International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Gesture-Based Interaction for Enhanced Presentation Control

Nayim A Dalawai[1], Prathamraj P B[2], Shashank S Padasalagi[3], Musaddiqua Rajannavar[4], Dayananda B S[5], Suresh R[5] Jyothi A P[6]

[1,2]Student in Dept of MME, Ramaiah University of Applied Sciences, Bengaluru, Karnataka, India
[3]Student in Dept of CSE, Ramaiah University of Applied Sciences, Bengaluru, Karnataka, India
[4]Student in Dept of CSE, KLE college of Engineering and Technology, Belagavi, Karnataka, India
[5]Faculty in Dept of MME, Ramaiah University of Applied Sciences, Bengaluru, Karnataka, India
[6]Faculty in Dept of CSE, Ramaiah University of Applied Sciences, Bengaluru, Karnataka, India

ABSTRACT:
In recent years, a natural interaction method has been the main area of study in the subject of Human Computer Interaction. It was observed that the computers have adopted real-time hand gesture based recognition techniques for numerous applications. A vision system is required for hand motion detection. The main method of interaction entails building a virtual Human Computer Interaction (HCI) device using a web camera. The hand gestures used in modern vision based HCI techniques are investigated in this paper. Using the developed HCI user can remotely operate the presentations.

KEYWORDS: Hand tracking, Computer vision, Image processing, Gesture recognition, PowerPoint presentation, User interaction, Hand landmarks, OpenCV, numpy, Camera setup, Image display, Finger detection, Button controls, Annotation, Image manipulation, Real-time processing, User interface, Image resizing, Landmark tracking, Hand gestures, Hidden Markov Model (HMM).

INTRODUCTION

The interactions with the environment and surroundings through nonverbal communication would not be possible without the use of hand gestures. Understanding and classifying gestures might help in deciphering their meanings. Additionally, the classification of hand gestures is a crucial technique in human-computer interaction. Particularly in virtual reality applications, these gestures can be utilized to operate gadgets in the workplace and act as a substitute to conventional input devices like a mouse, keyboard, laser pointers etc

Modern human-computer interaction techniques are used by the interactive presentation system to create a more useful and approachable interface for managing presentation displays. When these hand motion choices are used during presentations instead of a traditional mouse and keyboard control, the experience is substantially improved. Nonverbal or non-vocal communication involves the use of body language, such as gestures, to effectively convey a specific message. The Python framework and other tools including an open CV, a cv zone, NumPy, hmmlearn is a set of algorithms for unsupervised learning, inference of Hidden Models, Matplotlib is a cross platform, data visualization and graphical plotting library, Seaborn is a library that uses matplotlib underneath to plot graphs and media pipe were

mostly used to build the program. This technique aims to make presentations more useful and effective. The program uses movements to write, undo, and get the cursor on various text sections. An attempt is made to enable hand motion control for the slideshow, which enables to improve the user experience.
Minor gesture variations done can be identified through the use of machine learning, and they have been converted into some fundamental techniques to manipulate presentation slide shows using Python language. Different gestures, such as showing index finger, index and middle finger, and little finger and others, can be used to manage and control slides. For a standard presentation flow, this program uses a hand gesture-based human-machine interface. The program has undergone active development over the past few years to make rapid and simple motion image-based methods recognise dynamic hand movements. Users can control presentations using this method in a more convenient, sensible, and natural way. The HMM gesture recognition method is inspired by the success of HMM in speech recognition problems. The similarity between speech and hand gestures shows that strategies that work for one problem will also work for another. More importantly, HMMs can model physical events in the body. Since orientation is a continuous phenomenon of continuous time, HMM is a suitable choice for orientation recognition.
What is gesture: A gesture is a nonverbal form of expression that involves bodily gestures, stances, or actions. It involves expressing a message, intention, or feelings by using various body parts, especially the hands. Hand gestures, face expressions, body language, and other physical indicators are examples of gestures. They are essential to human communication because they enable people to engage with one another, express themselves, and share information without relying primarily on spoken or written language. Across cultures and circumstances, gestures can vary, and they frequently have cultural or contextual connotations that are well-understood by a particular group of individuals.
Why gesture: The user's hand movements are detected and tracked using a hand detector. The interaction mechanism in this project is absolutely dependent on hand gestures. The code allows the user to control and navigate the PowerPoint slideshow by recognising and analyzing the hand movements. The presentation slides can be interacted with naturally and simply using hand gestures. The use of hand gestures instead of conventional input methods like a keyboard or mouse offers a more immersive and interesting approach to engage with the presentation. The hand detector module precisely maps the finger placements, landmarks, and hand movements to certain functions for directing the slideshow.

