# Computer Vision Quiz through Hand Gestures Movements

**N Kumaran[1], Sarvepalli Srivathsav[2], Sri Goutam SG[3]**

[1]ASSISTANT PROFESSOR, Dept. of Computer Science, SCSVMV University, Kanchipuram.

[2]Final Year Student, Dept. Of Computer Science, SCSVMV University, Kanchipuram.

[3]Final Year Student, Dept. Of Computer Science, SCSVMV University, Kanchipuram

***Abstract***— Nowadays, computer-based multiple-choice exams are used universally in the recruitment, examination, and other processes. Making the setup for a computer-based test is very expensive because it requires a lot of equipment and software. We are working on a project that will employ computer vision to reduce the amount of hardware, like a keyboard and mouse, needed for computer-based exams. The suggested solution would use computer vision to display the question on screen after saving it to a csv file first. Second, we will search for hand movements in the air that can be used to respond to the presented questions using a webcam and computer vision. The automatic grading system in it will then display the grade. This approach made it easy to respond to questions and even made the evaluation procedure straightforward. To verify that no malpractice occurred during the exam, this technology ensures more security by counting the number of people who are in front of the webcam, if more than one person the exam automatically ends. This technique had a 95.7% success rate in accurately detecting hands. Importantly, because of very little technology is needed, the exam won't be terminated in the middle due to a hardware failure or malfunction.

**Keywords -** Computer Vision, CSV, Hand Movements, Webcam

## 1.INTRODUCTION

Computer use is now widespread and has a big impact on how we live our lives. Computer applications require human-computer interaction.

We frequently use hand gestures, which are simply the movement, position, or posture of the hands, to communicate nonverbally. To categorize hand motions in videos and images for various purposes, several research are being conducted. The actual movement of the human hand is what creates gestures. Hand gesture recognition is a natural and intelligent method of Human-Computer Interaction (HCI).

The term "computer-based testing," or "CBT," refers to the administration of tests using computers as opposed to the traditional pen-and-paper approach. The internet or a computer-aided facility can be used to administer such a test online. Computer technology is used in computer-based testing (CBT), which means that applicants use computers to respond to questions displayed on monitors. The test-taker enters the response using a keyboard or mouse. The "client computer" the test-taker is using is referred to as such.

Providers of online examination content should focus on creating useful assessment questions and giving students exam feedback. Students or trainees might use a computer to complete exercises or tests with the help of online examination systems. The issue with the examiner workload in the manual system has been resolved. This will verify the right response, save the examiner's time, and ensure that the exam is administered efficiently.

In contrast to the existing approach, reviewing the answer sheets after completing the test wastes the examiner's time. A web-based online assessment system's major goal is to assess the student using a fully automated method thoroughly and efficiently that not only saves a ton of time but also provides quick and accurate results.

## 2. LITERATURE SURVEY:

The detection of hand gestures can be done in Python in a variety of ways, including using an artificial neural network, background removal, HMM, transfer learning CNN, etc.

A Hand Gesture Recognition and Appliance Control Using Transfer Learning had been implemented by Srinivasa Rao K, Gundam Bhavani, and Pagolu Sriharika. In this implementation, Python is used to recognize and identify human hand motions. This process flow consists of background removal, hand ROI segmentation, contour detection, and finger recognition using a transfer learning CNN trained model. Even though these studies only employed short datasets and are only useful in constrained circumstances, the results reveal accuracy rates of nearly 100%. **[1]**

Using OpenCV and Python, Surya Narayan Sharma and Dr. A Rengarajan developed Hand Gesture Recognition in 2021. Using computer vision with Python and Numpy, the aim of this technique of implementation is to develop a comprehensive system that can recognize, locate, and interpret hand motions. Only plain backgrounds can be used for this detection, but accuracy needs to be increased. Implementation and accuracy issues will result from even a little problem. **[2]**

A vision-based system for hand gesture recognition was proposed by Shaminder Singh and Anuj Kumar Gupta. Artificial Neural Networks, HMM, and Fuzzy C-means clustering were the main classifiers employed in this implementation. ANNs have frequently been used to categorize and separate dynamic motions while simultaneously capturing the correct hand form. This strategy, nevertheless, only works in a few specific situations. Although these researches only employed limited datasets, the results reveal accuracy rates of almost 100% in several circumstances. A uniform model cannot be created for a dataset of conventional hand gestures. **[8]**

