyboard or other specialised device might make presentations seem tedious at times. Enabling users to control the slideshow with hand gestures is the aim of this work. Gestures have become increasingly common in human-computer interaction in recent years. Several PowerPoint functionalities have been attempted to be controlled by hand movements by the system. This system maps motions using multiple Python modules and uses machine learning to identify motions with minute variances. Creating the perfect presentation is becoming increasingly difficult due to a number of aspects, including the slides, the keys to switching the slides, and the audience's composure. An intelligent presentation system that is based on hand gestures makes it simple to update or modify the slides. Allowing viewers to explore and manipulate the slideshow with hand movements is the technology's main objective. The technique recognises various hand motions for a variety of tasks using machine learning. A means of recognition opens up a line of communication between people and machines.**

Index Terms*—Gesture, Gesture Recognition, Human Computer Interaction, Presentation, Annotation, Slide change.

I. Introduction

In today's ever-evolving education landscape, traditional classroom presentations are under- going a digital transformation. As digital learning gains prominence, there is a pressing need for a more dynamic and engaging means of controlling presentations. The existing tools not only limit the interactive potential of educators but also create accessibility challenges, particularly for those with physical disabilities. These issues hinder the effectiveness of teaching and can disrupt the flow of lessons [1]. Educators seek innovative ways to engage students using technology, making the "Gesture-Enhanced Presentation Control for Education" a highly relevant task.

979-8-3503-5059-3/24/$31.00 ©2024 IEEE

In AR/VR, hand tracking is essential for facilitating natural engagement and communication [10], and it has been a subject of intense discussion in the field of study. For many years, research has been conducted on vision-based hand pose estimation [2].The slides are editable by users. The interactive presentation system creates a more useful and approachable user interface for manipulating presentation displays by utilising state-of-the-art human-computer interaction techniques. When these hand gesture choices are used in place of a traditional mouse and keyboard control, the presentation experience is substantially improved. Nonverbal communication refers to the use of body language and gestures to convey a certain message. The Python framework was primarily employed in the construction of the system, together with NumPy, MediaPipe, openCV, and CV zone technologies. The goal of this approach is to improve presentations' usefulness and efficiency [5].

II. Literature Review

The paper [6] offers a thorough examination of computer vision based hand gesture recognition system. From mathematical algorithms like the row vector to machine learning approaches, it critically analyzes strengths and limitations. By scrutinizing methods such as edged image analysis and vector passing, the survey identifies research gaps and showcases deficiencies. This foundation justifies the chosen techniques, providing vital background and directionality. It aids in positioning the paper's contributions by benchmarking current challenges and showcasing field deficiencies, delivering crucial insights for hand gesture recognition understanding and system improvement. They employed the Row Vector Algorithm, the Diagonal Sum Algorithm, the Mean and Standard Deviation of the Edged Image, and the Edging and Row Vector Passing Algorithm.

This paper [2] introduces a system for controlling PowerPoint slides through hand gestures using a combination of a thermal camera and a webcam for robust hand tracking.

The methodology covers illumination invariant hand region extraction, gesture recognition through skin segmentation and SVM classification, and slide control mappings. Experiments demonstrate high accuracy in classifying gestures like swipe left and right to switch slides. The literature review analyzes existing research in gesture recognition and hand tracking, identifying challenges in accuracy and processing lag. The conclusion sums up key innovation, potential applications in interactive presentations, and limitations like small gesture vocabulary, suggesting enhancements through multidimensional dynamic time warping. The methodology included OpenCV, Haar Cascade Classifier, Skin Color Segementation, and Gesture Recognition.

This paper [3] introduces an innovative vision based hand gesture recognition system designed for PowerPoint presentation control, encompassing both static and dynamic gestures. The literature review scrutinizes existing methodologies, highlighting limitations in accessibility and vocabulary across various approaches. Leveraging a sophisticated 7-layer convolutional neural network (CNN) built upon the 20BN-Jester baseline, the system extracts spatial and temporal features from dynamic hand gesture video frames, significantly improving accuracy. The training process, conducted on the 20BN-Jester dataset using PyTorch on a GPU system, results in a highly accurate model capable of real-time classification. The methodology involves multi-phase processing, from opening a PowerPoint presentation to capturing live webcam video of hand gestures, transformed into a 20-frame image array for classification. Python is the chosen programming language, with Tkinter for GUI, PyTorch for deep learning, OpenCV for computer vision, and PyAutoGUI for simulating virtual keyboard keypresses, ensuring a robust and versatile integration of functionalities. The system recognizes diverse gestures contributing to enhanced accessibility and user-friendliness in PowerPoint presentations. Their methodology used Convolutional Neural Network Architecture, 20BN-Jester dataset, multi-phase approach, PyTorch and PyAutoGU.

