AIRPORT CHECK-IN SYSTEM USING HAND GESTURES

Sakshi Kodarlikar[1] , Shruti Marathe[2] , Soham Patil[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 even more critical in the era of COVID-19. We're wary of touch like never before, an aversion likely to persist until a viable vaccine becomes available – and potentially beyond. Reducing or eliminating touch points is essential for safely reopening businesses. The pandemic has crystallized the value proposition of touchless technologies, accelerating their development and deployment. A desire for contactless sensing and hygiene concerns are the top drivers of demand for touchless technology. As the pandemic has swept over the world, gesture control and touchless user interfaces have become a hot topic. Both provide the ability to interact with devices without physically touching them. The global COVID-19 pandemic has created a greater need for alternatives to common daily practices to reduce the spread of the virus. During the coronavirus pandemic, it's not surprising that people are reluctant to use touchscreens in public places. In this scenario, it is good to use touch less technologies. Because touch-based technologies can cause the spread of virus. People are more tend to use touch less systems as much as possible in order to reduce the spread of virus. Airlines have switched to self-service check-in, which is touch-based system. It's not safe to use that services anymore. There's never been a need for touch less tech like there's a need for it now. With the help of Tensor Flow model, we will be detecting hand points and identifying the different gestures.

Keywords: Hand gestures, Mediapipe, tensorflow.js, handpose model

I. INTRODUCTION

As we all know, world is currently suffering from the crisis of COVID – 19 virus. Touching any random object can cause major harm in anyone's life. To minimize the spread of any virus, nowadays people are more tend to use touch less systems as much as possible. We know that almost all the airlines have switched to self-service check-in, which basically based on touch-based system. In 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 the airport check-in machine usable with the help of gestures i.e., try to build touch less model with the help of hand gestures.

Touch-based systems aren't safe anymore, as it can contribute in the spread of disease. This Touch less check-in kiosk ensures that it is safe to use as compared to traditional kiosk. With the help of hand gestures, user successfully collect boarding pass. This touch less hand gesture-based model is very accurate and capable of performing all the steps with the help of 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. HAND DETECTION

The finger tracking system is focused on user-data interaction, where the user interacts with virtual data, by handling through the fingers the volumetric of a 3D object that we want to represent. This system was born based on the human-computer interaction problem. The objective is to allow the communication between them and the use of gestures and hand movements to be more intuitive, Finger tracking systems have been created. These systems track in real time the position in 3D and 2D of the orientation of the fingers of each marker and use the intuitive hand movements and gestures to interact.

The ability to perceive the shape and motion of hands can be a vital component in improving the user experience across a variety of technological domains and platforms. For example, it can form the basis for sign language understanding and hand gesture control, and can also enable the overlay of digital content and information on top of the physical world in augmented reality. While coming naturally to people, robust real-time hand perception is a decidedly challenging computer vision task, as hands often occlude themselves or each other (e.g. finger/palm occlusions and hand shakes) and lack high contrast patterns.

III. TECHNOLOGIES

A) Handpose Model

The handpose package detects hands in an input image or video stream, and returns twenty-one 3-dimensional landmarks locating features within each hand. Such landmarks include the locations of each finger joint and the palm. In August 2019, we released the model through MediaPipe - you can find more information about the model architecture in our blog post accompanying the release. Refer to our model card for details on how handpose performs across different datasets. This package is also available through MediaPipe.

Handpose is a relatively lightweight package consisting of ~12MB weights, making it suitable for real-time inference.

TensorFlow is the premier open-source deep learning framework developed and maintained by Google. Although using TensorFlow directly can be challenging, the modern tf.keras API beings the simplicity and ease of use of Keras to the TensorFlow project. Using tf.keras allows you to design, fit, evaluate, and use deep learning models to make predictions in just a few lines of code. It makes common deep learning tasks, such as classification and regression predictive modelling, accessible to average developers looking to get things done.

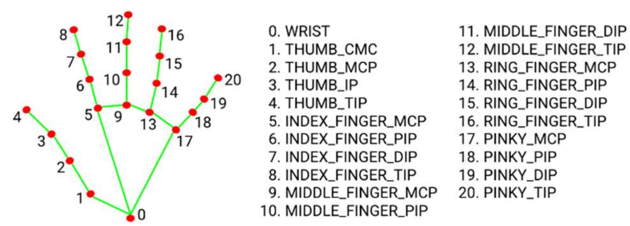

Fig 1 – Different 3D landmarks plotted on hand

B) Electron.js

Electron is a framework that enables you to create desktop applications with JavaScript, HTML, and CSS. These applications can then be packaged to run directly on macOS, Windows, or Linux, or distributed via the Mac App Store or the Microsoft Store.

