Department of Computer Science

BSCCS Final Year Project Report
2022-2023

22CS035

Optimizing Unmanned Hotels Using Gesture Recognition

(Volume ___ of ___)

Student Name	:	FUNG King Shun
Student No.	:	56659723
Programme Code	:	BSCEGU2
Supervisor	:	Prof WONG, Hau San Raymond
1st Reader	:	Prof LIAO, Jing
2nd Reader	:	Prof LI, Minming

For Official Use Only

Student Final Year Project Declaration

I have read the project guidelines and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I hereby declare that the work I submitted for my final year project, entitled:

Optimizing Unmanned Hotels Using Gesture Recognition

does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Project Guidelines.

Student Name:	FUNG King Shun	Signature:	
Student ID:	56659723	Date:	July 16, 2023

Acknowledgements

I would like to express my deepest gratitude to my Supervisor Prof WONG, Hau San for their invaluable guidance and support during my FYP.

I am deeply indebted to Prof WONG, Hau San for their invaluable contribution towards the successful completion of my final year project. His guidance, support, and expertise have played a pivotal role in shaping this endeavor into a rewarding experience.

Abstract

The project aims to address the challenges faced by traditional hotel operations during public health crises, such as the recent epidemic. And to solve by leveraging non-contact gesture recognition technology. Also, with the Hong Kong government's "Hong Kong Smart City Blueprint" and its goal of creating a "Smart City 2.0". The project incorporates various components, including an online booking website for unmanned hotels, a simulated check-in process using public machines, and gesture authentication for unlocking room doors. To achieve this, I utilized PHP and Python programming languages. PHP was employed for website development and backend functions like database management, while Python was primarily used for training deep neural network model to detect gesture authentication. Overall, the project aims to optimize unmanned hotels through the implementation of gesture recognition technology and thereby improve traditional hotel business operations.

Table of Contents

1. Introduction

This part mainly explains the motivation for choosing this project and the structure of the theme. The following theme includes background, current development problem, solution, project goals, and project scope.

1.1. Background

Under the covid-19 epidemic, people's lives have begun to change to varying degrees. Among them, the worst consequence of the epidemic is human-to-human contact. Another reason for the accelerated development of the virus around the world is the cross-infection between countries.

Although the Hong Kong government has implemented a regulation which is compulsory quarantine in hotels for people returning to Hong Kong from high-risk areas [1], there is still a certain chance of being infected in the hotel. Therefore, some countries have begun to develop unmanned hotels to solve this problem.

Apart from the covid-19 issue, Hong Kong is now planning a Smart City blueprint. One of the plans is to develop unmanned facilities, and unmanned hotels must be the direction of future development. So, this project also tries to improve the existing hotel business model.

1.2. Current Development Problem

Hong Kong as an international society, and every day different tourists or workers arrive. According to the tourism statistics of the Hong Kong Tourism Board, in the past 2021-2022, about 230,000 people visited Hong Kong (Figure 1 [2]).

Although unmanned hotels can reduce the chance of infection, occupants still need to use the public machines in the lobby to complete the check-in / check-out procedure before entering the room / leaving the room. These public machines are operated in traditional ways, such as touch screen or button control. Therefore, another concern is that occupants may become infected by touching these public machines.

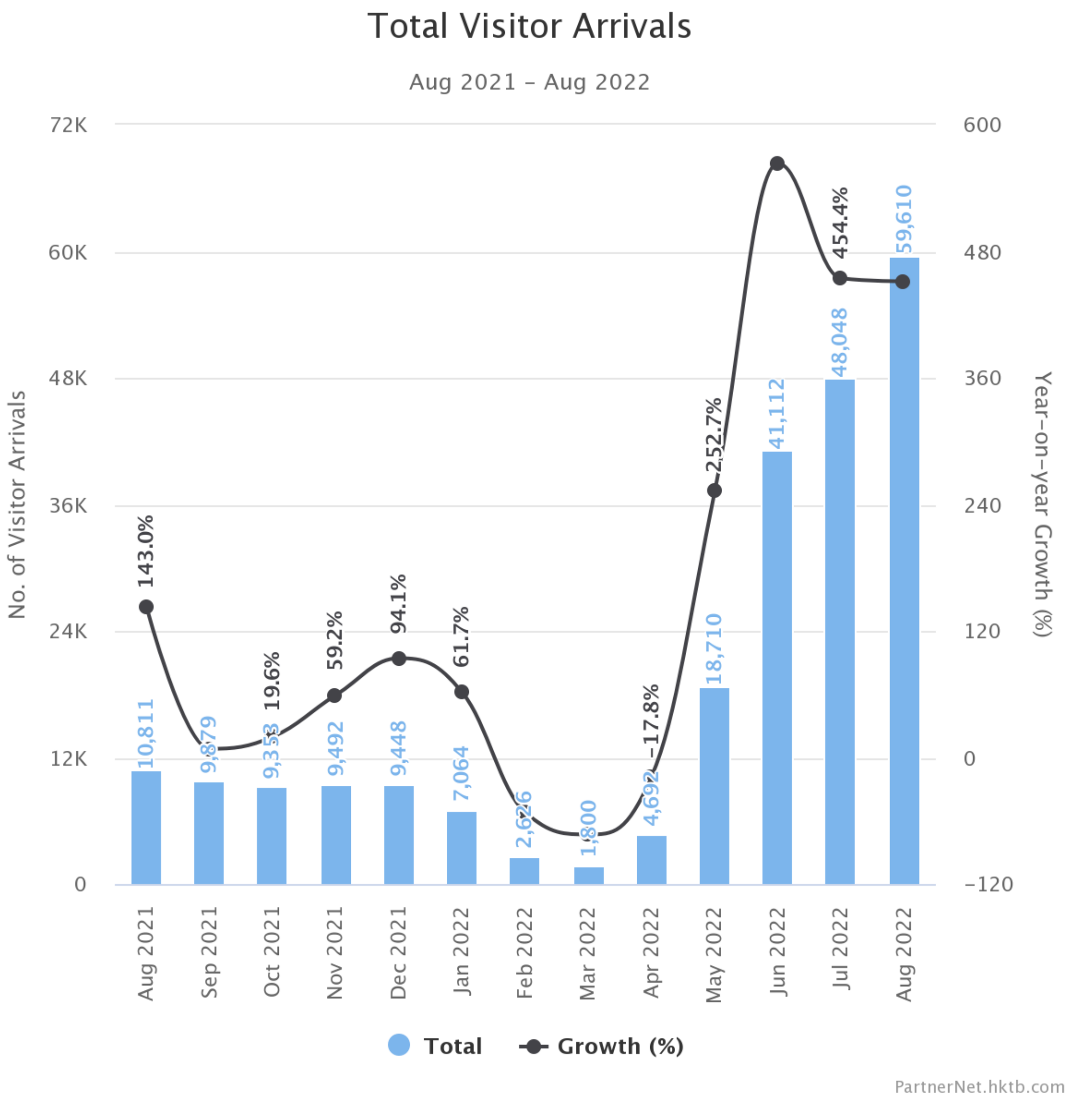


Figure 1 Total Visitor Arrivals in Hong Kong from Aug 2021 to Aug 2022

Furthermore, a big project called “*Hong Kong Smart City Blueprint”* is underway. The goal of the Hong Kong government is to build Hong Kong into a world-famous smart Hong Kong with a thriving economy and a high-quality life through the implementation of this "Hong Kong Smart City Blueprint" project [24].

In this blueprint, the term “smart airport” was introduced [25]. The content talks about the application of automation technology in airport operations. And unmanned facilities are one such development discovery. In fact, an unmanned convenience store called “travelwell” started operating at Hong Kong International Airport on April 19, 2023 [26]. This proves that it is feasible and potential to build an unmanned hotel at Hong Kong International Airport in combination with the smart city project. Therefore, it is necessary to improve the current operating procedures of unmanned hotels to provide a better experience.

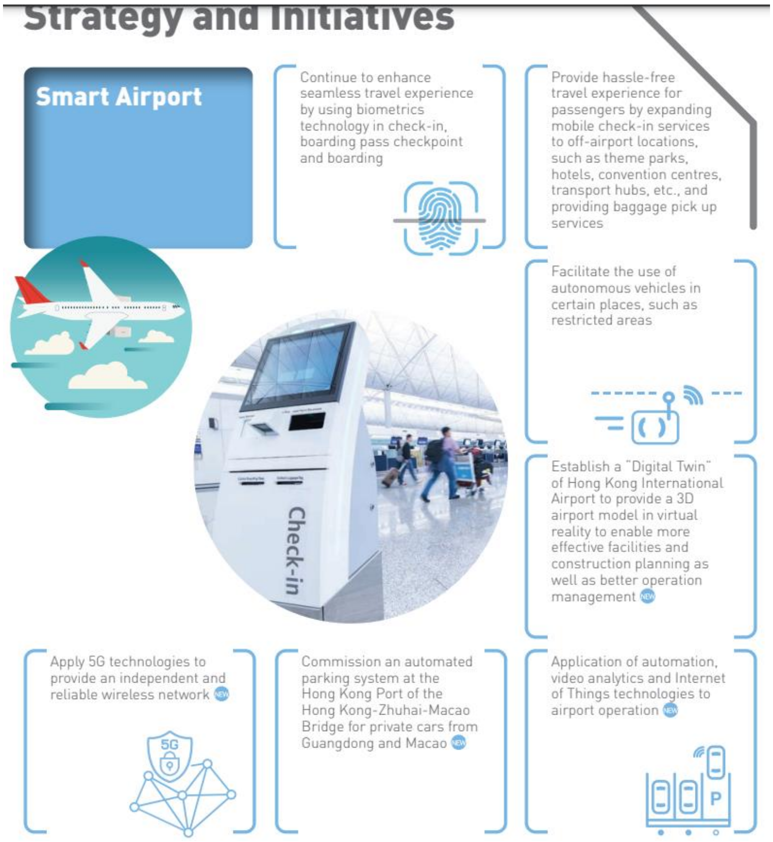


Figure 2 Smart Airport Explanation of HK Smart City

1.3. Solution

The human-machine interface is developing in a non-contact direction, among which gesture recognition is the current key development technology. Especially, gesture recognition does not require touching the device, gestures are used to create appropriate commands for a running program [3]. So, gesture recognition is a suitable solution to support the management of unmanned hotels.

1.4. Project Goals

This project aims to optimize the operation process of the current unmanned hotel using gesture recognition. Implementing gesture innovative technology, not only replaces the current traditional method of controlling public machines but also refers to gesture authentication to solve the long-term problem of needing to touch the door handle or use the keycard to open the door. Using detect your hand or finger method in a specific way in front of the webcam to tell the machine what actions need to be done and what actions should be taken for identity verification.

To sum up, the ultimate goal is to reduce the risk of infection when using unmanned hotels after implementing the gesture recognition technology application. And improve the unmanned hotel to cooperate with the development project of Hong Kong's smart city.

1.5. Project Scope

First, two kinds of registrants will use the services in unmanned hotels, namely walk-in and online booking. The system will develop corresponding operating procedures for different kinds of registrants. A high-level project organization chart architecture is shown (Figure 3).

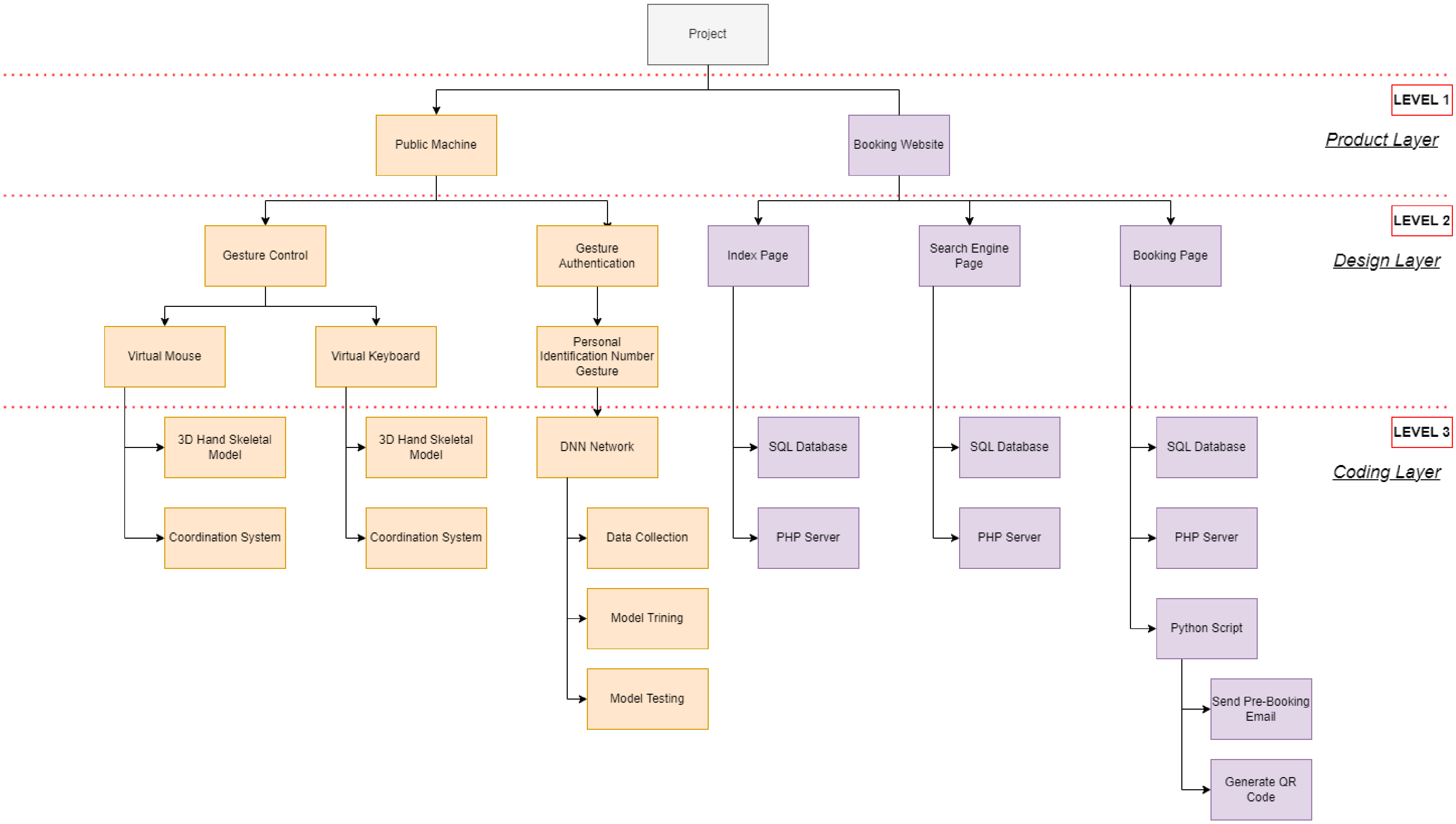


Figure 3 Project Organization Chart Architecture

The application will be:

- Virtual Mouse

It provides an infrastructure between the user and the computer using only a camera and the finger. It allows users to interface with machines without the use of a real mouse [4]. Also, users can use different gesture recognition action control mouse functionalities, such as mouse pointer moving, mouse clicking, etc.