RELATED WORK

There are various methods for hand gesture identification, such as picture feature extraction and AI-based classifiers, that have been investigated in prior research. Effective strategies, utilizing instruments like picture feature extraction and classification algorithms, are needed to extract and recognise small movements and motions in 2D and 3D hand gestures [1]. On using hand gestures to control PowerPoint presentations. Subtle motions are detected and distinguished using machine learning algorithms, and then they are mapped to standard slideshow functions. The study seeks to improve the interactivity and user experience of PowerPoint presentations by employing this strategy with Python[2]. The uses a variety of real-world and in-the-moment commercial application scenarios to graphically show all the promising and possible machine learning (ML) and deep learning (DL) methods. It is possible to enable machines and gadgets to learn for themselves and act intelligently. Personalized, prognostic, predictive, and prescriptive insights can be obtained by combining real-time and runtime data with big data.[3]

The increasing focus on natural interaction methods in Human-Computer Interaction (HCI) and the widespread adoption of real-time hand gesture-based recognition applications in various computer environments. It mentions the necessity of a camera for hand motion detection and emphasizes the use of a web camera to create a virtual HCI device. The main objective of the research paper is to investigate the hand gestures utilized in modern vision-based HCI techniques [4]. In recent years there has been a significant increase in the utilization of hand gestures for controlling and managing presentations, addressing the limitations posed by keyboards and specialized gadgets in traditional slideshow interaction methods [4].Worldwide military efforts are actively incorporating AI technology to enhance decision-making, strategic planning, and adversary capabilities. However, research is needed to address cybersecurity threats and optimize performance in targeting and tracking. AI offers the potential to revolutionize military operations by enabling faster information analysis and empowering certain countries with a tactical advantage [5].

In today's digital age, presenting a slideshow during a presentation is an attractive and efficient technique to help speakers persuade the audience and deliver information. Slides can be controlled with tools like a mouse, keypad, laser pointer, etc. The disadvantage is that managing the devices requires prior device knowledge. A few years ago, gesture detection became increasingly important for controlling software like media players, robots, and games. The utilization of gloves, markers, and other objects is enhanced by the hand gesture recognition system. However, the use of such gloves or markers increases the system's cost. The proposed hand gesture detection technology is based on artificial intelligence[6].

Machine learning (ML) and deep learning (DL) are subsets of artificial intelligence, with DL being the most human-like. Cloud computing complements ML/DL by providing on-demand resources, enabling efficient utilization of servers. This book chapter explores the adoption of ML/DL in the cloud, covering their concepts, algorithms, and projects such as cervical cancer detection. It also highlights applications like cognitive cloud, IoT cloud, and AI-as-aService, along with their advantages and conclusions.[7]

The development of energy optimization protocols in wireless sensor networks (WSNs), focusing on the unique characteristics and challenges of WSNs compared to other wireless networks. It highlights the introduction of LEACH as a communication protocol for energy efficiency. The study aims to provide valuable insights for academics, industries, and researchers by analyzing various contributions, limitations, and technologies used in WSN energy optimization protocols[9].

In the latest trends, gesture recognition has been used to advance machines because of its cooperative ability. Gestures are a type of nonverbal communication that allows humans and computers to communicate with one another. Hand gesture recognition is widely used in artificial intelligence to enhance features and user interaction. Here certain libraries of Python (OpenCV, cvzone2) were used that help in capturing, image pre-processing, and detection, along with mapped action pairs to perform specific tasks[10].

