TOUCHLESS ECOSYSTEM USING HAND GESTURE RECOGNITION SYSTEM

Soham Patil[1], Sakshi Kodarlikar[2], Shruti Marathe[3], Deepti Vijay Chandran[4]

[1-3]Students, Dept. of Computer Science, Smt. Indira Gandhi College of Engineering, Navi Mumbai, Maharashtra, India.
[4]Professor, Dept. of Computer Science, Smt. Indira Gandhi College of Engineering, Navi Mumbai, Maharashtra, India.

---***---

Abstract: Touchless interfaces are indeed more critical in the period of COVID-19. Reducing or eliminating touchpoints is essential for safely resuming businesses. The pandemic has shaped the value proposition of touchless technologies, accelerating their development and deployment. An urge for contactless interaction and hygiene concerns are the top drivers of demand for touchless technology. As the pandemic has swept over humankind, gesture control and touchless user interfaces have become hot content. Both give the capability to interact with devices without physically touching them. The global COVID-19 pandemic has created a more need for alternatives to common diurnal practices to reduce the spread of the virus. During the coronavirus pandemic, it's not astonishing that people are reluctant to use touchscreens in public places. In this scenario, it's logical to utilize touch-less technologies. Because touchless technologies can prevent the spread of viruses. People tend to use touchless systems as much as possible in order to reduce the spread of viruses. Airlines have switched to self-service check-in, which is a touch-based system. It's not safe to use that service currently. There's never been a need for touchless tech like there's a need for it now. With the help of the Tensor Flow model, we will be detecting hand points and identifying the different gestures.

Keywords: Hand gestures, Mediapipe, tensorflow.js, hand pose model, touchless ecosystem

I. INTRODUCTION

The world is presently suffering from the crisis of the COVID – 19 virus so, touching any random object can beget major harm in anyone's life. To minimize the spread of any virus, nowadays people tend to use touchless systems as much as possible. We know that almost all the airlines have switched to self-service check-in, which is generally based on the touch-based system. In the current scenario, it's not safe to use that service as there are lots of people who come in contact with the machine. In the last few years, gesture-controlled interactive surfaces have become widespread. So, our project aims to make an ecosystem of gesture-based usable UI which can be used without touching the machine itself i.e. trying to make a touch-free model with the help of Machine Learning to use hand gestures.

Touch-based systems aren't safe anymore, as they can contribute to the spread of disease. This Touchless ecosystem will consist of the most used services by people like ATM, Airport Checkin, Restaurant ordering, paying bills in a mall or store, etc. With the help of hand gestures, the user can interact with the respective services like in a Check-In kiosk the user can select the preferred seats, enter boarding pass number, print the boarding pass, etc. Whereas in ATMs wherein there is a necessity for a higher level of security, the ecosystem provides multiple levels of security. Firstly the user has to scan a QR code which is provided by the bank, then an OTP Verification mail needs to be cleared, and lastly, the user's pin to successfully log in to his/her account and make transactions. Likewise, multiple systems can be added to the ecosystem as per necessity. This touch-less hand gesture-based model is very accurate and capable of performing all the steps with the help of the Tensor flow hand pose model. Thus, using deep learning models and the very user-friendly UI we have successfully achieved our objectives and implemented gestures to navigate and use the Check-in system.

II. MODULES/FRAMEWORKS USED

A) HANDPOSE MODEL

The handpose model is a tensorflow.js model which is based on the google mediapipe model which is used to detect hands in real-time. It draws a bounding box around the hand in the input image or video stream and after processing it further returns twenty-one 3D landmarks locating features within each hand. These landmarks define different locations on each finger joint and the palm. The key points then can be used to train different gestures like pinching.