- Virtual Keyboard

It provides a virtual user interface for the keyboard. The user can enter text or numbers using the finger. And the correct result will be displayed on the computer.

- Gesture Authentication

Registrants will need to customize their own gesture authentication when checking in with the public device. After finishing the check-in process, occupants can use their customized gesture authentication to unlock the hotel room door.

- Online Booking Website

A website to provide a complete online booking system that is connected to the database. The website can solve problems related to reservations. Also, the website will display room information and the availability of the rooms, so that users can book rooms in advance.

2. Literature Review

In this section, I will first discuss why traditional hotels should start changing hotel management systems in the 21st century and develop them into smart hotels or unmanned hotels. Review new business models, like adapting new operations models and an emerging trend of providing a touch-free environment of an unmanned hotel during COVID-19, trying to rebuild traveler's confidence [5] and in line with the development of the future smart city. Also, I will specifically explain the problem of different kinds of hand gesture technology applications and how to reform the gesture better.

2.1. Public Health Crisis

The hotel business is a traditional service industry because it needs to hire a large number of employees to assist the operation. The advantage of this business model is it can contribute to the country's economic situation, and industries with a global economic contribution (direct, indirect, and induced) exceeding 7.6 trillion USD in 2016 [6].

However, in the past few years, there have been many crises and disasters that have harmed the hospitality industry. Among them, the traditional way of operating a hotel involves many human contacts, including many customers and employees. At the same time, it is not only a financial crisis, but it also reflected public health crisis is a very serious problem in hotels managed traditionally [7]. And the COVID-19 pandemic has also exposed the vulnerability of the traditional hotel service industry [8].

2.2. Smart City Infrastructure

Smart city infrastructure aims to use technology to improve the quality of urban life, provide sustainable and efficient cities, create a more livable urban environment and improve the well-being of residents. The HK Government announced the "Hong Kong Smart City Blueprint" in December 2017, which proposed proposals under six categories: "Smart Mobility", "Smart Living", "Smart Environment", "Smart Citizens", "Smart Government" and "Smart Economy" total 76 measures [24].

This shows the smart city is one of the important criteria for judging a city's development level and explains why advances in technology are so important.

2.3. Hand Gestures Based on Wired Glove Approach

The data glove or wired glove is an input device that is a glove worn on the hand contains various electronic sensors and monitors the hand's movements. Data gloves capture hand motions using various sensors, such as optical fiber sensors, resistance sensors, and inertial measurement units (IMUs) [14]. A modular design of the data glove using different sensors is shown (Figure 4 [14]). This type of hand gestures method can collect all the data with hand position, and fingers position without loss of data or catching any noise data.

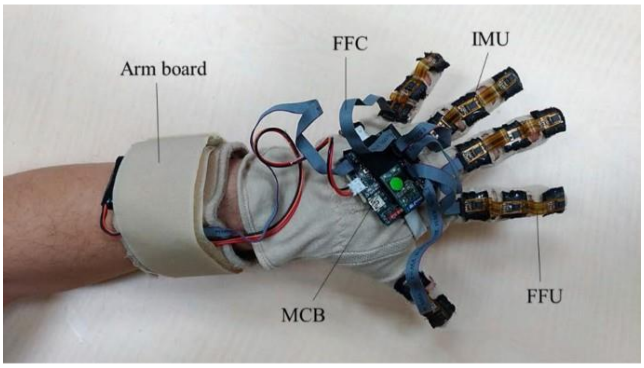


Figure 4 Modular design of the data glove

Nevertheless, the main disadvantage and weakness is the issue of wearing. And this is an Achilles heel. Because its comfort is very unsatisfactory if the user wears heavy gloves for a long time. Further, the wired glove is difficult to adjust the hand sizes for different users to use. Certainly, there are other issues with fragile gloves, high implementation costs, the need for wired connections., etc.

2.4. Deep Learning Hand Gesture Recognition Using Image Detection

Deep learning recognition like Convolutional Neural Net (CNN), Long Short-Term Memory (LSTM) and Recurrent Neural Networks (RNN) is a very powerful tool doing on image gesture. Moreover, the most common deep learning method used in hand gestures is Convolutional Neural Net (CNN). A convolutional neural network (CNN) is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex [16]. Just put each frame of preprocessed image data into the classifier for classification. This deep learning method can avoid skin color segmentation, palm detection, and skin area cropping (Figure 5[16]). It can reduce the computer processing time to run it faster.

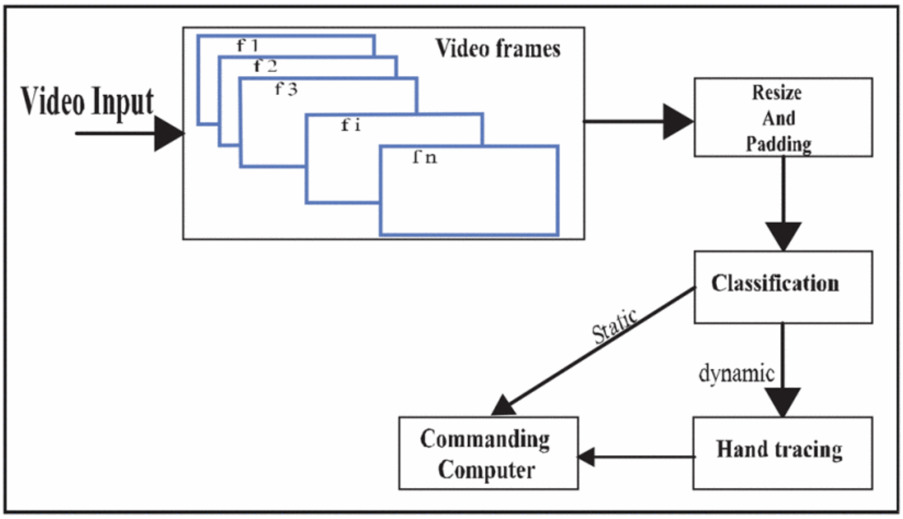


Figure 5 Workflow of CNN-based gesture recognition

But using CNN as the hand gesture method may exist some drawbacks. In fact, creating a CNN model that can recognize objects at the same level as humans is difficult. Even though CNN mimics the human visual cortex, it still has flaws. CNN often struggles to classify images if they contain some degree of tilt or rotation. When we are collecting the dataset, it is not possible to capture all images at different angles, positions, lighting, and shape.

3. Methodology & Resources

In this section, different possible approaches will be analyzed, and the best solution will be found and used based on their feasibility, limitation, etc. The content contains a Vision-Based Hand Gesture, Website Server Architectures and Version Control & Virtual Environment.

3.1. Vision-Based Hand Gesture

Actually, researchers had proposed and implemented several hand gesture representations, the two major categories of hand gesture representation are 3D model-based methods and appearance-based methods as depicted (Figure 6 [9]). As you can see, the figure showed two sub-topics namely Appearance Based Approach and 3D Model Based Approach which both are the most popular hand gesture recognition methods in the world. Next, I analyzed their method what technology was applied and how it was realized in gesture recognition. Finally, according to their pros and cons, the most suitable method will be selected for this project.

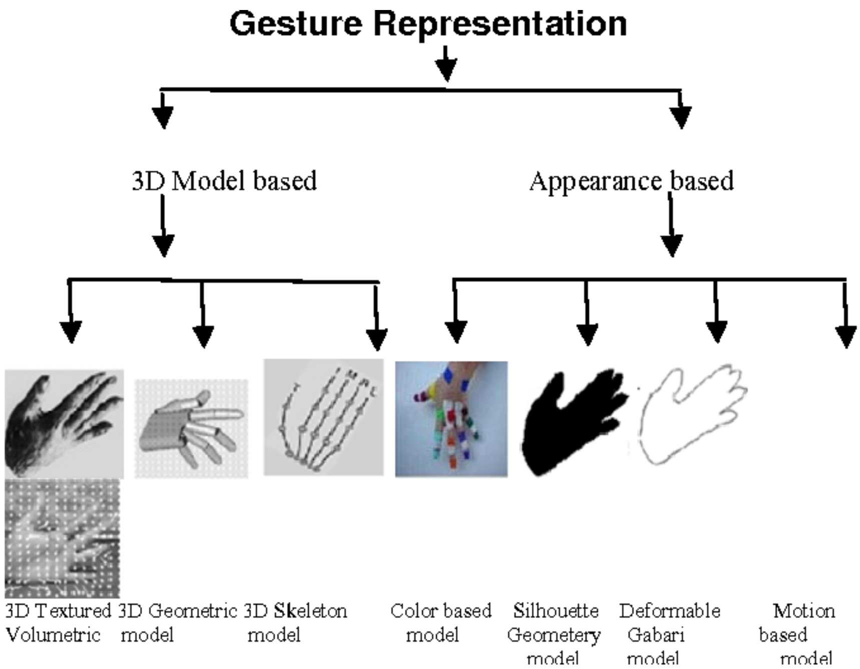


Figure 6 Vision-based hand gesture representations

Although appearance-based and 3D model-based has different branches, such as color based, silhouette geometry based, deformable gabarit based, motion based, and 3D skeleton based, they can be classified into 2D static or 3D static. Generally, there are three steps to handling gesture recognition, namely image preprocessing, tracking, and recognition (Figure 7 [10]).

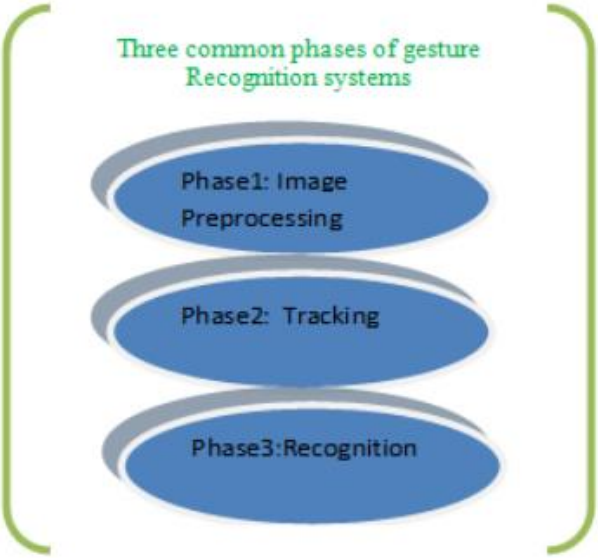


Figure 7 Three common stages of gesture recognition system

3.1.1. Image Preprocessing

The purpose of image preprocessing is to improve the quality of the image so that it can be better analyzed. When capturing a real-life RBG image, it exists a lot of relevant data, such as pixels, brightness, contrast, etc. So, segmentation is crucial, isolating the main object from the image background. Here are some of the technologies of segmentation in visual features:

- Segmentation

  Skin color segmentation is one of the methods that have been used extensively for hand segmentation. The major decision is in providing a skin mask model to normalized RGB, HSV, YCrCb, YUV, etc., and then select which color space will be employed. The aim of this algorithm is trying to find the human skin color in the image spit the background and hand (Figure 8 [11]).

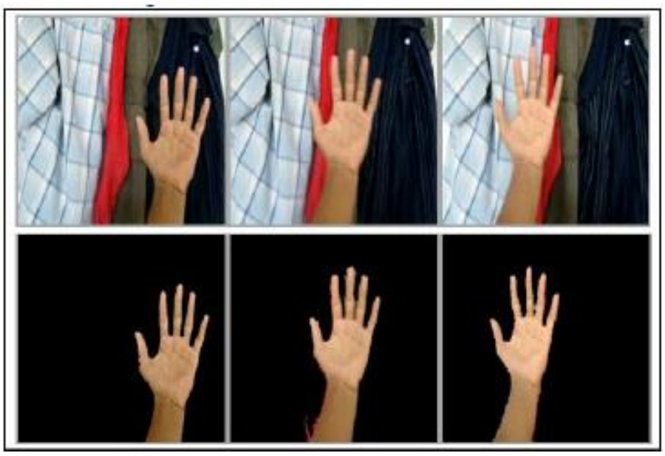


Figure 8 Color-skin segmentation of hand detection

- Remove Image Noise

After extracting the hand object by color skin segmentation, the image needs to be converted into a binary image (black and white image) of the hand on the background. But it may have some noise. Therefore, the ideal situation is to remove a bit of noise. The following figure is the noise image sample. The left binary images are noisy images and the right binary images are after the noise reduction (Figure 9 [12]).