A hand gesture-controlled virtual mouse system for seamless human-computer interaction is presented in this paper [4]. Using a webcam to record the user's hand movements, the system uses computer vision and machine learning models to detect and identify pointing and clicking actions in real-time. These predicted hand poses are seamlessly translated into virtual cursor operations, allowing touchless spatial control.The literature review traces the evolution of gesture recognition techniques from initial glove-based tools to modern solutions, analyzing pros and cons of past approaches. The proposed methodology addresses limitations like hardware restrictions and system lag by blending MediaPipe, speech recognition, and natural language processing for an efficient and responsive interface. The Algorithms and Tools used here include Google's MediaPipe, Single Shot Detector model, Hand Landmark model.

This paper [5] presents a system for controlling presentations using hand gestures, built using OpenCV and Google's MediaPipe framework. A webcam captures video input of the user's hand gestures, which are recognized by MediaPipe. Specific gestures like raising different numbers of fingers are then mapped to control commands for the presentation - changing between slides, accessing a pointer to draw on slides, and erasing drawings. The main technical challenge discussed is accurately recognizing gestures with background noise and variations in lighting. The system is designed to provide an intuitive hands free way of controlling presentations that could be used in real-world scenarios with basic hardware. Key libraries utilized include OpenCV for image processing and frame detection, MediaPipe for gesture recognition, and NumPy for numeric computing to transform the inputs into outputs. Overall, it demonstrates a practical application of computer vision and gesture recognition to facilitate more natural human-computer interaction. The Algorithms and Tools used here are BlazePalm, Hand Landmark Model, Hidden Markov Models (HMM), K-means clustering, Fast Fourier Transform (FFT), Non-maximum suppression and Encoder-decoder models.

The so-called Virtual Whiteboard, which is based on electronic pens and sensors, is given in the paper [8] and may offer an alternative to contemporary electronic whiteboards. With the tool in hand, the user can write, draw, and manipulate the contents of the whiteboard with just his or her hands. It is not necessary to have extra equipment like infrared diodes, infrared cameras, or cyber gloves. Dynamic hand gesture recognition is the foundation for user interaction with the Virtual Whiteboard computer application. When examining a video feed from a webcam connected to a multimedia projector that displays content from a whiteboard, gestures are identified. Kalman filtering helps to track the positions of hands in the image. In the paper the hardware and software of the Virtual Whiteboard is discussed with a special focus on applying Kalman filters for prediction of successive hand locations. The effectiveness of Kalman filter-supported recognition was evaluated for the motions used to manage the contents of the whiteboard, and the efficiency without filtering is provided.

The problem of estimating the entire 3D hand shape and pose from a single RGB image is a new and difficult one that is tackled in this study [7]. The majority of existing techniques for 3D hand analysis from monocular RGB images are limited to guessing the 3D positions of hand keypoints; they are unable to accurately convey the 3D shape of the hand. On the other hand, the research describes an approach based on Graph Convolutional Neural Networks (Graph CNNs) that can reconstruct a complete 3D mesh of the hand surface, which includes more detailed information on the 3D shape and attitude of the hand. They provide a large-scale synthetic dataset comprising both 3D postures and ground truth 3D meshes in order to train networks under complete supervision. Using the depth map as a weak supervision in training, the researcher presented a weakly supervised method for fine-tuning the networks using real-world datasets without 3D ground truth. Through rigorous evaluations on their suggested new datasets and two public datasets, proposed research indicate that proposed technique can build accurate and reasonable 3D hand mesh and can accomplish superior 3D hand pose estimate accuracy when compared with state-of-the-art methods. The difficulties faced by patients receiving physical therapy are discussed in the paper [9], with a focus on the boredom of repeating exercises that may cause patients to lose enthusiasm. It offers a remedy in the shape of hand rehabilitation software, which makes use of hand gesture detection and recognition technologies to enhance patient engagement and enjoyment during rehabilitation. The MediaPipe Hands algorithm is used by the system to recognise gestures and detect hands.