Multiple-choice tests administered using a computer are almost always employed in today's hiring, testing, and other procedures. These software programmers were made with a variety of programming languages. In the same way that Er. Pranay Meshram, Rajat Tarale, Vaishnavi Upare, and Shruti Wakodikar developed the idea for an online MCQ test PHP and SQL are used in the web application development of the online examination system. The administration can see, delete, and update student information with efficiency. However, the main drawbacks of this type of application are that setup necessitates more expensive hardware systems, and if there are any interruptions, it is impossible to finish exams on time. Additionally, as there is no automatic evaluation mechanism in place for them, it takes longer to review the test and publish the results. **[4]**

Face detection and identification using opencv and python had been implemented by Dhawle, Tejashree, Ur-vashi Ukey, and Rakshandha Choudante. The major objective of this study is to develop or construct a system that will use the computer's camera to recognize and identify a person's or an individual's face using the Python programming language and the OpenCV tool known as the Open Face. Dlib and face recognition are the system's two most important libraries. Face recognition technology has several advantages, including quicker processing, identity automation, privacy invasion, massive data storage, the best results, increased security, real-time face recognition of employees and students at corporate offices, smartphone unlocking, and many more. There are some drawbacks to this system, including its high costs or funding requirements, the requirement for high-quality cameras, poor image quality that may limit its efficacy, the significance of image size because it becomes difficult to recognize faces in small images, face angles that may decrease the accuracy of face recognition, and the requirement for enormous amounts of storage**. [5 ]**

## 3.PROPOSED METHOD AND METHODOLOGY:

Our objective is to create a software system that uses computer vision and a webcam to answer questions on multiple-choice assessments that are computer-based utilizing hand gestures. The candidate's score will also be computed by this system and displayed on the screen. There won't be any hardware interruptions, unlike with older systems, due to the system's minimal hardware configuration, which also makes implementation and maintenance expenses affordable.

We employ hand gestures in this project to provide answers to queries, and the cvzone module in Python is used to track the hand movements. This computer vision package facilitates the execution of AI and image

processing operations. It primarily makes use of the OpenCV and Mediapipe libraries. Cvzone was employed in our project. HandTracking modules for utilizing the media pipe algorithm. It offers a hand detector feature that can find hands and palms. This library makes use of Media Pipe, a Google cross-platform library that uses machine learning methods to extract 21 3D hand landmarks from a single frame.

**Using Media Pipe for hand tracking takes two steps:**

• **Palm detection:** Media Pipe uses the entire image as input and outputs a cropped image of the hand.

• **Identification of hand landmarks** - MediaPipe identifies the 21 hand landmarks on the hand's cropped image.

MediaPipe Hands is a high-quality hand and finger tracking system. It extrapolates 21 3D hand landmarks from a single frame using machine learning (ML). Unlike other cutting-edge systems, which mostly rely on potent desktop settings for inference, our method provides real-time performance on a cell phone and even scales to numerous hands. We believe that opening up access to this hand perception functionality to a wider research and development community will spark the development of creative use cases and new lines of inquiry.

A model for detecting palms that makes use of the complete image and generates an orientated hand bounding box is part of the MediaPipe Hands ML pipeline. A hand landmark model that produces extremely precise 3D hand key points while operating on the palm detector-cropped image region. This method is comparable to the one used in our MediaPipe Face Mesh solution, which combines a face detector with a face landmark model.

Giving the hand landmark model a properly cropped hand image greatly reduces the requirement for data augmentation (such as rotations, translations, and scaling) and allows the network to concentrate its resources on accurate coordinate prediction. Palm detection is only performed to delocalize the hand when the landmark model is unable to do so. The crops in our pipeline can also be produced using the hand landmarks found in the previous frame. A dedicated hand renderer subgraph is utilized for rendering, and a hand landmark tracking subgraph from the hand landmark module is used in the pipeline's implementation as a MediaPipe graph. The hand landmark tracking subgraph uses both a hand landmark subgraph from the same module and a palm detection subgraph from the palm detection module internally.