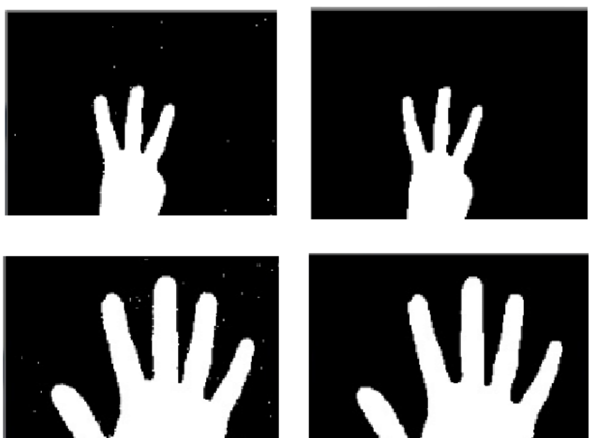


Figure 9 Sample of noise image and after noise reduction

3.1.2. Tracking

After image preprocessing is complete, hand detection can be performed. In fact, there are many detection methods. Furthermore, detection methods can be used for tracking if the program runtime for operating on images is fast enough. Use the frame-by-frame method for tracking. Here are some detection techniques:

- Object Shape

  After sophisticated post-processing image pre-processing, removed the occlusions or useless objects from the background and ensure the image quality. Then, a contouring algorithm will be used based on edge detection results in a large number of edges to extract the contours of hand objects in the image (Figure 10 [13]).

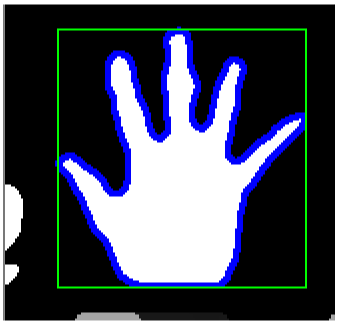


Figure 10 Result of hand contour detection

3.1.3. Recognition

For gesture recognition, the goal is to judge what action to do by reading your hand position, posture, and movement. Moreover, using vision-based hand gesture recognition can be further classified into 2 types, namely static and dynamic. Static gesture recognition means the hand position and posture without change as time goes on. On the other hand, dynamic gesture recognition means then considered as a path between an initial state and a final state. So, it must have a time factor. Here are some examples where hand recognition can be achieved:

- Dynamic Time Warping Algorithm

  The idea of dynamic time warping (DTW) is for measuring the similarity between two temporal sequences. In fact, if only need to do the simple measuring on two different arrays and analyze how they are similar. It can be done by using a very classic algorithm call Euclidean Matching (Figure 11 [21]). However, Euclidean Matching Algorithm adopted one-to-one match so that is not well used in dynamic hand recognition. Hence, a new algorithm (DTW) was developed. DTW can fix the problem of time series because it builds one-to-many and many-to-one matches to find minimized total distance (Figure 12 [22]). To sum up, dynamic time warping (DTW) may be a solution for dynamic gesture recognition.

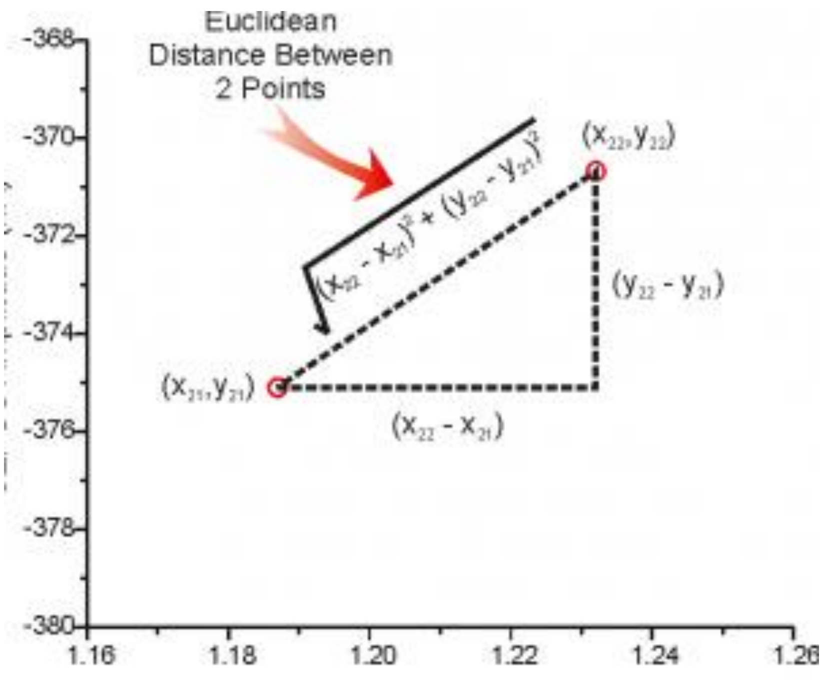


Figure 11 Euclidean Matching Algorithm

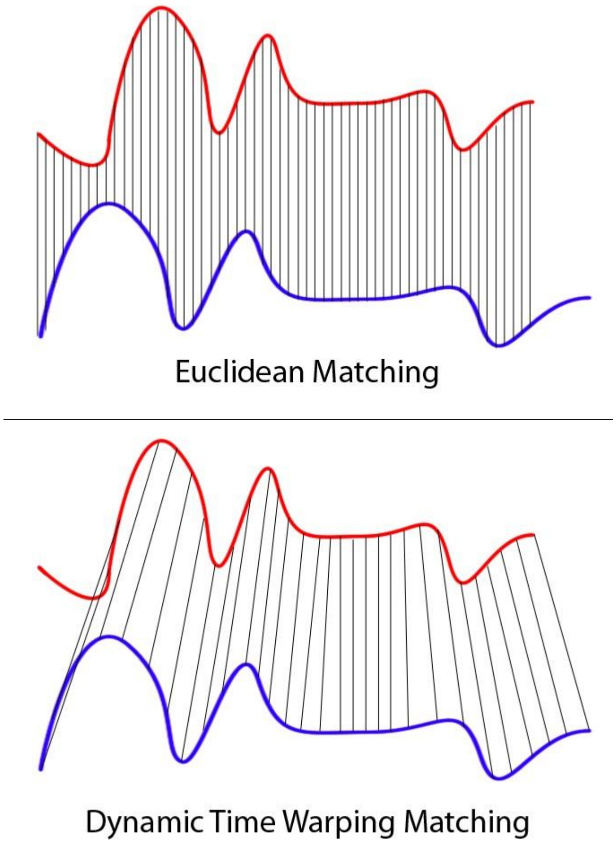


Figure 12 Dynamic Time Warping Algorithm

➢ Support Vector Machine

Using a Support Vector Machine (SVM) also can do gesture recognition. SVM is a kind of machine learning used in classification. The principle is passing the input data to some high-dimensional space. Trying to find the best classification or regression in a high-dimensional space. The following figure (Figure 13 [23]) is to explain how to classify using SVM in the 2D world. If in higher dimensional 3D, 4D, 5D……nD it is difficult to find the solution in our real world because for dimensional more than 3D we are incomprehensible and imaginable. However, SVM can project real-world problems into mathematical problems and put them into different dimensions to find solutions. The following figure (Figure 14 [23]) is to explain using

SVM to upgrade 2D data into 3D data. To sum up, a Support Vector Machine (SVM) may be a solution for static gesture recognition.

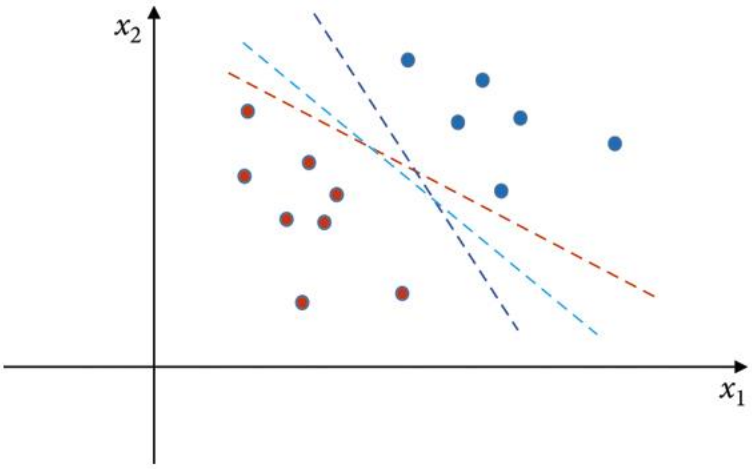


Figure 13 Example of classifying in 2D dimension

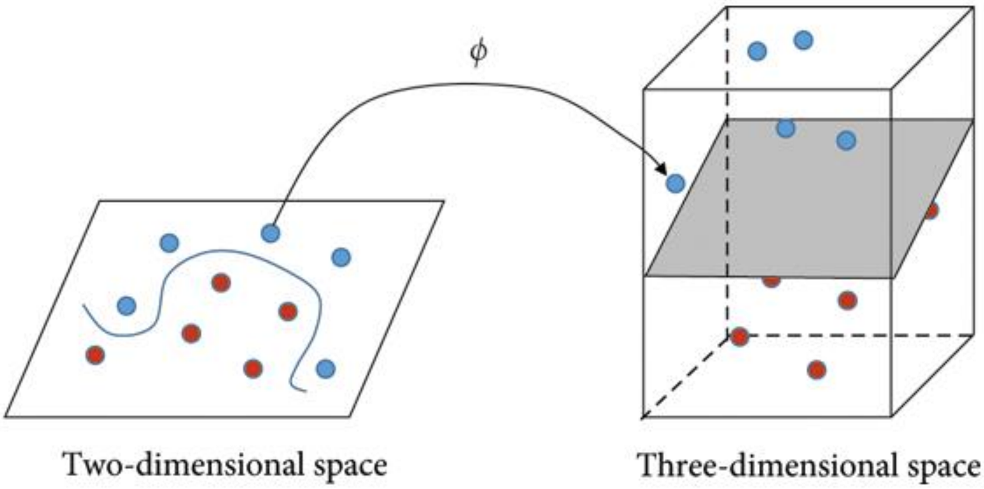


Figure 14 Mapping data from two-dimensional space to three-dimensional space

3.2. Proposed 3D Hand Model Solution

Actually, the most challenging part is the recognition, because this part will directly affect the quality of this project. Although for the above analysis we can use different types of recognition models like we can train a new model to identify A-Z English words and 0-9 numbers. For this method, a very large data set is required and this approach is not the most efficient. So, in this project, a solution for coordinate calculation using a 3D skeleton is proposed.

3.2.1. 3D Hand Skeleton Model

Considering the need for coordinate calculation, it is suitable to use machine learning (ML) to extract 3D key points and finger tracking from the video frames. Practically, an open-source cross-platform framework called MediaPipe has achieved different kinds of AI functions, and one of the functions is 3D Real-Time Hand Tracking supply skeleton key point object. For the model architecture, it can be divided into 2 sections, they are called palm detector and hand landmark model (Figure 15 [17]).

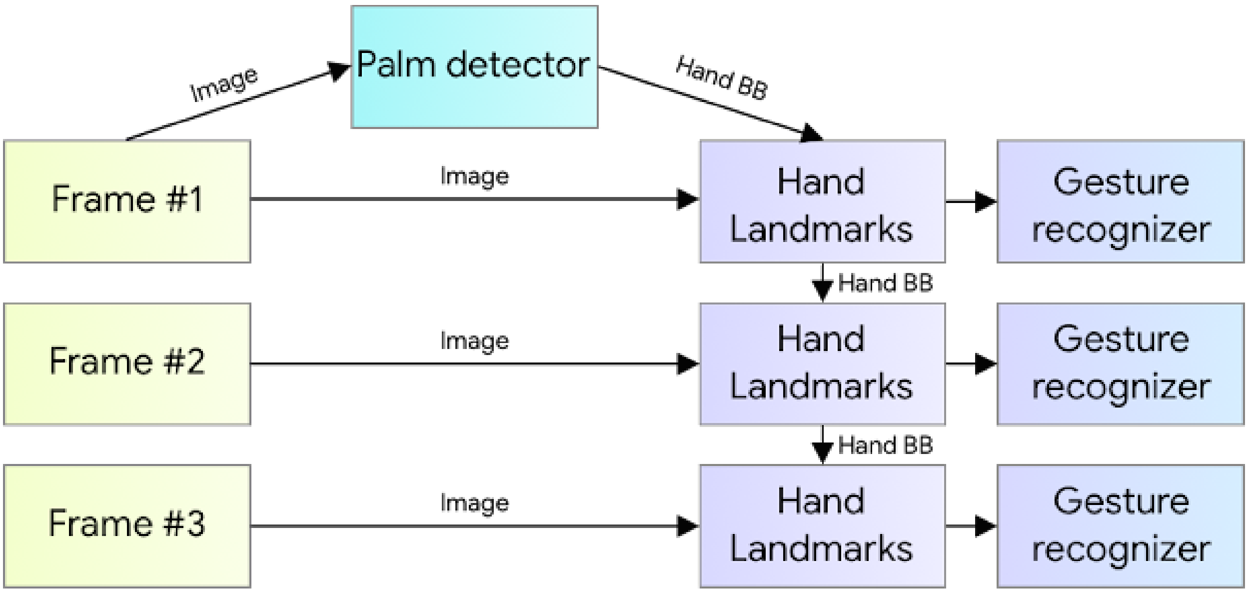


Figure 15 Hand perception pipeline overview

3.2.2. BlazePalm Detector

Detecting hands is a very complex task, and hands are not as simple as face inspection. Because the face has fixed features such as eyes, mouth, and nose, but a hand doesn’t have these fixed features. It makes hand detection greatly increases the uncertainty of predictions.

Consequently, BlazePalm detector had improved the detector changed from the original hand detector to a palm detector. The reason for choosing palm detection is that estimating the bounding boxes of these relatively smaller objects like palms and fists is much simpler than detecting hands with articulated fingers.

Moreover, since it examines smaller objects, the hand non-maximum suppression algorithm can still work well even when the hand is occluded. The non-maximum suppression algorithm helps us to eliminate redundant objects box to find the best box when we are detecting the object (Figure 16 [18]). For details, please refer to the following non-maximum suppression algorithm (Figure 18 [18]) and Intersection over Union algorithm (Figure 17[18]).