PROBLEM IDENTIFICATION

It intends to expand the application created for the aforementioned project in further work to especially address the needs of people with disabilities. The goal is to develop an inclusive and accessible interface that enables people with disabilities, such as those who are physically limited or have speech difficulties, to effectively interact with PowerPoint presentations. This can be done by including extra features and functionalities that allow users to explore the slides and manage the presentation using speech commands, including voice recognition. In order to accommodate people with limited hand mobility, and also want to include assistive technology like eye-tracking or head-tracking gadgets. These improvements will make PowerPoint more accessible and inclusive, enabling people with disabilities to fully participate in presentations and use it as a tool for communication and expression. By emphasizing accessibility and inclusivity hoping to empower and enable people with disabilities to participate fully in social, educational, and professional contexts, thereby advancing equality of opportunity and closing the accessibility gap.

PROPOSED METHODOLOGY

Environment Setup: To create a new project in PyCharm IDE, begin by installing the IDE on your computer and launching it. Once PyCharm is up and running, click on "Create New Project" and provide a suitable project name and location. Next, configure the Python interpreter by selecting the desired version and ensuring that it is properly installed. To install the necessary libraries, open the terminal within PyCharm and execute the commands "pip install cvzone" to install the cvzone library, "pip install opencv-python" for OpenCV, and "pip install numpy" for NumPy. This will ensure that all the required libraries are installed and ready for use in your PyCharm project.

Data Collection and Preprocessing: To gather a collection of hand gesture pictures or videos for testing and training a gesture recognition system, you can search for publicly available datasets or capture your own samples using a camera. Once collected, preprocess the data by resizing the images/videos to a consistent resolution, normalizing the pixel values to a common range, and enhancing the visual quality through techniques such as contrast adjustment or noise reduction. These preprocessing steps will help improve the performance and generalization of the model by ensuring uniformity and optimal quality of the input data.

Hand Detection and Tracking: To identify and follow the user's hand in real-time video frames recorded by the camera, use the HandDetector module from the cvzone library. According to your needs, change the detection confidence threshold and the maximum number of hands.

Gesture Recognition: Process the hand region using computer vision techniques to extract the necessary elements for gesture identification. Apply a gesture recognition method to classify and recognise various hand gestures based on the extracted features, such as a machine learning or deep learning model. Utilize the preprocessed dataset to train the gesture recognition model.

User Interaction and Control: Create a user interface with OpenCV routines to show the presentation slides and live video feed. Implement interactive features based on hand motions the system recognises, such as moving forward or backward via slides. The presenting programme should translate the recognised movements into appropriate actions, such as emulating keyboard shortcuts or mouse events.

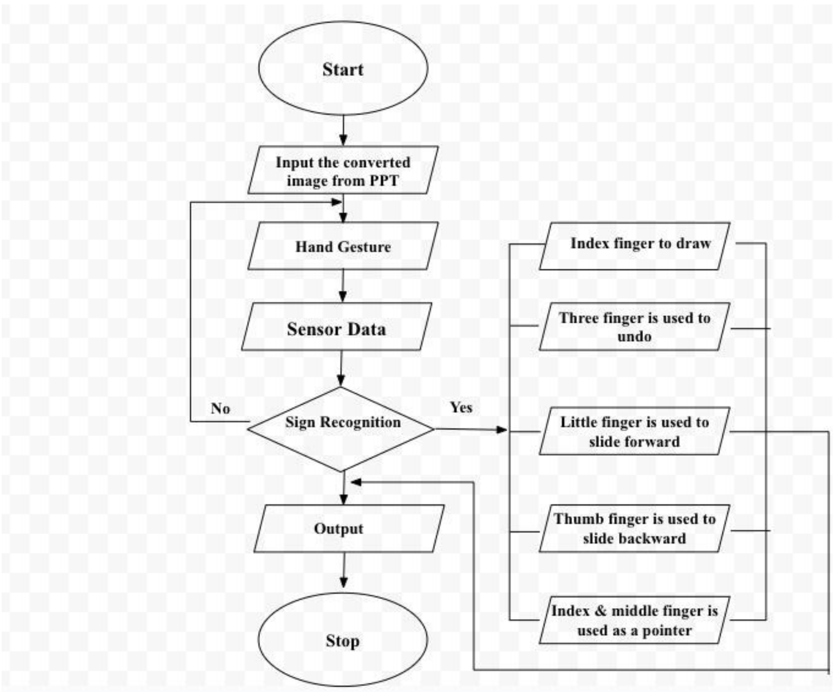
Testing and Evaluation: Test your gesture recognition system with a variety of hand motions and users to determine how well it performs. To evaluate the system's performance in real world circumstances, take into account important performance indicators including accuracy, speed, and resilience. Based on the outcomes of the review, adjust and fine-tune the system to enhance overall performance.