### 3.1 PALM DETECTION MODEL:

In a manner similar to the face detection model in MediaPipe Face Mesh, we created a single-shot detector model intended for mobile real-time uses to detect initial hand locations. The issue of detecting hands is incredibly challenging because both our lite model and complete model must be able to recognize occluded and self-occluded hands while operating over a wide scale span (20x) relative to the image frame. It is somewhat challenging to identify hands correctly from their visual features alone, in contrast to faces, which exhibit high contrast patterns,such as in the region of the mouth and eyes. Accurate hand localization is instead made possible by adding additional context, such as aspects of the arm, torso, or person.

Our approach employs many techniques to address the aforementioned problems. First, we train a palm detector rather than a hand detector because it is far easier to estimate the bounding boxes of stiff objects like fists and palms than it is to detect hands with articulated fingers. The non-maximum suppression technique also performs well in two-hand self-occlusion scenarios like handshakes since palms are smaller objects. The number of anchors can be reduced by a factor of 3–5 by modelling palms with square bounding boxes, or anchors in machine learning terminology. In order to provide greater scene context awareness, even for small objects, an encoder-decoder feature extractor is used. Finally, to support many anchors coming from the high scale variance, we reduce the focus loss during training. With the methods, we can detect palms with an average precision of **95.7%**. With no decoder and a standard cross entropy loss, the baseline is only **86.22%.**

### 3.2 HAND LANDMARK MODEL:

After identifying the palm over the full image, our next step is to achieve exact key point localization of 21 3D hand-knuckle coordinates inside the identified hand regions using regression, or direct coordinate prediction. Self-occlusions and partially visible hands have no effect on the model's ability to obtain a trustworthy internal

hand posture representation. To gather ground truth information, we manually added 21 3D coordinates to about 30K actual photographs, as seen below (we take Z-value from image depth map, if it exists per corresponding coordinate). In addition, to better cover the range of potential hand positions and provide extra supervision on the nature of hand geometry, we render a high-quality synthetic hand model over a variety of backgrounds and map it to the related 3D coordinates.

- **The figure below depicts the 21 hand points that Media-Pipe recognizes:**

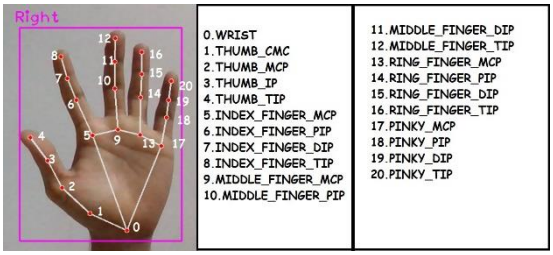


Fig-1: 21 handmarks in mediapipe

### 3.3 Dlib:

Dlib was used in this project to count the faces in the webcam. The C++ programming language was used to create the general-purpose, cross-platform software library known as Dlib. Its design is significantly influenced by concepts from component-based software engineering and design by contracts. It is therefore first and primarily a collection of separate software parts. The software is open-source and distributed under the Boost Software License. Dlib has developed to now contain a large number of tools since development first started in 2002. It now has software elements for dealing with networking, threads, graphical user interfaces, data structures, linear algebra, machine learning, image processing, data mining, XML and text parsing, numerical optimization, Bayesian networks, and many more tasks. Although Dlib is primarily a C++ library, you can use some of its capabilities in Python programs.

## IV. IMPLEMENTATION:

These steps describe how this system operates:

★ Start Quiz:

Before Starting the Exam, the candidate must enter his name login this name will be visible permanently during the complete exam.

★ Live Video Capturing:

This is the Initial step taken by the system Immediately after the candidate logged into the exam. Video capturing is done with the help of an initialized webcam using OpenCV library.