Figure 16 Sample image before NMS and after NMS

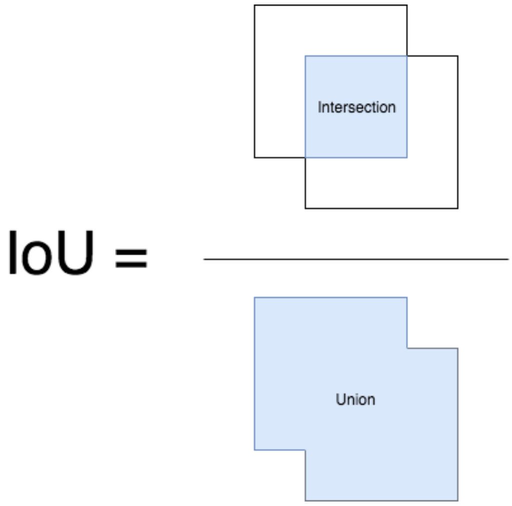


Figure 17 Intersection over Union algorithm

Algorithm 1 Non-Max Suppression

procedure NMS(B,c)
    $B_{nms} \leftarrow \emptyset$ **Initialize empty set**
    for $b_i \in B$ **do** => **Iterate over all the boxes**
        $discard \leftarrow$ False **Take boolean variable and set it as false. This variable indicates whether b(i) should be kept or discarded**
        for $b_j \in B$ **do** **Start another loop to compare with b(i)**
            if $\text{same}(b_i, b_j) > \boldsymbol{\lambda}_{\mathbf{nms}}$ **then** **If both boxes having same IOU**
                if $\text{score}(c, b_j) > \text{score}(c, b_i)$ **then**
                    $discard \leftarrow$ True **Compare the scores. If score of b(i) is less than that of b(j), b(i) should be discarded, so set the flag to True.**
        if not $discard$ **then** **Once b(i) is compared with all other boxes and still the discarded flag is False, then b(i) should be considered. So add it to the final list.**
            $B_{nms} \leftarrow B_{nms} \cup b_i$
    return B_{nms} **Do the same procedure for remaining boxes and return the final list**

Figure 18 Non-Max Suppression Algorithm

Also, BlazePalm used an encoder-decoder feature extractor like FPN (Figure 19 [19]) for larger scene-context awareness even for small objects. Because the low-level feature semantic information is relatively small, but the target position is accurate. Conversely, the high-level features are rich in semantic information, but the target location is relatively coarse. Therefore, high-level palm detector architecture is needed for auxiliary.

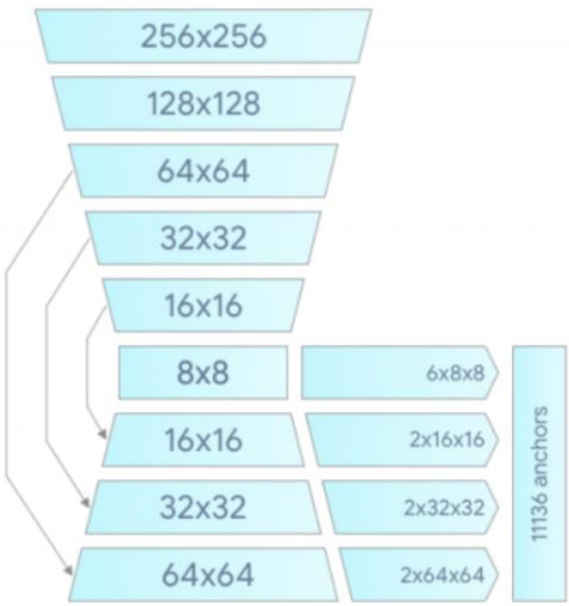


Figure 19 Palm detector model architecture

With the above techniques, the research paper achieves an average precision of 95.7% in palm detection. Using a regular cross entropy loss and no decoder gives a baseline of just 86.22% [17].

3.2.3. Hand Landmark Model

When the palm detection is completed, the hand landmark model will be followed by direct coordinate prediction. The coordinates returned will be 21 3D hand x, y, and z coordinates in the hand area for accurate key points positioning. The following is the corresponding 21 3D hand key point number and description (Figure 20 [20]).

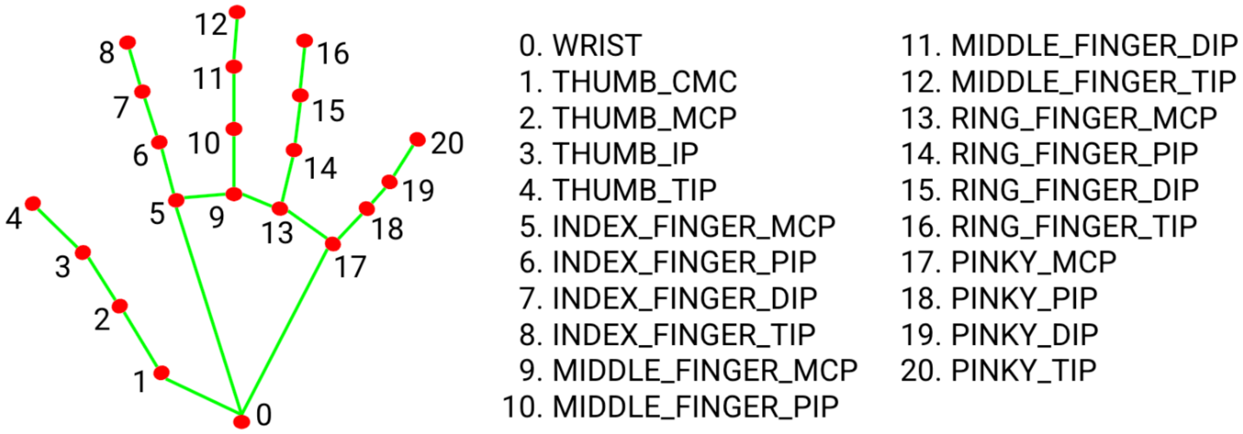


Figure 20 21 hand landmarks

In addition, the dataset used about 30K real-world images with 21 3D coordinates and a high-quality synthetic hand model over various backgrounds to map it to the corresponding 3D coordinates (Figure 21 [17]) and used a high-level model for training (Figure 22 [17]). Finally, after applied the performance boost used synthetic hand it got a better result. The figure below summarizes the regression accuracy (Figure 23 [17]).

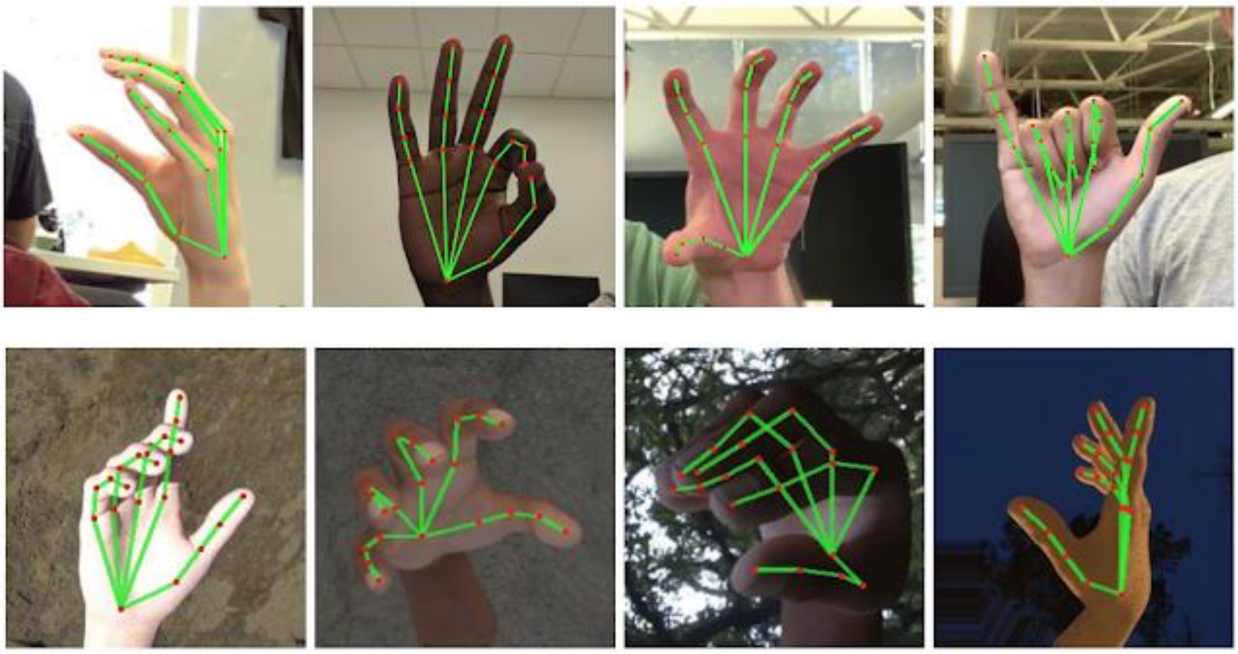


Figure 21 Sample dataset of real-world images and synthetic hand images

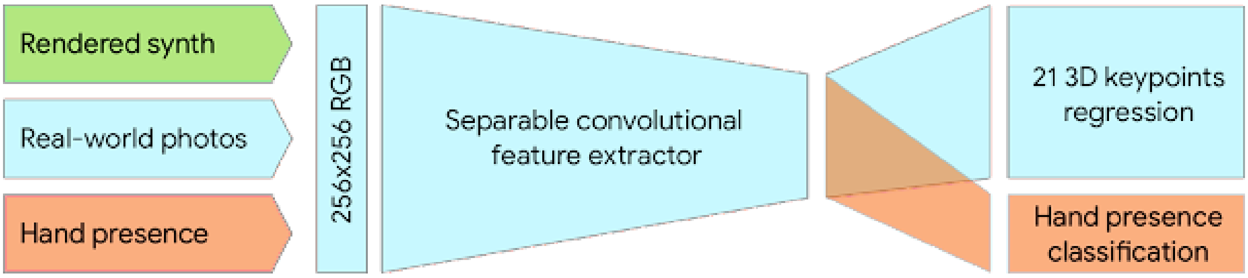


Figure 22 A high-level model training diagram

Dataset	Mean regression error normalized by palm size
Only real-world	16.1 %
Only rendered synthetic	25.7 %
Mixed real-world + synthetic	**13.4 %**

Figure 23 Summarizes of regression accuracy

3.3. Proposed Gesture Control Solution

In this part, I will introduce how to combine coordinate calculation and a 3D skeletal model to create a virtual mouse and virtual keyboard. First, I will explain what the detailed design and the ideas are. Next, how to provide different computer functions based on the virtual mouse and virtual keyboard.

3.3.1. Virtual Mouse Design

Considering the user's usage habits and experience, the design should be humanized, simple and easy to use. Therefore, the idea is to use a design that resembles a laptop trackpad (Figure 24).

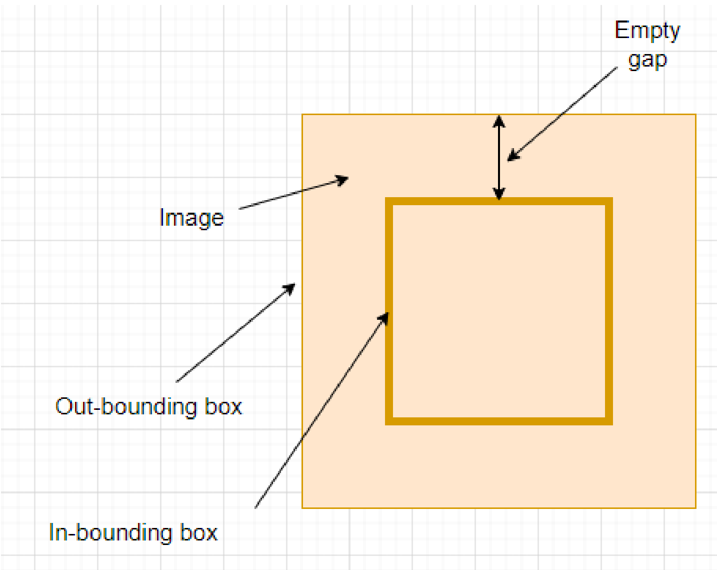


Figure 24 Virtual Mouse Pad

As you can see, the orange color area is the frame that will display which is captured by the webcam. The system will resize the frame and display full size on the out-bounding box. Next, is drawing the in-bounding box on the image. And the mouse controls are only effective inside the in-bounding box. The reason why the mouse control is only valid for the in-bounding box is that it may not work if the control is on the out-bounding box in some special cases. Such as the below figures (Figure 25; Figure 26). Where the user controls the top right and bottom right, it will make model miss detection because the captured frame doesn't recognize the whole hand and it affects the accuracy of detection. So, I use the in-bounding box to fix the problem (Figure 27), using the empty gap to detect half of the hand.