Import Required Library: The code begins by importing the required libraries, which include HandDetector from the cvzone module, cv2 for OpenCV functions, os for file operations, and numpy for mathematical operations.

Setup Parameters and camera: Set the screen's height, width, and gesture threshold parameters. Using cv2, adjust the camera's dimensions.Adjust the width and height of the video after capture.

Initialize the Hand Detector: Create a HandDetector instance from the cvzone module with the detection confidence and maximum hands to be detected that you desire.
Define Variable and Load Presentation Image: Set up several variables to keep track of button presses, annotations, and motions. Indicate the location of the presentation images' storage folder. Utilizing os.listdir, load the presentation image list and sort it by length.
Start the Main Loop for Realtime Gesture Recognition: To constantly process the video frames that the camera has taken, enter an infinite loop. Cap.read() should be used to read the video frame, and cv2 to flip it horizontally.For a mirrored display, flip.
Detect and Track Hand: To find and follow the user's hand in the video frame, utilize the findHands function from the HandDetector instance. Using cv2.line, mark the gesture recognition boundary on the frame by drawing a gesture threshold line.
Process Hand Landmarks and Recognise Gestures: Extract hand landmarks and other details from the hand that was detected. Analyze the index finger's position with relation to the screen's size. Based on the finger states and hand position in relation to the gesture threshold, identify particular hand gestures.
Perform Action Based on Recognised Gesture: If a gesture is recognised, take the appropriate By altering the imgNumber variable with a single finger gesture, the displayed slide can be changed. The annotations on the currently shown slide are removed with a fist closed gesture. The user can annotate the slide with an extended index finger gesture.
Display the Video Frame and Slide Images: Using cv2.imshow, display the current slide's image as well as the current video frame with annotations. For a picture-in-picture effect, resize the video frame and insert it in a tiny window inside the slide image.
Handle Key Press: Utilize cv2.waitKey to look for key press events. If the 'q' key is pushed, the programme will exit the loop and be terminated as shown in below flow chart.

FLOW CHART STEPS



Step 1: Start
Step2: Import the necessary libraries.
Step3: Convert the pdf into the image format.
Step4: Set up parameters and camera dimensions.
Step5: Initialize the hand detector.
Step6: Define variables and load presentation images.
Step7: Start the main loop for the real time gesture recognition.
Step8: Read the video frame.
Step9: Flip the frame horizontally for a mirrored display.
Step10: Detect and track hands in the frame.
Step11: Draw a gesture threshold line on the frame.
Step12: Process hand landmarks and recognize gestures.
Step13: Extract hand landmarks and other information.
Step14: Map the index finger position to the screen dimensions.
Step15: Recognize specific hand gestures based on the fingers' states and hand position.
Step16: Index finger is used to draw. Little fingers are used to slide forward. Thumb is used to slide backwards. Index and middle finger is used as a pointer. 3 fingers are used to undo.
Step17: Perform actions based on recognized gestures.
Step18: Change the displayed slide based on left or right movement.
Step19: Erase annotations made on the current slide with a closed gesture.
Step20: Allow the user to make annotations on the slide with an extended index finger gesture.
Step21: Display the current slide image and video frame with annotations.
Step22: Handle key press events:
Step23: If the 'q' key is pressed, break out of the loop and end the program.
Step24:Stop