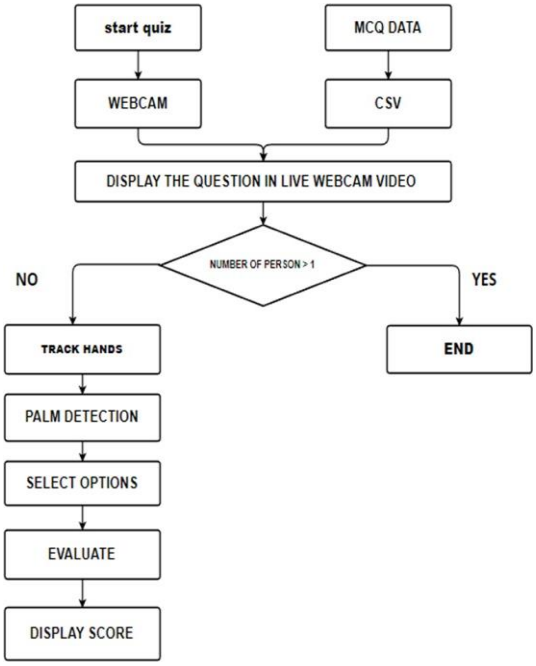


Fig.- 2: **Workflow**

★ MCQ Details:

The questions, options and correct answer for the question are stored in a csv file. Later this data is useful for displaying during the exam and for evaluating the student's answer. And to calculate the score of the candidate.

★ Presenting the Questions:

In this step the system extracts the details from the csv file where the questions and options are already stored in it the csv file is extracted using csv library in python. Then the questions and options are displayed on the screen on live video using OpenCV and CVZONE libraries.

★ Counting number of Faces:

In this stage, the system uses the C++-written dlib package to identify the candidate's face. This library makes it easier to count the faces in front of the webcam. It is useful to know how many people are present while the exam is taking place. Once the number of people has been counted, if more than one person is found, the

system will debar the candidate from the exam and the process will continue until the exam is over in order to ensure that no malpractice occurred.

★ Tracking and Detecting Hand:

Next the system tracks the hand and detects the palm of the candidate through a webcam so that it can understand the gesture shown by the candidate. This detection and Tracking are done with the help of a hand tracking class in CVZONE library. This CVZONE library uses Media Pipe Algorithm which is created by google for tracking hands and detecting the palm. This Media pipe will detect 21 hand landmarks in the palm which make the system get better understanding about the gestures shown. Answering the questions is based on clicking the option button. The system identifies the clicked button using the distance between the index finger and Middle finger if the distance is less than the button in front of the hand is considered as clicked. For the exit button it considers the distance between ring and thumb fingers.

The distance between the fingers is calculated by the CVZONE library as it identifies the hand landmarks according to that the tip of index finger and middle finger is given with landmark numbers as 8 and 12 respectively by the media pipe algorithm. Then the distance between these points is calculated if the distance is less than 60 then it considers the option is clicked. Similarly, the distance between the tip of the ring and thumb finger is calculated using their landmarks mentioned in media pipe.

★ Displaying the Score:

The answer selected by the user is stored in an array as the index of the option number and it is verified with the options stored in the csv file. If both are the same then it is considered and taken for grading. Grading is done with the formula:

**Score (%) = (no. of correct answer / total no. of questions) * 100**

- Then this score is displayed on the screen using OpenCV and CVZONE library.