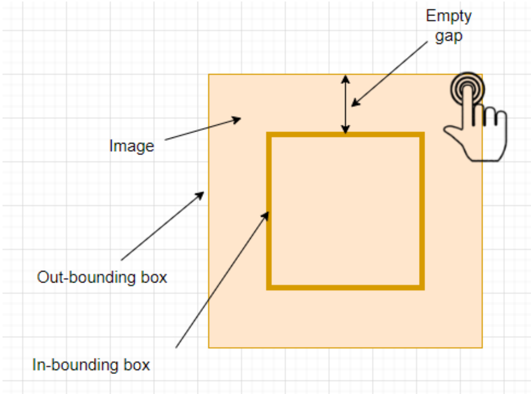


Figure 25 In-Valid Control (1)

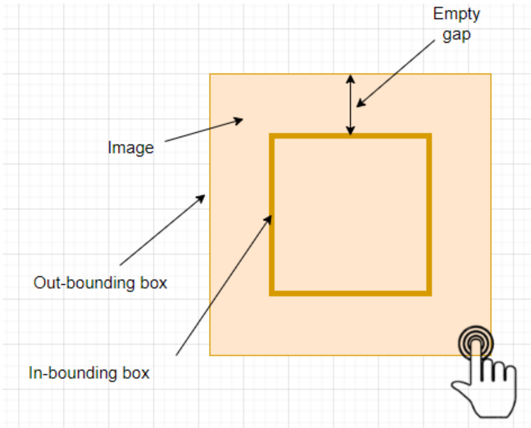


Figure 26 In-Valid Control (2)

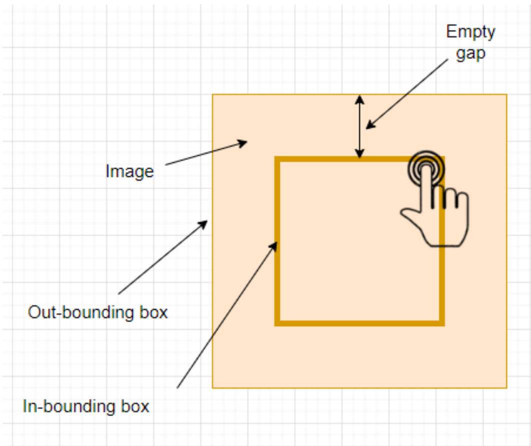


Figure 27 Valid Control (1)

3.3.2. Virtual Keyboard Design

A virtual keyboard (Figure 28) can use the same settings as a virtual mouse. Draw all the buttons on the image, user can select or click the virtual button by hand. So, in this way can provide a computer keyboard input function.

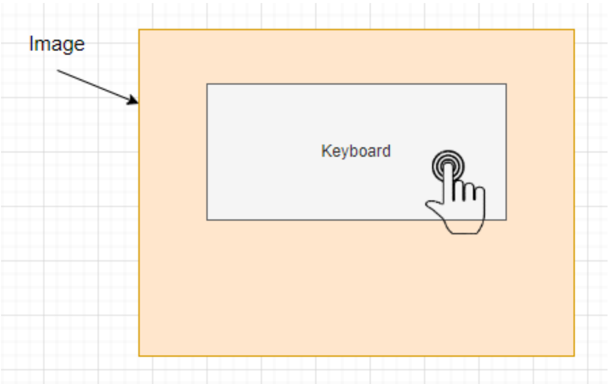


Figure 28 Virtual Keyboard Pad

For each button design, it will have 2 designs. One is the selected design (Figure 30), which is pointing user whose current hand position has selected this button. The second design is unselected (Figure 29), which means the user had no selection on the button and shows a normal design.

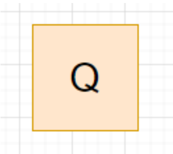


Figure 29 Q Button on No Selection

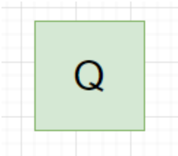


Figure 30 Q Button on Current Selected

Also, for the alphabetic sort order displayed on the keyboard, it uses an order called "QWERTY layout" which is the same as a computer keyboard. This is because on this sorting order format, it provides the best experience. Reduce the learning time for first-time use, minimize the distance of finger movement with their habit and have faster typing speed.

3.3.3. Virtual Mouse Actions and Keyboard Actions

After defining the virtual mouse design and virtual keyboard design, that is apply different gesture actions of mouse functions and keyboard functions. In fact, the system only needs 3 actions for the mouse and keyboard for common use including “left-click”, “moving” and “change mode”. But here the challenge is how to use key points to design the hand patterns. One of the solutions is to calculate each key point and compare the key point coordinates to check if each finger is up or down.

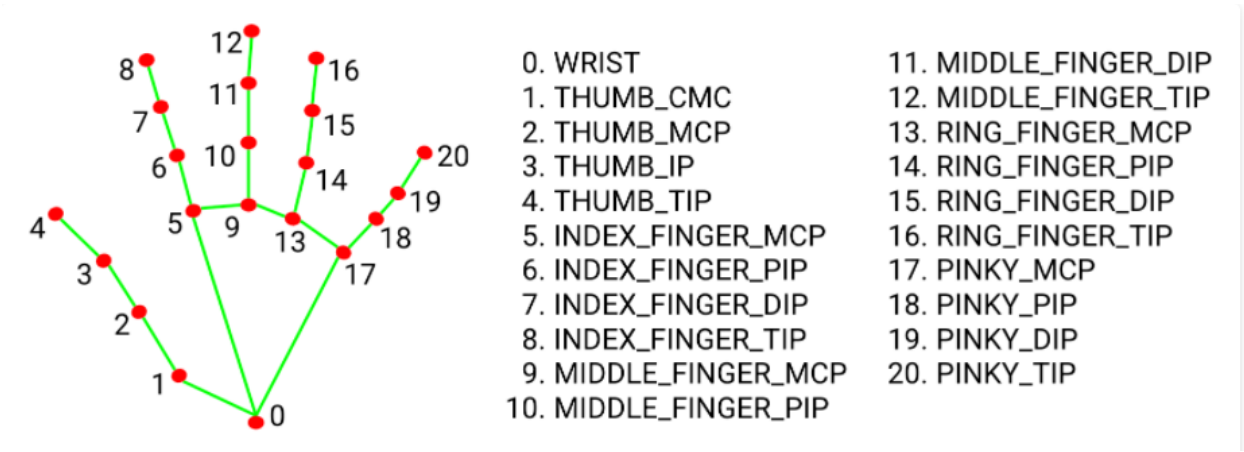


Figure 31 21 Hand Landmarks

The figure above shows the relationship of each key point, for each key point it returns (X, Y, Z) coordinates. To detect finger up or down patterns, we can analyze the coordinate changes. For each finger the first pair of key points like (4,3), (8,7), (12,11) … should be the first point on the Y-axis > the second point on the Y-axis when the finger is up. It indicates the Y-axis key points (4>3), (8>7), (12>11), (16>15), (20>19). Conversely, when the finger moves down, the Y-axis key points should change to the opposite (4<3), (8<7), (12<11), (16<15), (20<19). In this way, we can get the patterns using an easy method. Also, based on this calculation it can represent a number of each finger 1 is up and 0 is down. Finally, the hand finger up-down detection will return 5 digits code like (0,1,0,0,0) to express the up-down of (thumb, index finger, middle finger, ring finger, little finger). In this case (0,1,0,0,0), it means only the index finger up in 5 fingers. Now, we can define the gesture actions, below is the description:

- Left-Click

  For left-click, the pattern of 5 digits code is (1,1,0,0,0) which is the thumb finger and index finger both are detected up then the system will do the left-click function (Figure 32) on the computer.

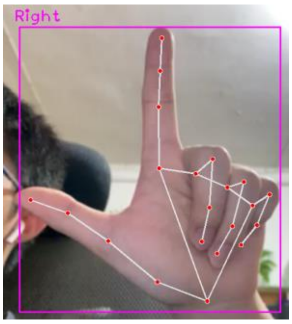


Figure 32 Left-Click Gesture

- Moving

  For moving, the pattern of 5 digits code is (0,1,0,0,0) which is only the index finger is detected up then the system will keep tracking as the mouse moves on the computer. The key points of index number “8” X and Y coordinates are used as pointer locations. Once the user moves the hand position, the key point “8” coordinate will change. So, it can do it as a pointer moving action (Figure 33).

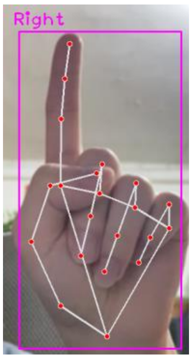


Figure 33 Moving Gesture

- Change Mode

Sometimes users may need to change the mouse mode or keyboard mode where their inputting. So, a new gesture is defined to make user can quickly change the mode. The action can be done by the pattern of 5 digits code (1,1,1,1,1) which is all the fingers are detected up then the system will do the change mode function on the computer (Figure 34).

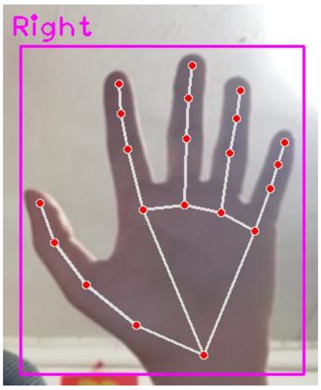


Figure 34 Change Mode Gesture

3.4. Proposed Gesture Authentication Solution

In this section, I mainly describe how to use gestures for authentication and what solutions are used. Here the solution is using Personal Identification Number Gesture for the gesture password with Deep Neural Networks for the gesture. In the following parts is the explanation:

3.4.1. Personal Identification Number Gesture (PING)

A Personal Identification Number (PIN) is a unique numeric code used to verify and authenticate an individual's identity. In this project, it is very suitable to solve the authentication problem. Since the PIN code is easy to remember, the system only requires the user to enter four digits as the password.

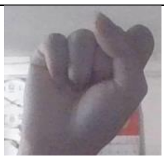
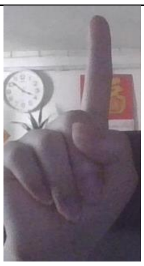
Also, consider the learning time for users to use the system for the first time. A PIN is the most commonly used in the real world (Figure 35), such as mobile phones, computers, ATMs, etc. Therefore, the user has the experience of how to use it. This allows users to authenticate quickly and easily, reducing wait times and improving user experience.

Additionally, a PIN is a unique number that can be securely verified. Each user has a unique PIN, making it difficult for others to impersonate them.

Figure 35 Personal Identification Number (PIN)

Overall, the use of a Personal Identification Number (PIN) is a simple yet effective way of authentication method. Therefore, the project suggested extending the Personal Identification Number (PIN) technology to Personal Identification Number Gesture (PING).

Try to use a set of gestures representing the number 1-9 (Figure 36). In the hotel check-in process, users can customize their own PING password and use the same PING password to unlock the hotel room door.

Number	Gesture
0	
1	
2	
3	

4	
5	
6	
7	
8	
9	

Figure 36 Gesture Authentication Table

3.4.2. Deep Neural Networks (DNN)

In the previous section, the method of using Personal Identification Number Gesture (PING) was introduced. The next part is how to identify different digital gestures. In fact, the easiest way is to repeatedly use the key point coordinate to check if each finger is up or down which I have introduced and used in Virtual Keyboard.

But using the same method on gesture authentication may have drawbacks because at the security level a room door unlocks process is the most important part. The system must have a reliable model for the detection. For this reason, I choose to use the AI method to train a Deep Neural Networks (DNN) model to solve the real word authentication issue.

From a high level to training a DNN model includes 2 main parts: data acquisition (Figure 37) and learning process (Figure 38).

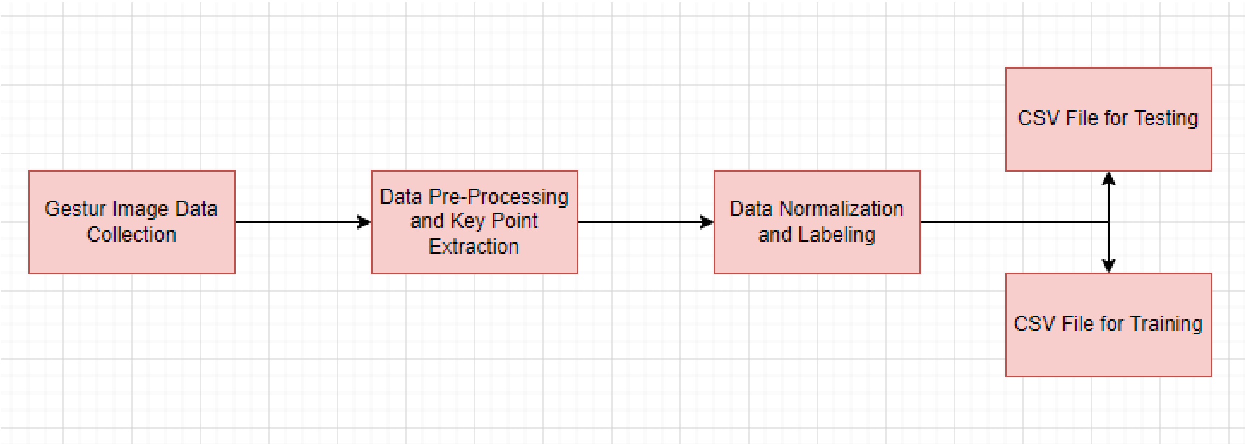


Figure 37 Data Acquisition

The steps of data acquisition are below:

1. Gesture Image Data Collection

   This 1st step is to collect training data and testing data that will be used for model training and testing, and one way is to collect relevant data from the internet. But collecting data from the internet is not the best solution in my point of view. Because collecting data from the internet, it is difficult to ensure the consistency and reliability of the data. This makes comparing and analyzing data more challenging, and there is a lack of control over data quality. Also, it may have legal and ethical issues. Internet data are raising legal and ethical issues related to privacy and copyright infringement. Certain websites may prohibit unauthorized crawling or use of their content.

   So, I chose to collect the data by myself, using my dataset. I will create a Python program for webcam capture. For each category of gestures, I will capture 1,000 images for training and 200 images for testing. In the end, I will have 9 (gesture classes) X 1,000 (training images) + 9 (gesture classes) X 200 (test images) = 10,800 images for training and testing, which is the "raw data".

2. Data Pre-Processing and Key Point Extraction

   This 2nd step is to extract all valid images on the raw data images using the 3D hand skeleton model. Extraction can be done by looping through all raw data images using a Python program, each image will return a total of 21 (points) X 3 (positions of x, y, z) = 63 values on one record. The next step is to do data pre-processing for dimensionality reduction. Remove the location of the Z value, because the z value is useless in my case. Gestures only need X and Y for positioning and not Z position. Furthermore, reducing features can less the complexity of training, reduces the computation time required for training, and improves model accuracy due to less misleading data.