WORKING MODEL

That is vital to explain the fundamentals of the Markov Process before going into detail about the Hidden Markov Model. There are two components to a Markov process: 1) a finite number of states since we consider gesture trajectories to be a set of spatially distributed locations. A group of Gaussian geographical regions can be utilized to show the data distribution. 2) Transition probabilities for moving from one state to another, given its current state; and 3) Emission probability is a measure of how likely the system will emit a particular observation, given its current State. One effective method for simulating real-world processes is the Markov process. Assuming a limited set of presumptions are satisfied. These presumptions, known as Markov Properties, include the possibility of moving through a series of transitions from any state to any other state, a fixed set of states, and fixed transition probabilities.

An example can help clarify the Markov Model. Let us imagine that someone is making three successive gestures. The three gestures are: 1) One Finger; 2) Two fingers; 3) Three fingers. On any given period of time only one of the three gestures can be displayed and the next gesture depends only on the present gesture, so this is called a simple Markov Chain. Also assume that on any period of time a person can have one of the following factors; 1) Real Time Hand Gesture; 2) Finger Orientation, moreover these factors depend on the gesture displayed on that particular period of time.

Through this example, we came to know that the gestures are the Hidden States and factors that depend on the gestures are known as Observed Variables. Given that the gesture was made with one finger in the past and two fingers in the present, Markov analysis may estimate the likelihood that the gesture will consist of three fingers in the future.Let's examine the Hidden Markov Models mathematical presumptions and associated graphic.

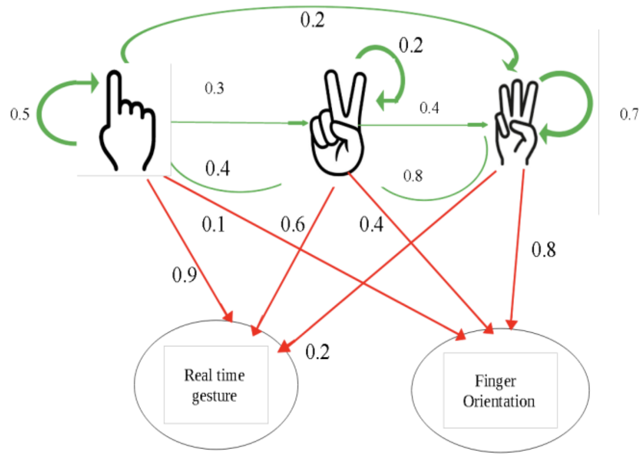


Fig 1.1 Hidden Markov Model

Hidden Markov Model = Hidden Markov Chain + Observed Variable

ALGORITHM FOR HIDDEN MARKOV MODEL

Input = hidden_states, observations, transition_probability, emission_probability,
Output = Probability of Hidden States, Log Probability, Graph of hidden states vs observed states

```
Hidden_states = []
for i in range(3):
Hidden_state = input(f"Enter item{i+1}: ") Hidden_state.append(Hidden_states)

Observations = ['Real Time Hand Gesture', 'Finger Orientation']

starting_probability = np.array([0.5, 0.3, 0.2])

transition_probability_1 = np.array([[0.5, 0.3, 0.2],
[0.4, 0.2, 0.4],
[0.0, 0.3, 0.7]])

emisiion_probability_1 = np.array([[0.9, 0.1],
[0.6, 0.4],
[0.2, 0.8]])

def create_hmm_model(starting_probability, transition_probability, emission_probability):
model = hmm.CategoricalHMM(n_components=3)
model.startprob_ = starting_probability
```

```
model.transmat_ = transition_probability_1
model.emissionprob_ = emisiion_probability_1

def generative_observation_sequence, hidden_states(model, n_samples):
observation_seq, hidden_states = model.sample(n_samples=10) return observation_seq, hidden_states

def log_probability(model, observarion_sequence):
log_probabililty, hidden_states =
model.decode(observation_seq, lengths =
len(observation_seq), algorithm='viterbi')
return log_probability, hidden_states

def plotting_results(hidden_states, observation_sequence):
sns.set_style("whitegrid")
plt.plot(hidden_states, '-o', label="hidden State")
plt.xlabel('Time Step')
plt.ylabel('Most Likely Hidden State')
plt.title("Hand Gestures")
plt.legend()
plt.show()
```