## 5. RESULT AND ANALYSIS:

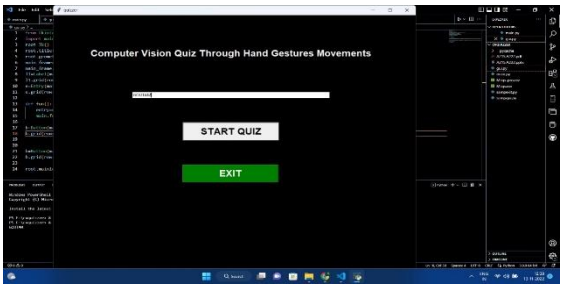


Fig. 3. Login Page

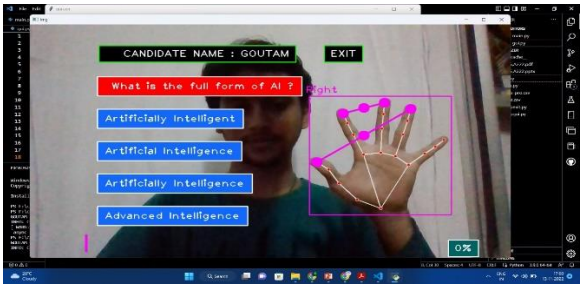


Fig. 4. Answering and Hand Gesture Detection

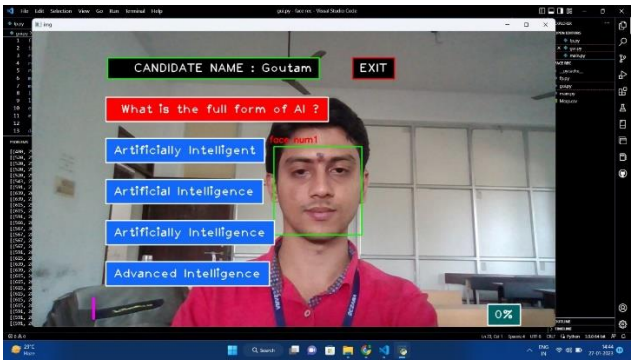


Fig. 5. Face Detection

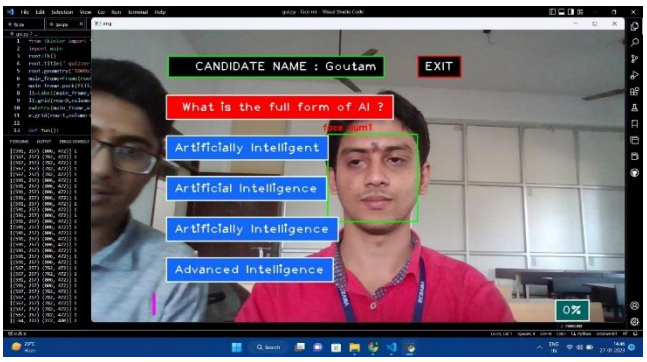


Fig. 6. Detection of multiple face

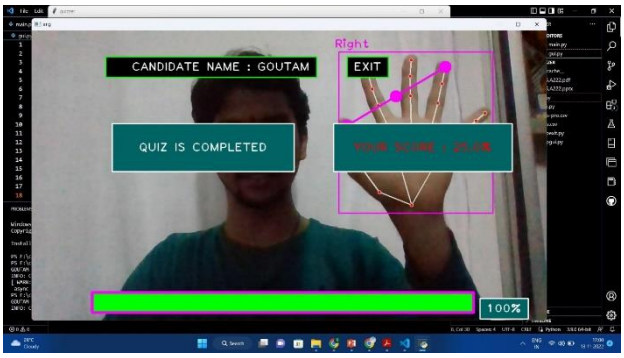


Fig. 7. Displaying Score

- **TABLE -1: Comparison of Accuracy rate of used method vs other methods for hand tracking**

| METHOD | ACCURACY |
|---|---|
| Edge Detection | 78.1% |
| Background Subtraction | 96.6% |
| BP Neural network with Levenberg-Marquardt (L-M) algorithm | 88.4% |
| Fuzzy ARTMAP Classifier | 92.19% |
| Haar like and HOG with SVM classifier | 83% |
| RBF Technique with ANN classifier | 93% |
| Shape Based Features with ANN classifier | 85% |
| **Media Pipe Algorithm (ours)** | **95.7%** |

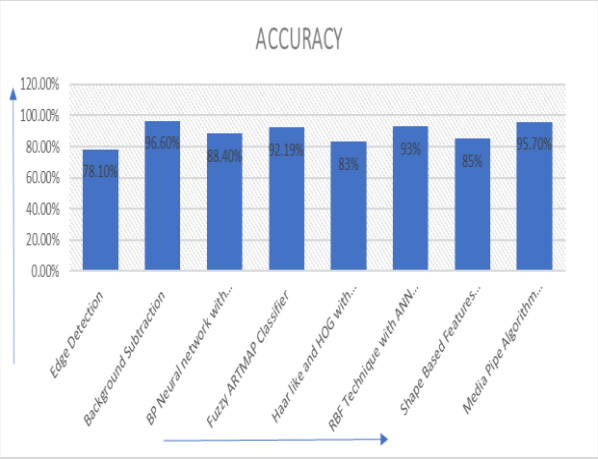


Fig. 8. **Comparison of Accuracy rate of used method vs other methods for hand tracking**