3. Data Normalization and Labeling

   The 3rd step is normalizing all the data between 0 ~ 1 values and doing the labeling of each gesture class. Data normalization is an important step in DNN training because it helps to ensure that the input data is on a similar scale and range. Normalizing the data ensures that all features are given equal importance during training, which can improve the accuracy of the model. This makes it easier for the model to learn and make accurate predictions. And below is the formula used in normalization on each key point.

$$X' = X / \text{image width}$$

$$Y' = Y / \text{image height}$$

4. CSV File Format

   The 4th step is to save training data and testing data in CSV format for easy analysis, loading, reading and future use.

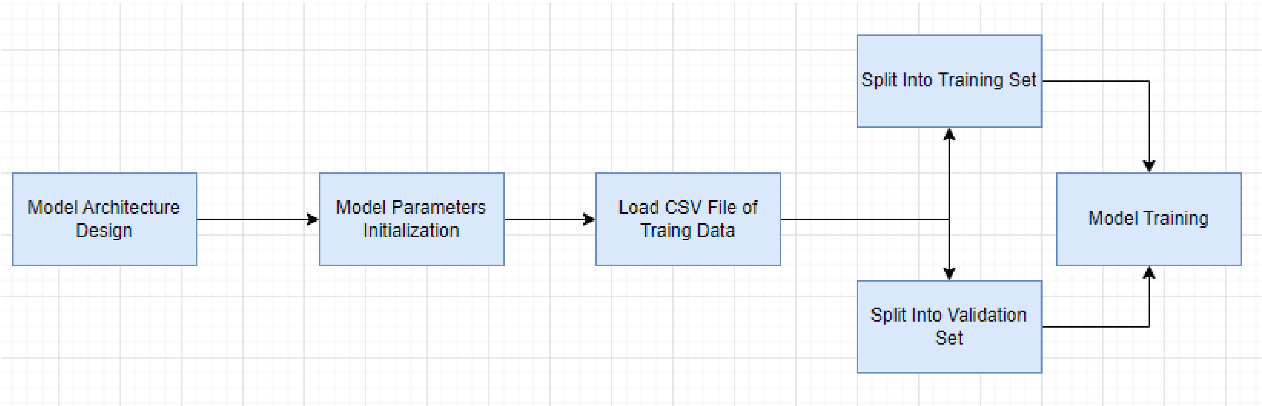


Figure 38 Learning Process

The steps of the learning process are below:

1. Model Architecture Design

   The 1st step is building a sequential model to connect all the input layers, hidden layers and output layers. In this model, the input layer is connected to the 1st hidden layer -> 2nd hidden layer -> .... -> output layer. The connections between each layer are fully connected so that each neuron in one layer is connected to every neuron in the next level layer and built it sequentially. So sequential model allows for complex relationships in building the network.

2. Model Parameters Initialization

   The 2nd step is initializing the model parameters such as activate function, loss function and optimize function.

   Choosing the activation functions is used for the output of a given neuron's input and captures the complex relationship between input and output. Choosing an appropriate activation function can significantly affect the performance of the model.

Choosing the loss functions is used to measure the performance of a model by comparing its predicted values with actual values. It provides feedback to the optimizer on how to adjust the weights of the model to improve its performance. Choosing an appropriate loss function can also significantly affect the accuracy of the model.

Choosing the optimizer function is used to update the weights of the model according to the loss function. It determines how much weight each parameter update should be given and helps prevent overfitting during training. Choosing an appropriate optimizer function can affect the speed and accuracy of the model.

3. Training Set and Validation Set

The 3rd step is to split the dataset into a training set and a validation set. This step can help monitor the performance of the model on the validation set during training and prevents overfitting and tunes hyperparameters accordingly.

3.5. Website Architectures

The following is the server architecture required by the website:

Database Server	MY SQL
Web Server	Apache

I chose MY SQL + Apache as a combination. This is because I need a web server to provide information browsing services for unmanned hotels on the internet, which will run PHP code. Also, I need a database to store data such as hotel room information, account information, etc. Therefore, the MY SQL + Apache combination is the easiest to install because they are free and already have a collection of installation packages, namely XAMPP (Figure 39 [15]). In my case, I'm using only one computer running as a database server, web server, and gesture recognition system for Python code, so XMPP can support these architectures.

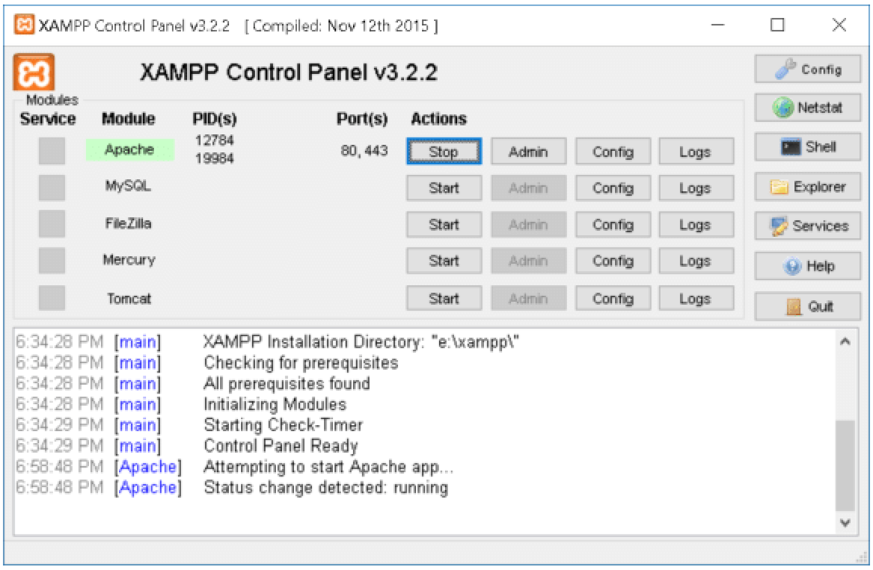


Figure 39 The main graphic user interface of XAMPP

3.6. Database Structure (ERD)

A good database design for a hotel online booking system should adhere to the following principles. Firstly, it should be normalized, meaning data elements within the database should only appear in one place to reduce redundancy and inconsistency. Additionally, it should be designed intuitively, so items are stored logically and efficiently such as arranging tables hierarchically according to the application's purpose and related types of data. Finally, relations between different entities within the system should be established using primary keys and foreign keys where necessary. This would provide flexibility whilst preventing additional errors from occurring.

Specifically, information about customers such as name and contact details would need to be stored separately from bookings that link customers with rooms or reservations with certain dates. The main requirements of a good design also include support for scalability since the needs of customers can vary greatly over time and a well-organized structure makes it easier if new features or functionalities need to be added later on. The system needs to have a good understanding of the relationship between each table in order to create a successful online booking system.

Additionally, the system needs a place to store room images. Since these are multimedia data, the system needs a special way to save it. In particular, room showcase images cannot be stored directly in the database. Because the system will take a long time for the query and wastes database space. In order to solve the problem of saving multimedia data, the idea is to use the URL method. The solution should be to store the images in the web server folder. We fetch images from web servers using URLs and show images stored in web servers using URLs.

Based on the above principles and requirements can analyze a completed database design. To summarize the analysis, the system should need the following tables: Customer Information Table, Rooms Information Table, Booking Record Table and Room Service Table. And their relationship should look like the below ER diagram (Figure 40):

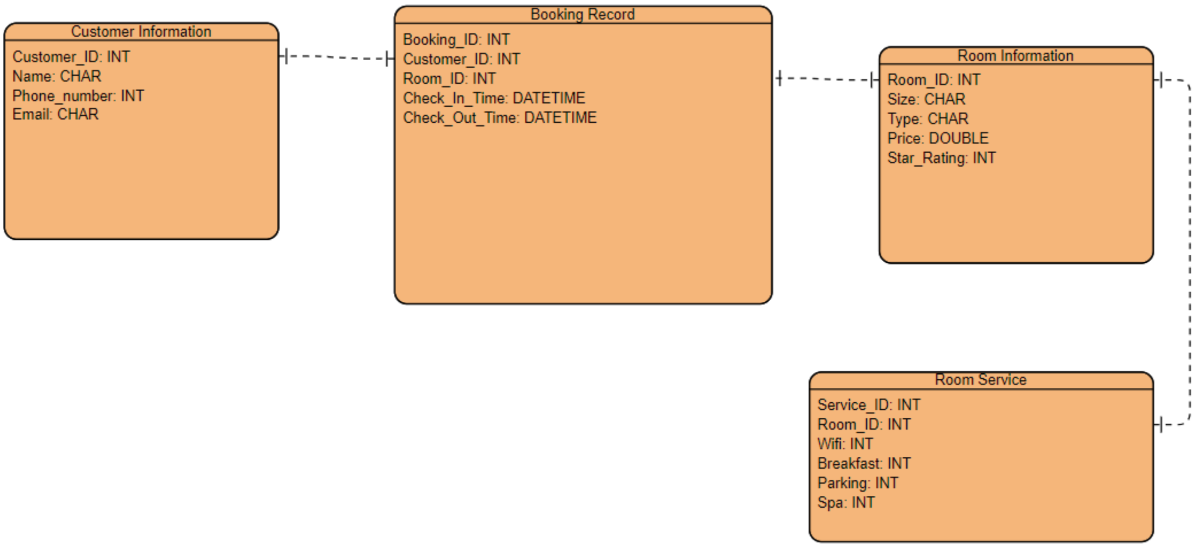


Figure 40 ER diagram

3.6.1. Customer Information Table

Description	Column Name	Data Type	Remark	Can be NULL
A unique ID that identifies the record	Customer_ID	INT	Primary key (PK)	N
Customer names of users who registered to use the system	Name	CHAR		N
HK phone number	Phone_number	INT		N
Email address	Email	CHAR		N

Customer Information Table: This table contains all the information of customers. It has columns like name, phone number and email of customers who have booked a room in the hotel. Consider that the above data will be used in the check-in process. Therefore, all input data cannot be set to NULL values. Below are some example records represented in the database:

Customer_ID	Name	Phone_number	Email
000001	Fung Si Fong	53242354	abc@gmail.com
295425	Chan Siu Man	39565123	ferf44@gmail.com
532256	Chan Tai Man	61205935	rece32@yahoo.com.hk

3.6.2. Room Information Table

Description	Column Name	Data Type	Remark	Can be NULL
A unique ID that identifies the record and this also express the room number	Room_ID	INT	Primary key (PK)	N
Room size and the possible value can be S/M/L	Size	CHAR		N
Room type and the possible value can be Single/Double/Triple/Quad	Type	CHAR		N
The star rating of this room	Star_Rating	INT		N
Room price	Price	DOUBLE		N

Rooms Information Table: This table contains all the information related to rooms of the hotel, including room type, room number, size, price, star and image to present in each room. Actually, the online booking website should display additional information for each room. First, the showcase image will be saved in the web server folder. To display the corresponding image, it only needs to use the Room ID to query, for example, to get the room id 010043 showcase images, the format name is "010043_01.jpg, 010043_02.jpg, 010043_03.jpg, ...". Next, if need to display the corresponding room location, it can also use the Room ID to query, for example, to get the room id 010043 locations, the format is the first two digits as floor “01/F” and the remaining digits as room number “0043”. Lastly, the currency format of the database is uniformly stored using HKD ($) format. Consider that the above data will be used for the check-in process and online booking information. Therefore, all input data cannot be set to NULL values. Below are some example records represented in the database:

Room_ID	Size	Type	Star_Rating	Price
010043	M	Single	4	300
144534	S	Double	3	150.2
068632	L	Triple	5	503.7
076341	L	Quad	4	600

3.6.3. Booking Record Table

Description	Column Name	Data Type	Remark	Can be NULL
A unique ID that identifies the record	Booking_ID	INT	Primary key (PK)	N
The relationship for the booker for this booking	Customer_ID	INT	Foreign key (FK)	N
The relationship for which room is reserved for this booking	Room_ID	INT	Foreign key (FK)	N
The date time for check-in	Check_In_Time	DATETIME		N
The date time for check-out	Check_Out_Time	DATETIME		N

Booking Record Table: This table captures booking records for each customer and hotel information like check-in and check-out times. There will be relations between Customer Information Table of each booking with a customer whose info was provided in Customer Information Table and Rooms Information Table of each booking with a particular Room. Through this data relationships, can link up together using Foreign key (FK), which makes it easily analyze booking patterns across different users or specific rooms over time. Consider that the above data will be recorded during the check-in. Therefore, all input data cannot be set to NULL values. Below are some example records represented in the database:

Booking_ID	Customer_ID	Room_ID	Check_In_Time	Check_out_Time
0423432	000001	010043	2020-01-01 10:10:10	2020-01-04 10:10:10
1231233	546721	065757	2022-10-6 17:25:00	2022-10-7 17:25:00
7954654	342759	034242	2022-10-6 17:25:00	2022-10-7 17:25:00
9534553	342342	035351	2022-10-6 17:25:00	2022-10-7 17:25:00

3.6.4. Room Service Table

Description	Column Name	Data Type	Remark	Can be NULL
A unique ID that identifies the record	Service_ID	INT	Primary key (PK)	N
The relationship for which room is reserved for this booking	Room_ID	INT	Foreign key (FK)	N
The service of free WiFi	wifi	INT		N
The service of free breakfast	breakfast	INT		N
The service of free parking	parking	INT		N
The service of free spa	spa	INT		N

Room Service Table: This table is to link up each room and its provided services. For each service, the value can be 0 or 1. When the value is 0, it means this room doesn't have the corresponding service. On the contrary, when the value is 1, it means this room has the corresponding service. Consider that the above data will be used on the website. Therefore, all input data cannot be set to NULL values. Below are some example records represented in the database:

Service_ID	Room_ID	wifi	Breakfast	Parking	spa
S1	010043	1	1	1	1
S2	065757	0	0	0	0
S3	034242	1	0	1	1

3.7. Coding Language and Environment

For the software part, I need to use different programming languages for the website and gesture recognition. Therefore, strong technical tools support is a must. The following are the main software composition and tools:

3.7.1. Programming

- Main Programming Languages
  - Python (3.7.0)

    Python is an interpreter, high-level, and general-purpose programming language. So, it was used to develop an AI part application in my project which is gesture recognition and gesture authentication.