The study [11] uses morphological processing and YCbCr thresholding to accomplish efficient gesture recognition for PowerPoint presentation control. The Hidden Markov Model is used to classify the gestures that have been identified. HMM is a statistical model that works well for tasks involving the recognition of patterns over extended periods of time.

The purpose of the paper [12] is to enable gesture-based control of PowerPoint presentations, and it does so by using multiple techniques. Machine learning algorithms are used in the study to identify and categorise hand gestures. By training the model to distinguish minor changes in movements, the system can accurately map these motions to specific actions, such as advancing or reversing slides. The Python programming language is used to implement the system, making use of Mediapipe and OpenCV packages.

The paper [13] provides a new way for controlling PowerPoint presentations using static hand gestures. This technique uses a webcam to record hand motions, making it a useful and user-friendly solution. The thinning method, a method for processing and analysing hand forms, is introduced in this study. The number of elevated fingers is determined by using the hand form parameters that are extracted using this procedure. This novel method improves gesture recognition precision. The fact that the suggested approach doesn't need any extra gear, like gloves, markers, or other gadgets, is one of its best qualities. This improves the system's accessibility and usability by enabling users to interact with their presentations using just their hands.

III. System Design

A. Architectural Diagram

The architectural design for gesture-enhanced presentation control as displayed in the above Fig. 1 begins with the webcam capturing the user's hand gestures, serving as the primary input method. OpenCV processes the video feed, extracting critical details such as hand position and shape. These details are then analyzed by MediaPipe, which employs sophisticated algorithms to recognize specific gestures based on predefined patterns. Following recognition, the identified gestures are relayed back to the presentation software, where they are interpreted into actions such as navigating slides or activating multimedia elements. This process involves several intermediary steps, including video capturing, framing, and hand detection as well as frames filtering to enhance accuracy. Feature extraction distills relevant information from the recognized gestures, which are then classified into predefined actions. The architecture further encompasses gesture mapping, where these classified gestures are matched with corresponding presentation control functions. Ultimately, the presentation control component interfaces seamlessly with the software, executing the mapped functions based on the recognized gestures. Throughout this interaction, the user plays a central role, activating the webcam input device and performing hand gestures within its view to control the presentation. Feedback mechanisms such as audible or visual signals confirm gesture recognition and execution ensuring a smooth and intuitive user experience.

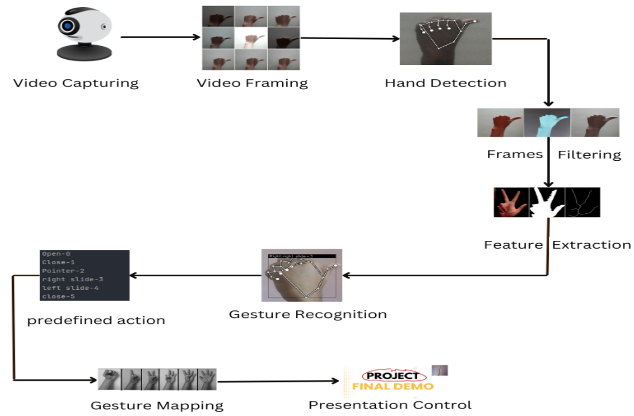


Fig. 1. Architecture Diagram

B. Phases of Gesture Recognition

- **Phase 1: Video Acquisition (Webcam):** This phase involves capturing the video stream from a webcam or any other camera input device. The quality and resolution of the captured video are crucial for accurate hand gesture recognition. The video stream serves as the input for subsequent phases in the gesture recognition system.
- **Phase 2: Video Pre-processing:** Video pre-processing is essential for preparing the captured video stream for hand gesture recognition. This phase typically includes several tasks such as:
  1) **Frame Extraction:** The continuous video stream is divided into individual frames for analysis.
  2) **Background Removal:** Removing the background from each frame helps isolate the hands from the rest of the scene, reducing interference and improving accuracy.
  3) **Hand Region Detection:** Identifying and delineating the regions of interest containing the hands within each frame. This can involve techniques like skin tone detection or background subtraction to locate the hands within the frame accurately.
- **Phase 3: Feature Extraction:** In this phase, relevant features are filtered and extracted from the detected hand region. These features provide the basis for identifying and interpreting different hand gestures. Frame filtering tasks may include:

1) **Frame Rate Reduction:**The system can sample the video at a reduced frame rate, such as every second or third frame, to focus on key points in time where meaningful gestures occur. This eliminates redundant data and allows the model to concentrate on frames that contain significant hand movements.
2) **Background Subtraction:**Background subtraction techniques are applied to isolate the hand region from the background. This helps to filter out irrelevant objects and noise, ensuring that only the hand gesture is processed. Common methods include Gaussian Mixture Models (GMM) or simple thresholding techniques that detect motion in the foreground.
3) **Blurring and Smoothing:**Applying blurring or smoothing filters (such as Gaussian blur) to the frames can help remove minor noise or irregularities in the video feed. This step enhances the quality of the input image, making the subsequent feature extraction process more robust.
4) **Skin Detection and Masking:** Skin detection algorithms can be applied to identify regions of the frame corresponding to human skin tones, focusing specifically on hand regions. This creates a mask that highlights the hand while ignoring non-skin areas, leading to more precise hand detection and feature extraction.

Feature extraction tasks may include:

1) **Hand Pose Estimation:** Determining the orientation and configuration of the hand(s) in the frame, including finger positions and hand shape. These landmarks provide a detailed map of the hand's orientation, configuration, and shape. Algorithms like MediaPipe Hands can accurately detect these landmarks in real time.
2) **Finger Tracking:** Tracking the movement and position of individual fingers within the hand region. By tracking the movement of each finger over time, the system can recognize dynamic gestures, such as a finger swipe or a specific finger motion sequence.
3) **Motion Trajectory Analysis:** Analyzing the trajectory of hand movements over time to detect gestures involving motion, such as swipes or gestures with directional components.Recognizing when a gesture starts and ends is essential for temporal gestures. This involves detecting the initial movement and when the hand comes to rest or returns to a neutral position.

- **Phase 4: Gesture Recognition:** Gesture recognition involves two main steps:
  1) **Gesture Classification:** Classifying the extracted features into specific gesture classes or commands. This can be achieved using machine learning models such as convolutional neural networks (CNNs) or rule-based algorithms.
  2) **Gesture Mapping:** Once gestures are classified, they are mapped to corresponding presentation control functions. For example, a specific hand pose or motion trajectory might correspond to commands like next slide, previous slide, or activate pointer mode.
- **Phase 5: Presentation Control:** In this final phase, the recognized gestures are used to control presentation software such as PowerPoint or Google Slides. This includes executing the mapped presentation control functions based on the recognized gestures, enabling seamless interaction with the presentation content. Presentation control functions may include slide navigation, pointer control, annotation or drawing tools activation, and other interactive features. The system interfaces with the presentation software through appropriate APIs or communication protocols to facilitate these actions.

IV. Implementation

The implementation of Gesture-Enhanced Presentation system commenced with the pivotal task of data collection and preprocessing. This is followed by selection and training the model for gesture recognition.

A. Data Collection and Preprocessing:

- **Data Collection:** A diverse set of hand gesture images representing various presentation commands like "Next Slide", "Previous Slide", "Start Presentation" and "Stop Presentation" is collected. Ensuring diversity, the dataset covers a wide range of hand shapes, positions, and lighting conditions. Quality assurance was maintained throughout the process to minimize noise and ensure clarity for effective model training.
- **Data Labeling:** Each image underwent a labeling process, where the images were manually annotated with the corresponding gesture it represents, including gestures like "Next Slide", "Previous Slide" and others. Key points within the hand gestures, such as the position of the index finger, were also labeled to provide crucial information for model training. Consistency in labeling was paramount to avoid confusion during model training and evaluation.
- **Data Preprocessing:** Before training the model, the collected dataset was preprocessed to enhance image quality and remove noise. This involved resizing images to a standard size, normalizing pixel values for consistent brightness and contrast, and applying various data augmentation techniques to increase dataset diversity. Relevant features, such as identifying landmarks or keypoints like the coordinates of the index finger, were extracted for effective gesture recognition.
  - **Model Selection:** For the classification task TensorFlow's Keras API was opted. This choice was driven by the availability of pre-built deep learning models tailored for gesture recognition. By utilizing this framework, the model was able to efficiently

utilize computational resources and simplify the development process.

- **Training Data:** After preprocessing, the dataset was divided into training and validation sets. The goal here was to ensure a fair distribution of samples across different gesture classes, which is crucial for the model to generalize well to new, unseen data.
- **Model Training:** With TensorFlow as the training platform, a structured approach was followed. Techniques like transfer learning and fine-tuning of pre-trained models was applied to optimize the model's performance, especially given constraints such as limited training data or computational resources.
- **KeyPoint Classifier:** To facilitate the gesture recognition, the KeyPointClassifier class was implemented. This supported the deployment of a TensorFlow Lite interpreter, specifically configured with the chosen model file and thread specifications. With this setup, the landmark coordinates could be taken as input resulting in the accurate prediction of the class index, enabling precise classification of hand gestures based on these key points.