**x - axis : Methods y - axis: Accuracy**

## 6. CONCLUSION:

Python and computer vision are used to build hand motion and facial recognition methods with computer-based test applications. The method of answering questions and the evaluation process were made simpler by this application. With the help of this program, the exam will always be taken by a single person, preventing exam fraud. This application is low-cost because it only needs a little hardware setup. The system must have better illumination and a good webcam in order to work properly. Any institution that needs to hold competitive exams will find this method to be very helpful. This technique had a 95.7% success rate in accurately detecting hands. Additionally, this system includes an automatic evaluation mechanism that assesses the applicants' responses considering the CSV file's correct answer. Anyone may use the system easily because of its user-friendliness.

## ACKNOWLEDGMENT

The authors would like to thank Sri Chandrasekharendra Saraswathi Viswa Mahavidhyalaya (SCSVMV) Deemed to be University for supporting this work.

## REFERENCES

1. Hand Gesture Recognition and Appliance Control Using Transfer Learning Srinivasa Rao K, Gundam Bhavani, Pagolu Sriharika Vol. 11, Issue 7, (Series-III) July 2021.
2. Hand Gesture Recognition using OpenCV and Python Surya Narayan Sharma, Dr.A Rengarajan, Bengaluru, Karnataka, India (IJTSRD) Volume 5 Issue 2, January-February 2021.
3. Friuli's, George F., Maria Papatsimouli, Lazaros Lazaridis, and Ioannis A.Skordas. "An Online Dynamic Examination System (ODES) based on open source software tools." *Software Impacts* 7 (2021): 100046.
4. Web Application for Online MCQ Test Er. Pranay Meshram,Rajat Tarale, Vaishnavi upare, Shruti Wakodikar , Aastha Bankar, Sarika Nimaje Vol-6 Issue-3 2020.
5. Dhawle, Tejashree, Urvashi Ukey, and Rakshandha Choudante. "Face detection and recognition using opencv and python." *Int. Res. J. Eng. Technol* 7, no. 10 (2020).
6. Ismael, Khansaa Dheyaa, and Stanciu Irina. "Face recognition using Viola-Jones depending on Python." *Indonesian Journal of Electrical Engineering and Computer Science* 20, no. 3 (2020): 1513-1521.
7. Oudah, Munir, Ali Al-Naji, and Javaan Chahl. "Hand gesture recognition based on computer vision: a review of techniques." *journal of Imaging* 6, no. 8 (2020): 73.

8. Computer Vision based Hand Gesture Recognition: A Survey Shaminder Singh, Anuj Kumar Gupta Vol.-7, Issue-5, May 2019 (IJCSE)
9. Yasen, Mais, and Shaidah Jusoh. "A systematic review on hand gesture recognition techniques, challenges and applications." *PeerJ Computer Science* 5 (2019): e218.
10. D Deshmukh, A., Mahender G Nakrani, D. L Bhuyar, and U. B Shinde. "Face recognition using OpenCv based on IoT for smart door." In *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India*. 2019.
11. Khan, Maliha, Sudeshna Chakraborty, Rani Astya, and Shaveta Khepra. "Face detection and recognition using OpenCV." In *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 116-119. IEEE, 2019.
12. Zhang, Zhenyuan, Zengshan Tian, and Mu Zhou. "Latern: Dynamic continuous hand gesture recognition using FMCW radar sensor." *IEEE Sensors Journal* 18, no. 8 (2018): 3278-3289.
13. Web Application MCQ B. pavithra, P.pavithra, R.Thiagarajan Volume 7 IssueNo.4 2017.(IJESC)
14. Xu, Xiaomeng, Sierra Kauer, and Samantha Tupy. "Multiple-choice questions: Tips for optimizing assessment in-seat and online." *Scholarship of Teaching and Learning in Psychology* 2, no. 2 (2016): 147.
15. Computer-Vision Based Hand Gesture Recognition and Its Application in Iphone Hsi-Chieh Lee, Che-Yu Shih & Tzu-Miao Lin (SIST,volume 21), 2013.