  - PHP

    PHP will be suitable for web development and used in embedded HTML and CSS. PHP will be mainly responsible for developing dynamic pages. The dynamic web page will be a UI to interact with the user.

- Main Library or API
  - OpenCV

    Since this project involves the field of computer vision and is committed to the development of real-time image processing, computer vision. For example: convert every second frame into images and then process them.

- CVZone

CVZone is a computer vision package it can be easy to run image processing and call the AI model and its related functions. Moreover, the core of CVZone uses the OpenCV and Mediapipe libraries. So, don't have to worry about compatibility issues. Therefore, I will use the CVZone API to handle gesture recognition and gesture authentication.

- Database Management
  - SQL

Another important thing is how the data is stored. Since the data will be shared with websites and public machines in both regions, the data must be accurate, consistent, and reliable. If a poorly designed database might make it more difficult to access the information or jeopardize the accuracy of data. It is recommended to use a relational database because our database data record tables and tables should be relational. Such as a hotel room remained by a hotel guest, they have a one-to-one relationship. Since this project will be using the relational database structure for this project. Therefore, SQL will be used to manage data processing in relational database management systems. It contains data insertion, query, update and delete, etc.

3.7.2. Version Control and Virtual Environment

- Environment
  - IDE:

    An IDE is a coding platform that will be used for program development, testing, or debugging.

Website	Brackets

Gesture Recognition and Gesture Authentication	PyCharm

- Virtual Environment Tools

<table>
<tr><td>Python version</td><td rowspan="2">Anaconda</td></tr>
<tr><td>Library</td></tr>
</table>

4. Implementation

4.1. Live Video Streaming

Since the method of gesture recognition is to use a webcam, it is different from gesture recognition using wearable devices. Considering it is a computer vision problem, we need to catch and transfer the real-world image to computer image data. The solution can be done by two-part camera detection and video frame.

4.1.1. Camera Detection

First, the program will open the default front-facing camera on the computer using the OpenCV API function called VideoCapture() with passing default parameter 0. And the system will pop up a window with a size of 640x480. The next process is resizing the window and enlarging the image to 160% of its original size. The reason for enlarging the image is when implementing a virtual mouse and virtual keyboard, there needs to have enough space to display the relevant keyboard words, numbers, and mouse moving panel.

4.1.2. Video Frame

When the first part of camera detection runs successfully then we use the OpenCV API function read() to keep checking the camera video to each frame image. Afterward, image processing is performed on each captured image, zoomed in by 160%, and flipped horizontally using the OpenCV API function flip() with passing default parameter 1 before displaying it on the window. Finally, repeating the step of the video frame process above and can become a video stream. To do that we can put the above process in an infinite while loop which is while(true).

4.2. Hand Detection

For hand detection, we are using the CVZone API function. CVZone is a package that runs image processing and AI functions. It's using the OpenCV and Mediapipe libraries as the core. In this project, I used HandTrackingModule class to assist in hand detection. The following are the steps:

4.2.1. Hand Detector Object

To use the hand tracking module, we need to use the HandTrackingModule class to create a hand detector object called "detector". This process can be done by using HandDetector() function by passing two parameters (detectionCon=0.7, maxHands=1). Hand tracking will set the maximum number of hand detections to 1 hand. Because whether controlling the UI or customizing the authentication pattern, applying 1 hand-solving method, can provide a more stable performance.

4.2.2. Hand Tracking

After creating the hand detector object, we can use the object-oriented method to call the API function "findHands" to keep track of hand computations and detect hand motion and orientation from the input image by passing two parameters (img, flipType=False). The API function will return two values the processed 3D hand skeleton images (Figure 41) and the 3D hand list of data including 21 hand landmarks (each landmark is composed of x, y, and z positions), bounding box position, the center position of the hand and the type of hand. The following is the sample output:

```
{
'lmList': [          [465, 451, 0],
```

[487, 390, -18],
[487, 319, -24],
[455, 269, -25],
[425, 249, -26],
[419, 322, -17],
[376, 279, -23],
[354, 256, -27],
[342, 241, -29],
[383, 352, -10],
[346, 301, -8],
[331, 268, -6],
[323, 246, -6],
[364, 382, -5],
[336, 336, -2],
[327, 305, 1],
[324, 284, 3],
[355, 410, -1],
[327, 378, 0],
[314, 357, 6],
[308, 344, 11]
],
'bbox': (308, 241, 179, 210),
'center': (397, 346),
'type': 'Right'
}

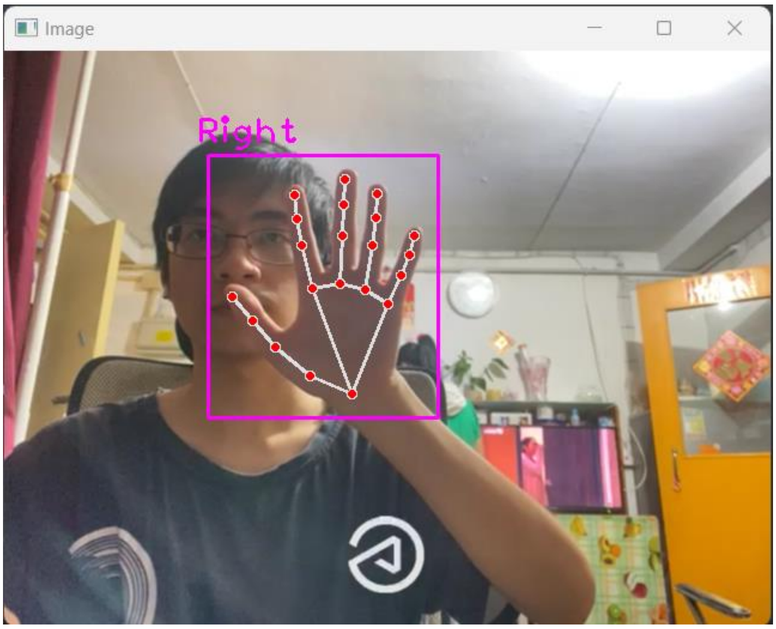


Figure 41 The 3D hand skeleton images

4.3. Gesture Control for Coordination System

For the virtual mouse and virtual keyboard, it is based on the coordination system design. In this context, it will show the coding and how to implement it.

4.3.1. Virtual Keyboard

In coding, I created a new object class file called "ButtonDesign", this class file has some functions to call to create. It provides two functions, one is the "draw()" function to draw the button object on the captured image, and the other is the "checkOnClick" function to check whether the button object is selected by the user's finger-pointing.

Next, in the main class file import the "ButtonDesign" class file and call the function "draw()" to draw each button on the fixed location on the image if the current mode is "keyboard". Also, considering that there should be a delay when calling the "keyboard" library API (keyboard.send()) for text input to type letter by letter and changing the control mode to "mouse". So, to fix the delay problem, I create a counter valuable to count the FPS and it takes 1~1.5s to input a letter.

```
buttonController = []
createButtonList = [["Q", "W", "E", "R", "T", "Y", "U", "I", "O", "P"],
                    ["A", "S", "D", "F", "G", "H", "J", "K", "L", " "],
                    ["Z", "X", "C", "V", "B", "N", "M", "BACK"],
                    ["1", "2", "3", "4", "5", "6", "7", "8", "9", "0"]]

for i in range(len(createButtonList)):
    for x, word in enumerate(createButtonList[i]):
        buttonController.append(Button.Button([100 * x + 50, 170 + i * 100], word, (70 * len(word), 70)))
```

Figure 42 Source Code of Creating Button Object

```python
    for x in range(0, len(buttonController)):
        img = buttonController[x].draw(img)

    if hands:
        # Hand 1
        hand1 = hands[0]
        lmList1 = hand1["lmList"]  # List of 21 Landmark points

        f1_x, f1_y = lmList1[8][0], lmList1[8][1]
        f2_x, f2_y = lmList1[4][0], lmList1[4][1]

        fingers1 = detector.fingersUp(hand1)

        for x in range(0, len(buttonController)):
            img, clicked = buttonController[x].checkOnClick(img, (f1_x, f1_y))

            if clicked:
                if changeKeyboardClickCount[0] == buttonController[x].text:
                    changeKeyboardClickCount[1] -= 1
                    if changeKeyboardClickCount[1] == 0:
                        print(buttonController[x].text)
                        if buttonController[x].text == "BACK":
                            keyboard.send("BACKSPACE")
                        else:
                            keyboard.write(buttonController[x].text)
                        changeKeyboardClickCount[1] = 50
                else:
                    changeKeyboardClickCount[1] = 50
                    changeKeyboardClickCount[0] = buttonController[x].text
```

Figure 43 Source Code of Keyboard Function

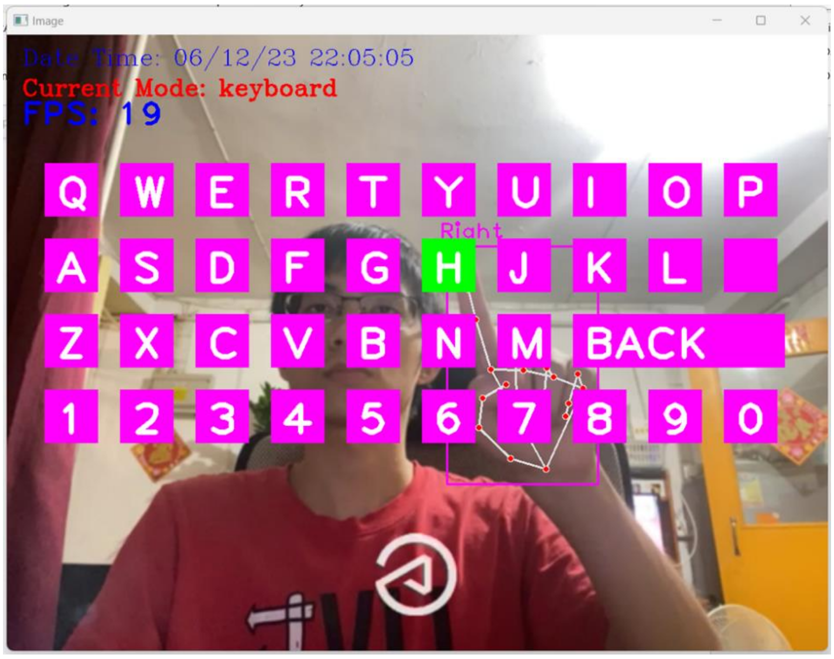


Figure 44 Virtual Keyboard UI

4.3.2. Virtual Mouse

At the same time, if the current mode is “mouse” the program will do the mouse pad drawing using OpenCV API “rectangle()”. To get the 5 digits code of the finger up or down, here is using the HandDetector API (fingerUp()) to check each finger. Based on the 5 digits code and if-else condition, we can use the library “mouse” and “pyautogui” API to provide the mouse moving or left-click. Summarily, considering that there should be a delay in changing the control mode to “keyboard” or left click. Therefore, create a counter valuable to count the FPS to do each action.

```
inaccuracy = 10

if f1_x in range(startPoint[0] - inaccuracy, endPoint[0] + inaccuracy) and f1_y in range(startPoint[1] - inaccuracy, endPoint[1] + in
    if fingers1[1] == 1 and fingers1[2] == 0 and fingers1[0] == 1 and fingers1[3] == 0 and fingers1[4] == 0:
        cv2.circle(img=img, center=(f1_x, f1_y), radius=20, color=(0, 255, 255))
        mouse.move(int((f1_x - (wCam * mousePlanProportion)) / mousePlanBoxSizeW * wScreen),
                   int((f1_y - (hCam * mousePlanProportion)) / mousePlanBoxSizeH * hScreen))
        changeModeCount = 100
        changeMouseClickCount = 50
    elif fingers1[1] == 1 and fingers1[2] == 0 and fingers1[0] == 0 and fingers1[3] == 0 and fingers1[4] == 0:
        cv2.circle(img=img, center=(f1_x, f1_y), radius=20, color=(0, 255, 255))
        cv2.circle(img=img, center=(f2_x, f2_y), radius=20, color=(0, 255, 255))
        changeMouseClickCount -= 1
        if changeMouseClickCount == 0:
            pyautogui.click()
        changeModeCount = 100
    elif (fingers1[1] == 1 and fingers1[2] == 1 and fingers1[0] == 0 and fingers1[3] == 1 and fingers1[4] == 1):
        changeModeCount -= 1
        changeMouseClickCount = 50
        if changeModeCount == 0:
            currentMode = 'keyboard'
            changeModeCount = 100
```

Figure 45 Source Code of Mouse Function

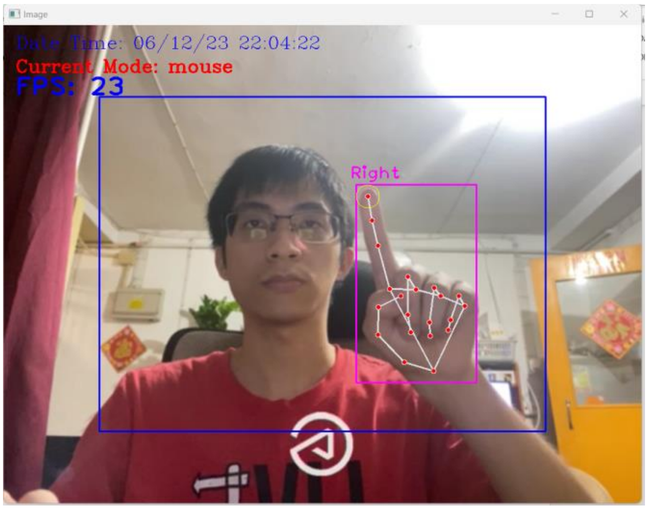


Figure 46 Virtual Mouse UI

4.4. Gesture Authentication for DNN Model

Since gesture authentication needs an AI model. The DNN training used Python code is provided to complete all the tasks including dataset collection, data pre-processing and model training. In the following paragraph, I will show the detailed step on how to train a DNN model for the gesture.