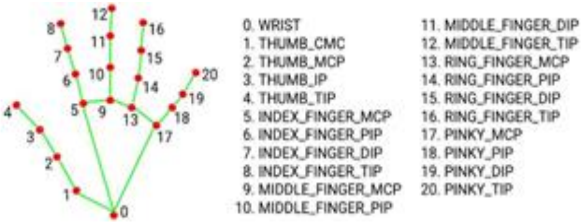


Fig. 1 Different 3D landmarks plotted on a hand

B) DJANGO

Django is a Python web framework to create MVT pages in a web app. We are using it to create the UI of the ecosystem as multiple apps can be created in Django which can be used to design multiple different systems in our ecosystem. It links the frontend part of the project to the

SQL server and provides a user-friendly admin page to manage and view the data in the database.

C) TENSORFLOW.JS

TensorFlow.js is a framework used for training and deploying machine learning models in the web frameworks or node,js. It has multiple tools, libraries, and community resources to easily build and deploy ML-powered applications. We are using the handpose model which is in tensorflow.js which can be easily implemented and used in web frameworks like Django etc.

III. METHODOLOGY

A) HOW HANDPOSE MODEL WORKS

The TensorFlow handpose model is used to detect 21 3D landmarks of a hand from a live video stream. The handpose model is based on google media pipe hands which are embedded in tensorflow.js which can be used in web applications. For a given input, the model predicts whether input contains a hand. If it contains, the model returns coordinates for the bounding box around the hand, also 21 key points within the hand, by outlining the location of each finger joint as well as the palm. First, we train a palm detector rather than a hand detector, since estimating bounding boxes of rigid objects like palms and fists is significantly simpler than detecting hands with articulated fingers. As palms are smaller objects, the non-maximum suppression algorithm works well also for two-hand self-occlusion cases, like handshakes. Moreover, palms are often modeled using square bounding boxes (anchors in ML terminology) ignoring other aspect ratios, and thus reducing the number of anchors by a factor of 3-5. Second, an encoder-decoder feature extractor is used for bigger scene context-awareness even for small objects. Lastly, we minimize the focal loss during training to support an outsized number of anchors resulting from the high scale variance. The centroid of the hand is then taken which is then used to simulate the cursor of the mouse. As the centroid moves along the screen the x and y distance is taken and the cursor is placed on the screen which is visible to the user and can be used to navigate around in the UI. The gesture is trained for the click function using the Handsfree() module where the hand variations are captured and stored to work where the Thumb_Tip and Index_Finger_Tip are pinched twice which can be then implemented as a new gesture in the web application.
The click function is implemented with the cursor which then is used together to simulate a mouse where the click is performed when the index finger and thumb are pinched twice within a duration of 80ms.

B) HAND DETECTION

The hand gesture detection system is focused on user-data interaction, where the user interacts by using different gestures to perform tasks that would otherwise need touch input to interact. The hand detection system tracks the real-time position of the hand in the 3D orientation of the fingers of each marker and uses gestures to interact. Here the handpose model is used which plots a bounding box around the hand. The bounding box image is then further processed and is used to plot different 3D landmarks on the hand. These key points plotted on the hand then can be used to train different gestures.

IV. SECURITY FEATURES

The systems like ATMs or Malls wherein transactions are to be made, multiple levels of security are needed to be in place to secure the transactions. The following Security measures can be implemented in the ecosystem. One or multiple measures can be implemented in one system to make it more secure.

A) QR CODE

In this type of security, a unique QR code is provided to the user which contains the user's details. When it's scanned the details are accessed by the system to log in to the user.
The following are the merits of using QR codes.

- Omnidirectional and Fast Scanning: QR codes can be read much faster and within 360 degrees can be scanned from any angle i.e. no need to place the scanner as per the code symbol.
- Small Size: QR code takes less space. A QR Code can hold the same volume of information contained in a 1-D barcode in only one-tenth the space.
- Huge Data Storage Capacity: QR code has high data storage capacity.
- Many Types of Data: The QR Code can handle numerals, alphanumeric characters, Japanese, Chinese or Korean letters, and binary data. Error correction:
- Wide Range of Uses: There are lots of potential uses of QR codes. They can be implemented to extend the user experience in-store, restaurants, websites, etc.