RESULTS

In Gesture-Based Interaction for Enhanced Presentation Control project, a intuitive and interactive system are implemented using specific finger gestures. The index finger is employed to draw orannotate on the screen (fig. 1), allowing users to easily highlight important information. To facilitate quick undo actions, the three fingers gesture is utilized, enabling users to effortlessly revert changes(fig. 2). For Seamless navigation few fingers are designated in this project i.e the little finger to slide forward, enabling smooth transitions between slides(fig. 3). Conversely, the thumb finger gesture allows users to effortlessly slide backward, facilitating efficient backward navigation(fig. 4). Finally, the combination of the index and middle finger acts as a pointer, empowering users to precisely indicate and emphasize specific elements on the screen(fig. 5). Through these various finger gestures, this project offers a seamless and engaging presentation experience.

Fig 1. Index finger to draw.

International Journal for Multidisciplinary Research (IJFMR)

E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

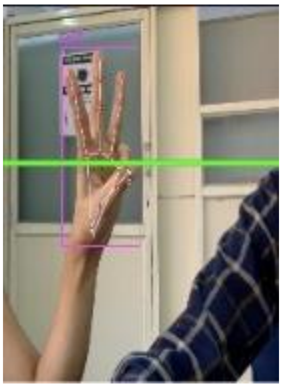


Fig 2. Three fingers used to undo.

Fig 3. Little finger is used to slide forward

Fig 4. Thumb finger is used to slide backward .

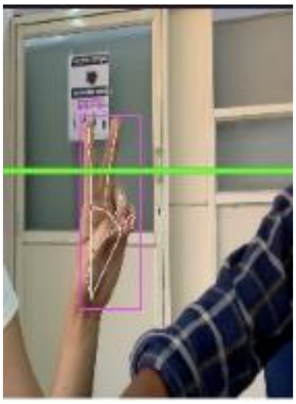


Fig 5. Index and middle finger is used as a pointer

1. Expected Output and Graph

Generated Sequence: [1, 0, 0, 1, 0, 0, 0, 1, 1, 1]
Log probability: -11.671571110371648
Associated Hidden States: [1, 0, 0, 1, 0, 0, 0, 2, 2, 2]

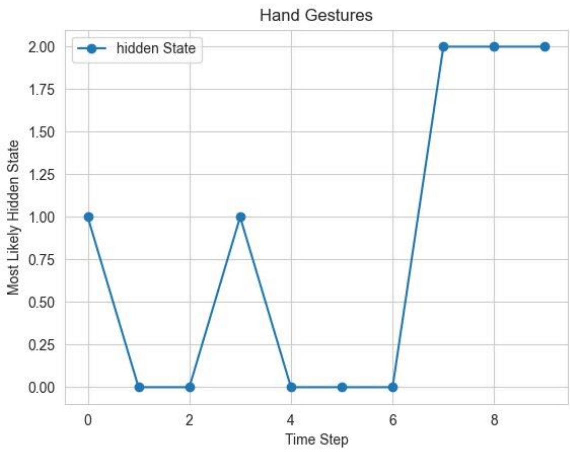


Fig 6. Hidden states vs Observation Sequence Graph

CONCLUSION

A real-time presentation control system employing hand gestures is demonstrated in the given code. The project allows users to browse presentations and annotate them with basic hand movements by utilizing machine vision techniques and the CVZone library.