B. *Real-time Gesture Recognition:*

- **Camera Integration:** Integrate a camera module (e.g., webcam) with the system to capture real-time video input.
- **Gesture Detection:** Utilized libraries like OpenCV and MediaPipe to detect and track hand gestures in real-time video streams. Apply the trained gesture recognition model to classify detected gestures.
- **Feedback Mechanism:** Provide visual feedback to the user in real-time, indicating the recognized gesture and corresponding action.

C. *GUI Development:*

- **Graphical User Interface:** A user-friendly GUI was implemented using Tkinter framework, featuring an intuitive interface that enables users to interact with the presentation software using hand gestures.
- **Control Elements:** Implemented control elements such as buttons or sliders for common presentation functions (e.g., next slide, previous slide, start/stop presentation).
- **Integration with Gesture Recognition:** Integrated the gesture recognition module with the GUI, ensuring seamless interaction between gesture input and presentation control.

D. *System Integration and Testing:*

- **Component Integration and Function Testing:** All the components of the system, including gesture recognition, GUI, and presentation control logic are integrated. This is followed by a thorough testing to ensure the system functions as expected in different scenarios and environments. Test for accuracy, responsiveness, and robustness to variations in lighting conditions and hand gestures are performed.

Fig. 2. GUI without input pptx file

V. Results and Discussion

The results depict the Graphical User Interface (GUI) of the proposed system as displayed in the Fig. 2, illustrating the initial state with no file selected. Additionally, it showcases an array of recognized hand gestures and their corresponding actions seamlessly integrated into the interface.

Users can seamlessly navigate through slides, move left or right, annotate slides with a red-colored line, and erase annotations as needed, providing a dynamic and engaging presentation experience.

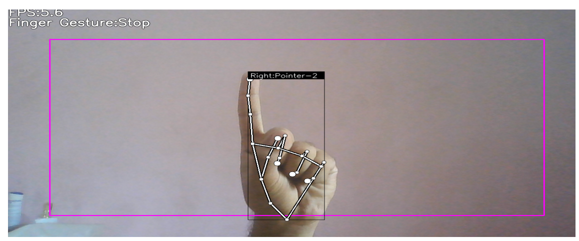


Fig. 3. Pointer to move cursor

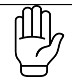
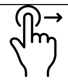
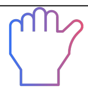
In Fig. 3, the background image detection with landmarks is showcased, highlighting the system's ability to accurately detect gestures and provide real-time feedback to the user. This functionality empowers programmers with a flexible approach to working with gestures, ensuring precise recognition and seamless integration into presentation control. TABLE I provides a comprehensive overview of the different gestures supported by the proposed system. The table outlines the specific actions associated with each gesture, ensuring users understand how to utilize them. This detailed description facilitates a deeper comprehension of each gesture's functionality and its intended role within the proposed system.

VI. Future Work

Despite the successful implementation of the Gesture-Enhanced Presentation system, there are several avenues for future work and enhancements:

1) **Integration with Voice Commands:** Expanding the system to support voice commands alongside hand

TABLE I
GESTURES SUPPORTED BY THE PROPOSED SYSTEM

Gesture	Action Performed	Description
	Switch Between Annotation and Pointer	This gesture allows users to smoothly transition between annotation and pointer modes, as well as back and forth from pointer to annotation mode.
	Clear Annotation	This gesture serves the purpose of clearing annotations made by the user previously.
	Mouse Pointer	This gesture serves a dual purpose: first, it enables users to navigate the cursor during presentations, facilitating seamless control and highlighting of key areas. Additionally, it empowers users to make annotations, jot down notes, and mark significant sections within the presentation, enhancing engagement and interaction.
	Next Slide	This gesture enables users to seamlessly transition to the next slide during presentations.
	Previous Slide	This gesture enables users to seamlessly transition to the previous slide during presentations.
	Exit Presentation	This gesture provides a way to the user to exit the presentation.

gestures would provide users with additional control options and further enhance the user experience. Integrating voice recognition technology would enable presenters to navigate slides and execute commands using natural language.