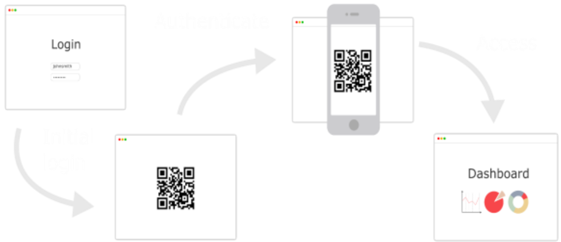


Fig. 2 QR Code Authentication

B) OTP VERIFICATION

An OTP is transferred to the user on his mail or phone number given by the user. When the user enters the correct OTP he/ she is logged into the system. The OTP system creator passes the user's secret pass- expression, in confluence with a seed entered from the server as a part of the challenge, through multiple repetitions of a secure hash function to produce a one-time password. After each successful authentication, the quantity of secure hash function repetitions is reduced by one. Hence, a unique sequence of passwords is generated. The server verifies the one-time password entered from the creator by calculating the secure hash function on time and comparing the result with the preliminarily accepted one-time password.

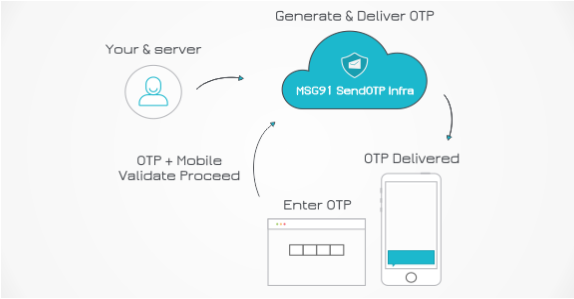


Fig. 3 OTP verification working

C) FACE DETECTION

This type of security requires a high-quality camera to detect the user's face and to recognize it. If the face matches, the system logs in to the user. The face recognition process involves comparing the picture with a set of stored faces to spot the person within the input image. Factors like shape, size, pose, occlusion, and illumination influence the method of face recognition. Basically, it has two different applications: basic and advanced. Major face recognition searches for unique factors such as the width of the nose, wideness of the eyes, the depth and angle of the jaw, the height of cheekbones, also the separation between the eyes.

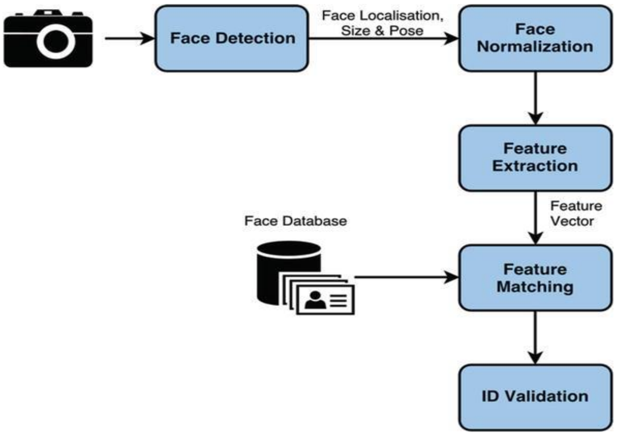


Fig. 4 Face Detection working

Using this information makes a unique numerical code. By utilizing these numerical codes, the system matches that image with another image and differentiates how the images are comparable to each other. The image provenance for face recognition includes pre-existing images from different databases and also from video camera signals. A face recognition system involves face detection, feature extraction, and face recognition.

D) PALM VEIN AUTHENTICATION

Palm Vein Authentication System compares the patterns of veins in the palm which is different for each individual and then the individual is authenticated only when the pattern of his palm matches with a pattern that is stored in the database of the system. For this user has to place his hand above the scanner then a near-infrared light maps the lines of the veins which is then stored by the system. It is the most secure way of scanning as the authentication data exists inside the palm which would be difficult to forge.