A webcam is used by the system to record the user's hand movements. The user's hand, together with the finger landmarks, are detected and tracked using the HandDetector class from CVZone. The system takes a number of actions based on the gestures it has identified, including moving to the previous or next slide, adding annotations, and removing annotations.

The programme demonstrates how gesture recognition and computer vision could improve human-computer interaction during presentations. The technology offers a more user-friendly and immersive approach to manage and interact with presentation slides by doing away with the requirement for physical input devices like touchscreens. The code also shows how the CVZone library, which streamlines the process of hand detection and gesture recognition, is flexible and extensible. This makes it possible for programmers to create interactive systems that make use of hand gestures quickly. The real-time presentation control system described in the code paves the way for more dynamic and engaging presentations where presenters may easily regulate the flow of slides and comment on them in real-time using organic hand gestures.

ACKNOWLEDGMENT

To Ramaiah University of Applied Sciences, Bangalore, we would like to convey our profound gratitude for their continuous support.

REFERENCES

1. Devivara prasad G, Mr. Srinivasulu M, “Hand Gesture Presentation by Using Machine Learning,” International Journal Of Innovative Research In Technology. September 2022, IJIRT, Volume 9 Issue 4, ISSN: 2349-6002.

2. Muhammad Idrees,Ashfaq Ahmad, Muhammad Arif Butt, and Hafiz Muhammad Danish, "Controlling Power Point Using Hand Gestures In Python," Webology (ISSN: 1735-188X) Volume 18, Number 6, 2021.
3. Sakthivel, U., Jyothi, A.P., Susila, N. and Sheela, T., 2021. "Conspectus of k-Means Clustering Algorithm. In Applied Learning Algorithms for Intelligent" IoT (pp. 193-213). Auerbach Publications.
4. G.Reethika, P.Anuhya, M.Bhargavi, "Slide Presentation By Hand Gesture Recognition Using Machine Learning," International Research Journal of Engineering and Technology, Volume: 10 Issue: 01, Jan 2023.
5. Jyothi. A P, A. Shankar, A. J. R. Narayan, K. Bhavya, S. S. M. S. Reddy and Y. A, "AI Methodologies in Upcoming Modern Warfare Systems," 2022 IEEE International Conference on Current Development in Engineering and Technology (CCET), Bhopal, India, 2022, pp. 1-7, doi: 10.1109/CCET56606.2022.10080014.
6. Abha Ingle, Karan Jadhav, Ratish Kumar Jha, Rakshit Karkera, Bhavesh Panchal, "Hand Gesture Recognition System Using Deep Learning," International Journal of Advances in Electronics and Computer Science, ISSN(p): 2394-2835 Volume-6, Issue-8, Aug.-2019.
7. Abhishek B, Kanya Krishi, Meghana M, Mohammed Daaniyaal, Anupama H S, "Hand gesture recognition using machine learning algorithms" Computer Science and Information Technologies Vol. 1, No. 3, November 2020, pp. 116~120 ISSN: 2722-3221, DOI: 10.11591/csit.v1i3.p116-120.
8. Jyothi, A.P., Usha, S. and Archana, H.R., 2022. "Adoption of Machine/Deep Learning in Cloud With a Case Study on Discernment of Cervical Cancer. *Advanced Analytics and Deep Learning Models"*, pp.65-109.
9. Jyothi, A. P., & Sakthivel, U. (2015). "Trends and technologies used for mitigating energy efficiency issues in wireless sensor network". International Journal of Computer Applications, 111(3), 32-40.
10. Indriani, Moh.Harris, Ali Suryaperdana Agoes, "Applying Hand Gesture Recognition for User Guide Application Using MediaPipe," Advances in Engineering Research, volume 207.
11. Chetana D. Patil, Amrita Sonare, Aliasgar Husain, Aniket Jha, Ajay Phirke, "International Journal of Creative Reattach Thoughts, Volume 10, Issue 11 November 2022, ISSN: 2320-2882.