Typically, you create a desktop application for an operating system (OS) using each operating system's specific native application frameworks. Electron makes it possible to write your application once using technologies that you already know

From a development perspective, an Electron application is essentially a Node.js application. This means that the starting point of your Electron application will be a package.json file like in any other Node.js application.

Electron consists of three main pillars:

- **Chromium** for displaying web content.
- **Node.js** for working with the local file system and the operating system.
- **Custom APIs** for working with often-needed OS native functions.

Developing an application with Electron is like building a Node.js app with a web interface or building web pages with seamless Node.js integration.

Electron APIs are assigned based on the process type, meaning that some modules can be used from either the Main or Renderer process, and some from both. Electron's API documentation indicates which process each module can be used from.

C) Tensorflow.js

TensorFlow.js is a JavaScript Library for training and deploying machine learning models in the browser and in Node.js.
TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications. TensorFlow offers multiple levels of abstraction so you can choose the right one for your needs. Build and train models by using the high-level Keras API, which makes getting started with TensorFlow and machine learning easy.

IV. 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 mediapipe hands which isembedded in tensordlow.js which can be used in web applications.
For the gestures to work:
Given an input, the model predicts whether it contains a hand. If so, the model returns coordinates for the bounding box around the hand, as well as 21 key points within the hand, outlining the location of each finger joint and the palm.

First, we train a palm detector instead of a hand detector, since estimating bounding boxes of rigid objects like palms and fists is significantly simpler than detecting hands with articulated fingers. In addition, as palms are smaller objects, the non-maximum suppression algorithm works well even for two-hand self-occlusion cases, like handshakes. Moreover, palms can be modelled using square bounding boxes (anchors in ML terminology) ignoring other aspect ratios, and therefore 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 a large amount of anchors resulting from the high scale variance.

The centroid of 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 is 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 is pinched twice within a duration of 80ms.

V. RESULT

The use of Deep Learning over Machine Learning is done because proposed project demands high need of accuracy in Classification Algorithms it is better to make use of Deep Learning. Just like Data

Classification is the most important aspect of Data Science, similarly Image Classification plays a very big role in Computer Vision. Image Classification first deals with image preprocessing, then it executes image segmentation and then it deals with key feature extraction that helps for finding uniqueness in different trained images and lastly executes matching identification.

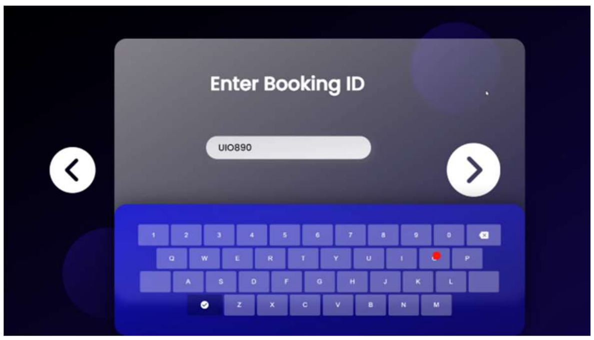

FIG 2 – Typing using the hand gestures

The deep learning models are found to be better accurate than the OpenCV models because the main limitation was change of color was happening very rapidly by the change in the different lighting condition, which may cause error or even failures. For example, due to insufficient light condition, the existence of hand area is not detected but the non-skin regions are mistaken for the hand area because of same color. This limitation is mostly overcome in the deep learning models as these models use 3D land marks to detect hand instead of the color based differentiation method used in the traditional old methods.

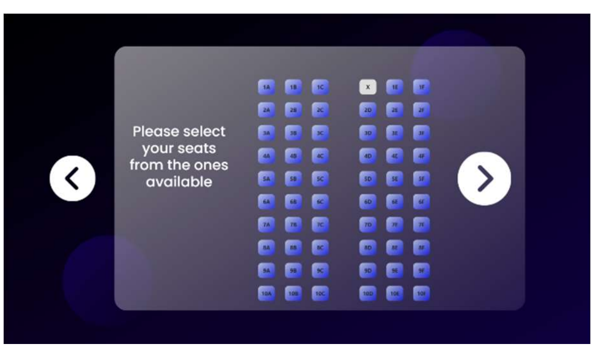

FIG 3 – Selecting seats using the hand gestures

VI. CONCLUSION

Touch-based systems aren't safe anymore, as it can contribute in the spread of disease. This Touch less check-in kiosk ensures that it is safe to use as compared to traditional kiosk. With the help of hand gestures, user successfully collect boarding pass. This touch less hand gesture-based model is very accurate and capable of performing all the steps with the help of 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.

VII. REFERENCES

[1] Dharmik Bhanushali, Dipanshu Someshwar, Vismay Chaudhari, Ms. Bhargavi Dalal (2020). Hand Gestures Recognition using DeepLearning. https://www.irjet.net/archives/V7/i3/IRJET-V7I31032

[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