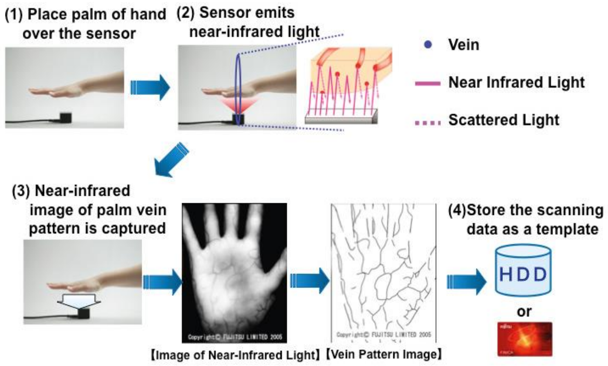

(1) Place palm of hand over the sensor
(2) Sensor emits near-infrared light
Vein
Near Infrared Light
Scattered Light
(3) Near-infrared image of palm vein pattern is captured
(4)Store the scanning data as a template
HDD
or
[Image of Near-Infrared Light]
[Vein Pattern Image]

Fig. 5 Palm vein authentication system

E) IRIS SCANNING

This requires a high-quality camera along with an iris scanner to obtain very high-resolution clear images of the iris which are then matched with the user's data in the system. If the iris matches the system logs in to the user. The main steps of iris recognition are as follows acquisition, segmentation, normalization, feature extraction, and feature matching. Currently, methods use machine learning models like DCNN, CNN. The obtained result is then compared with the result in the database and the iris are recognized.

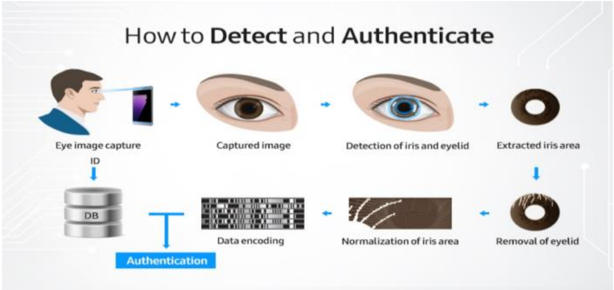


Fig. 6 Iris scanning

V. RESULT

The handpose model we are using gives highly accurate results even in many different lighting conditions which solves the major issue in hand recognition where the results produced are inaccurate if there is frequent change in lighting conditions or shift in colors. The gestures are also tracked accurately and produce the required results.

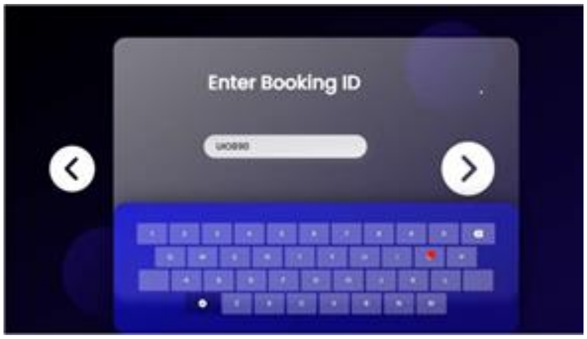
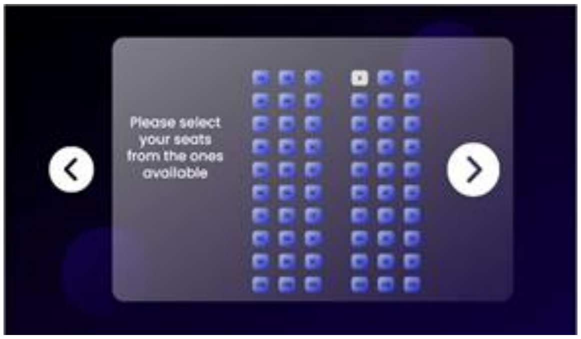


Fig. 7 Typing using the hand gestures



Please select your seats from the ones available

Fig. 8 Selecting seats using the hand gestures

Multiple security systems can be implemented to make the payments and transactions more secure and easy to perform. We have implemented a QR code that can be scanned to verify the user details. Further, an OTP verification is implemented where the user gets a code on his provided mobile number or email which can be used to login into the account.