2) **Enhanced Gesture Recognition:** Continuously improving the accuracy and robustness of gesture recognition algorithms is crucial for ensuring reliable performance across different environments and hand poses. Further research and development in this area could involve exploring advanced machine learning techniques and leveraging larger datasets for training.
3) **Multi-Modal Interaction:** Exploring the integration of multiple modalities such as hand gestures, voice commands and facial expressions could lead to more immersive and interactive presentation experiences. By combining different input modalities one can create a more versatile and adaptable system that caters to a wider range of user preferences and abilities.

VII. Conclusion

In conclusion, the development of the Gesture-Enhanced Presentation system for education has been a significant endeavor aimed at revolutionizing the way educators and students interact with presentation materials. By leveraging hand gesture recognition technology, a user-friendly interface that allows presenters to control presentation slides seamlessly using intuitive gestures is proposed. This system offers an innovative and engaging approach to delivering educational content, enhancing the learning experience for both presenters and audiences. Through rigorous testing and iterative design improvements, it is ensured that the system meets the requirements of educational settings and delivers reliable performance.

References

[1] Bobo Zeng, Guijin Wang, Xinggang Lin. ”A Hand Gesture Based Interactive Presentation System Utilizing Heterogeneous Cameras ”. TSINGHUA SCIENCE AND TECHNOLOGY ISSNll1007-0214ll15/18llpp329-336 Volume 17, Number 3, June 2012.

[2] Rida Zahra , Afifa Shehzadi , Muhammad Imran Sharif , Asif Karim , Sami Azam , Friso De Boer , Mirjam Jonkman , Mehwish Mehmood.”Camera-based interactive wall display using hand gesture recognition”,Computational Intelligence and Neuroscience, 2022.

[3] Muhammad Idrees , Ashfaq Ahmad , Muhammad Arif Butt , and Hafiz Muhammad Danish. ”Controlling Power Point using Hand Gestures in Python”. PWebology (ISSN: 1735-188X) Volume 18, Number 6, 2021.

[4] M. Kasar, P. Kavimandan, T. Suryawanshi, and S. Abbad, “AI-based real-time hand gesture-controlled virtual mouse,” Australian Journal of Electrical and Electronics Engineering, pp. 1–10, Feb. 2024, doi: 10.1080/1448837x.2024.2313818.

[5] Hajeera Khanum, Dr. Pramod H B.” Smart Presentation Control by Hand Gestures Using Computer Vision and Google’s MediaPipe”, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 09 Issue: 07 — July 2022 .

[6] Munir Oudah , Ali Al-Naji and Javaan Chahl “Hand Gesture Recognition Based on Computer Vision: A Review of Techniques”. IEEE conference on computer vision - 23 July 2020

[7] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. ”3D Hand shape and Pose Estimation from a Single RGB image”. In Proceedings of the IEEE conference on computer vision and pattern recog- nition, pages 10833–10842, 2019.

[8] M. Lech, B. Kostek, and A. Czyzewski, “Virtual Whiteboard: A gesture-controlled pen-free tool emulating school whiteboard,” Intelligent Decision Technologies, vol. 6, no. 2, pp. 161–169, Feb. 2012, doi: 10.3233/idt-2012-0132.

[9] Yeng, Angelina Chow Mei, et al. ”Hand Gesture Controlled Game for Hand Rehabilitation.” International Conference on Computer, Information Technology and Intelligent Computing (CITIC 2022). Atlantis Press, 2022.

[10] H.S. Shrisha, V. Anupama, ”NVS-GAN: Benefit of generative adversarial network on novel view synthesis”, International Journal of Intelligent Networks, Volume 5, 2024, 184-195,doi.org/10.1016/j.ijin.2024.04.002.

[11] Cahya, Rahmad., Arief, Prasetyo., Riza, Awwalul, Baqy. ”PowerPoint slideshow navigation control with hand gestures using Hidden Markov Model method.” 12 (2022).:7-18. doi: 10.31940/matrix.v12i1.7-18

[12] K., P., Kumari., Bandaram, Bharath, Goud., Kalvakuntla, Sumana., Bathula, Naresh., Bellamkonda, Harish. ”Automated Gesture Controlled Presentation Using Machine Learning.” International Journal For Science Technology And Engineering, 10 (2022).:1248-1251. doi: 10.22214/ijraset.2022.47517

[13] Savitha, M. ”Static Hand Gesture Recognition for PowerPoint Presentation Navigation using Thinning Method.” International Journal on Recent and Innovation Trends in Computing and Communication, 6 (2018).:187-189.