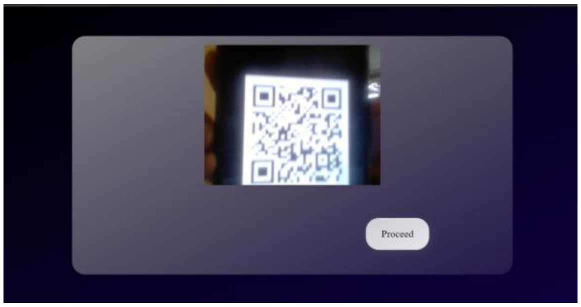


Fig.9 Scanning the QR Code

VI. CONCLUSION

Touch-based systems aren't safe anymore, as they can contribute to the spread of disease. This Touchless ecosystem ensures that all of its gesture-based systems are safer to use as compared to the systems. With the help of hand gestures, the user can easily navigate the UI and perform the same operations as in traditional systems but touch-free. This touch-less hand gesture-based model is very accurate and capable of performing all the steps with the help of the Tensor flow hand pose model along with robust security systems that can be implemented to make the transactions etc as secure as possible. Thus, using deep learning models, multiple security systems, and the very user-friendly UI we have successfully achieved our objectives and implemented an ecosystem that can perform the tasks of the most used systems by users touch-free.

REFERENCES

[1] Dharmik Bhanushali, Dipanshu Someshwar, Vismay Chaudhari, Ms. Bhargavi Dalal (2020). Hand Gestures Recognition using DeepLearning.

[2] B., Abhishek & Krishi, Kanya & M., Meghana & Daaniyaal, Mohammed & S., Anupama. (2020). Hand gesture recognition using machine learning algorithms. Computer Science and Information Technologies. 1. 116-120. 10.11591/csit.v1i3.p116-120.

[3] Fan Zhang Valentin Bazarevsky Andrey Vakunov Andrei Tkachenka George Sung Chuo-Ling Chang Matthias Grundmann.(2020). MediaPipe Hands: On-device Real- time Hand Tracking

[4] Nhu-Tai Do , Soo-Hyung Kim*, Hyung- Jeong Yang and Guee-Sang Lee(2020).Robust Hand Shape Features for Dynamic Hand Gesture Recognition Using Multi-Level Feature LSTM

[5] M. H. Eldefrawy, K. Alghathbar and M. K. Khan, "OTP-Based Two-Factor Authentication Using Mobile Phones," 2011 Eighth International Conference on Information Technology: New Generations, 2011, pp. 327-331, doi: 10.1109/ITNG.2011.64.

[6] Shally, & Aujla, Gagangeet & Aujla, Singh. (2014). A REVIEW OF ONE-TIME PASSWORD MOBILE VERIFICATION. International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR). 4. 113-118.

[7] Alapetite, Alexandre. (2010). Dynamic 2D-barcodes for multi-device Web session migration including mobile phones. Personal and Ubiquitous Computing. 14. 45-52. 10.1007/s00779-009-0228-5.

[8] Nithya, Alice & Lakshmi, C.. (2015). Iris recognition techniques: A Literature Survey. International Journal of Applied Engineering Research. 10. 32525-32546.

[9] Jain, Tarshi & Kumar, Rajendra. (2019). A STUDY OF VEIN RECOGNITION SYSTEM. Acta Informatica Malaysia. 3. 13-15. 10.26480/aim.01.2019.13.15.

[10] Guimarães, Nuno. (1991). Building generic user interface tools. ACM SIGPLAN Notices. 26. 89-96. 10.1145/117954.117961.

[11] Kumar, Ashu & Kaur, Amandeep & Kumar, Munish. (2019). Face Detection Techniques: A Review. Artificial Intelligence Review. 52. 10.1007/s10462-018-9650-2.