4.4.1. Dataset Collection

First, I wrote a Python program to capture hand images using the OpenCV library. Create a variable "counter" combine with a while loop to continuously capture frames and save them as an image using the OpenCV function imwrite() and loop through the processes 1000 times. For each gesture class, just change the save path of the code and run the same program.

```
import cv2

if __name__=="__main__":
    cap = cv2.VideoCapture(0)
    counter = 0

    while True:
        if counter<=1000:
            success, img = cap.read()
            cv2.imwrite('./dataset/training/img/0/' + str(counter) + '.jpg', img)
            counter += 1
            print(counter)

        cv2.imshow("Image", img)
        cv2.waitKey(1)
```

Figure 47 Source Code for Data Collection

4.4.2. Data Pre-Processing

The next step is data pre-processing to convert raw data into training data. The procedural steps include keypoint extraction and data normalization. First, the raw data image is read using the OpenCV function imread() and loaded with a 3D hand detector to extract keypoints on the image. Then return two variables (hands, img), which are the extracted key point information and the hand skeleton image. Next, get the image width and image height for normalization, which can be done by getting the image shape (index img.shape[0], img.shape[1]) represent (width, height) and saving the value in variables (imgW, imgH). Finally, loop through each key point to normalize the x and y coordinates between 0 and 1 value and removes the z coordinate. In the end, all data will be saved as .cvs files for each gesture class.

```python
def detect_hand(img):
    hands, img = detector.findHands(img, flipType=False)
    return hands, img

def getCSVfile():
    for item in data:
        os.remove('./dataset/training/CSV/' + str(item) + '.csv')
        print('Extracting class ' + str(item) + ' images key point and save as CSV file at \'./dataset/training/CSV/' + str(item) + '.csv\'')
        for filename in os.listdir('./dataset/training/img/' + str(item) + '/'):
            img = cv2.imread('./dataset/training/img/' + str(item) + '/' + filename)
            hands, img = detect_hand(img)
            imgW, imgH = img.shape[0], img.shape[1]

            try:
                with open('./dataset/training/CSV/' + str(item) + '.csv', 'a+', newline='') as f:
                    writer = csv.writer(f)

                    for p in range(0, len(hands[0]['lmList'])):
                        hands[0]['lmList'][p][0] = hands[0]['lmList'][p][0]/imgW
                        hands[0]['lmList'][p][1] = hands[0]['lmList'][p][1]/imgH
                        hands[0]['lmList'][p] = hands[0]['lmList'][p][:2]

                    writer.writerow(hands[0]['lmList'])

                    successProcess[int(item)] += 1
            except:
                failProcess[int(item)] += 1
                continue
```

Figure 48 Source Code for Data Pre-Processing

However, one of the attentions is not all the raw data images can be extracted from keypoints. Therefore, some of the images will fail in the processing and for the failed images it will ignore and not included in my training set. When the program is finished it will output a result of the statistics (Figure 49).

```
Extracting class 0 images key point and save as CSV file at './dataset/training/CSV/0.csv'
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Extracting class 1 images key point and save as CSV file at './dataset/training/CSV/1.csv'
Extracting class 2 images key point and save as CSV file at './dataset/training/CSV/2.csv'
Extracting class 3 images key point and save as CSV file at './dataset/training/CSV/3.csv'
Extracting class 4 images key point and save as CSV file at './dataset/training/CSV/4.csv'
Extracting class 5 images key point and save as CSV file at './dataset/training/CSV/5.csv'
Extracting class 6 images key point and save as CSV file at './dataset/training/CSV/6.csv'
Extracting class 7 images key point and save as CSV file at './dataset/training/CSV/7.csv'
Extracting class 8 images key point and save as CSV file at './dataset/training/CSV/8.csv'
Extracting class 9 images key point and save as CSV file at './dataset/training/CSV/9.csv'



All the extraction is completed, and the processed results are as follows:
Class 0 : 848 are successful image, 152 are failure image
Class 1 : 835 are successful image, 165 are failure image
Class 2 : 875 are successful image, 125 are failure image
Class 3 : 942 are successful image, 58 are failure image
Class 4 : 935 are successful image, 65 are failure image
Class 5 : 540 are successful image, 460 are failure image
Class 6 : 953 are successful image, 47 are failure image
Class 7 : 853 are successful image, 147 are failure image
Class 8 : 938 are successful image, 62 are failure image
Class 9 : 963 are successful image, 37 are failure image
```

Figure 49 Statistics of Data Pre-Process

名稱	修改日期	類型	大小
0.csv	15/7/2023 17:05	Microsoft Excel C...	512 KB
1.csv	15/7/2023 17:05	Microsoft Excel C...	500 KB
2.csv	15/7/2023 17:05	Microsoft Excel C...	526 KB
3.csv	15/7/2023 17:05	Microsoft Excel C...	566 KB
4.csv	15/7/2023 17:05	Microsoft Excel C...	563 KB
5.csv	15/7/2023 17:05	Microsoft Excel C...	326 KB
6.csv	15/7/2023 17:05	Microsoft Excel C...	572 KB
7.csv	15/7/2023 17:05	Microsoft Excel C...	512 KB
8.csv	15/7/2023 17:05	Microsoft Excel C...	564 KB
9.csv	15/7/2023 17:05	Microsoft Excel C...	579 KB
All_Training_Data.csv	15/7/2023 17:05	Microsoft Excel C...	4,341 KB

Figure 50 All The CSV Files for Training

4.4.3. DNN Model

This step is to build a DNN network for training. Here, I have used a Python library called "tensorflow" and its API functions for the creation. First, a sequential model is created using the Sequential() function and stored in the variable called "network". Next, add a Dense() layer as neurons for each hidden layer. Here hidden layer 1 has 42 dense, hidden layer 2 has 30 dense, hidden layer 3 has 20 dense, and hidden layer 4 has 10 dense.

For the first 3 kinds of hidden layers, it uses the "relu" activation function and the last hidden layer uses the "softmax" activation function for multi-class classification. Moreover, consider that it might have overfitting problems during training. So, between each hidden layer connection, I added a Dropout() layer that forgets 20% of what it learned from the forward hidden layer. Finally, the overall model structure is like this (Figure 52):

```
def DNNNetwork():
    network = Sequential()
    network.add(Dense(42, input_shape=(x_data.shape[1],), activation='relu'))
    network.add(Dropout(0.20))
    network.add(Dense(30, activation='relu'))
    network.add(Dropout(0.20))
    network.add(Dense(20, activation='relu'))
    network.add(Dropout(0.20))
    network.add(Dense(10, activation='softmax'))
    network.summary()
    return network
```

Figure 51 Source Code of DNN Network

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 42)                1806
_________________________________________________________________
dropout (Dropout)            (None, 42)                0
_________________________________________________________________
dense_1 (Dense)              (None, 30)                1290
_________________________________________________________________
dropout_1 (Dropout)          (None, 30)                0
_________________________________________________________________
dense_2 (Dense)              (None, 20)                620
_________________________________________________________________
dropout_2 (Dropout)          (None, 20)                0
_________________________________________________________________
dense_3 (Dense)              (None, 10)                210
=================================================================
Total params: 3,926
Trainable params: 3,926
Non-trainable params: 0
_________________________________________________________________
Train on 5209 samples, validate on 3473 samples
```

Figure 52 DNN Model Structure

4.4.4. Training and Evaluation

The last steps are training and evaluating the model's performance. I call the train_test_split() API to split the data into 80% of training and 20% of valuation for each class. Next, call the self-defunded function dataSummaryInfo() to get the overall statistics result (Figure 53).

```
Total number of data for training: 8682
Training data and validation data summary:

Training Set of class 0 has 509 samples
Validation Set of class 0 has 339 samples
Training Set of class 1 has 501 samples
Validation Set of class 1 has 334 samples
Training Set of class 2 has 525 samples
Validation Set of class 2 has 350 samples
Training Set of class 3 has 565 samples
Validation Set of class 3 has 377 samples
Training Set of class 4 has 561 samples
Validation Set of class 4 has 374 samples
Training Set of class 5 has 324 samples
Validation Set of class 5 has 216 samples
Training Set of class 6 has 572 samples
Validation Set of class 6 has 381 samples
Training Set of class 7 has 512 samples
Validation Set of class 7 has 341 samples
Training Set of class 8 has 563 samples
Validation Set of class 8 has 375 samples
Training Set of class 9 has 577 samples
Validation Set of class 9 has 386 samples
```

Figure 53 Statistics Result for Training

Next, call the compile() and fit() API to start the training. Since the prediction has two or more label classes, so I set the loss function as 'sparse_categorical_crossentropy' to compute the cross entropy loss between the labels and predictions. Additionally, in the optimizer I set the function as 'adam'. This optimization can stochastic gradient descent to update network weights during training. The last step is to fit the data to train a model within 800 epochs and 100 batch sizes.

```
def dataSummaryInfo(y_train, y_val):
    print('\n\n')
    print('Total number of data for training: ' + str(len(x_data)))
    print('Training data and validation data summary:\n')
    for i in y_train:
        trainingSetData[int(i)] += 1
    for i in y_val:
        valSetData[int(i)] += 1
    for item in data:
        print('Training Set of class ' + str(item) + ' has ' + str(trainingSetData[int(item)]) + ' samples ')
        print('Validation Set of class ' + str(item) + ' has ' + str(valSetData[int(item)]) + ' samples ')
    print('\n\n')


def training():
    x_train, x_val, y_train, y_val = train_test_split(x_data, y_data, test_size=0.2, stratify=y_data)
    dataSummaryInfo(y_train, y_val)

    model = DNNNetwork()
    model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    history = model.fit(x_train, y_train, epochs=800, batch_size=100, validation_data=(x_val, y_val))

    modelVisualization(history)
    modelEvaluation(model)
```

Figure 54 Source Code of Training

After finishing the training, it needs to evaluate the model. Here I try to plot a graph for the result using the matplotlib library to get the model accuracy image (Figure 56), model loss image (Figure 57) and model result (Figure 58).

```python
def modelVisualization(history):
    plt.plot(history.history['acc'])
    plt.plot(history.history['val_acc'])
    plt.title('Model accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper left')
    plt.savefig('./model/DNN/Model_Accuracy.png')
    plt.show()

    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('Model loss')
    plt.ylabel('Loss')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper left')
    plt.savefig('./model/DNN/Model_Loss.png')
    plt.show()


def modelEvaluation(model):
    _, accuracy = model.evaluate(x_data, y_data)
    print('The model accuracy: %.2f' % (accuracy * 100))
    model.save('./model/DNN/' + filename)
```

Figure 55 Source Code of Model Evaluation

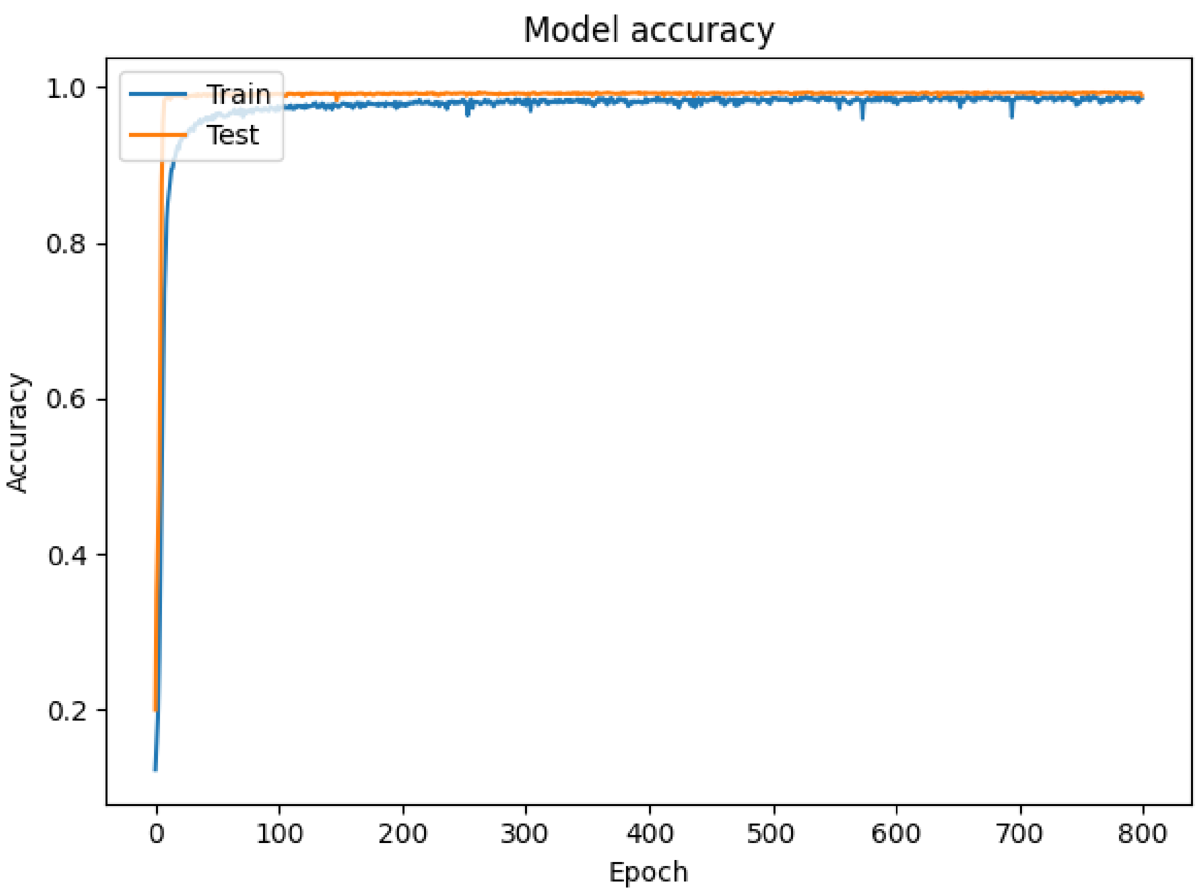


Figure 56 Model Accuracy Image

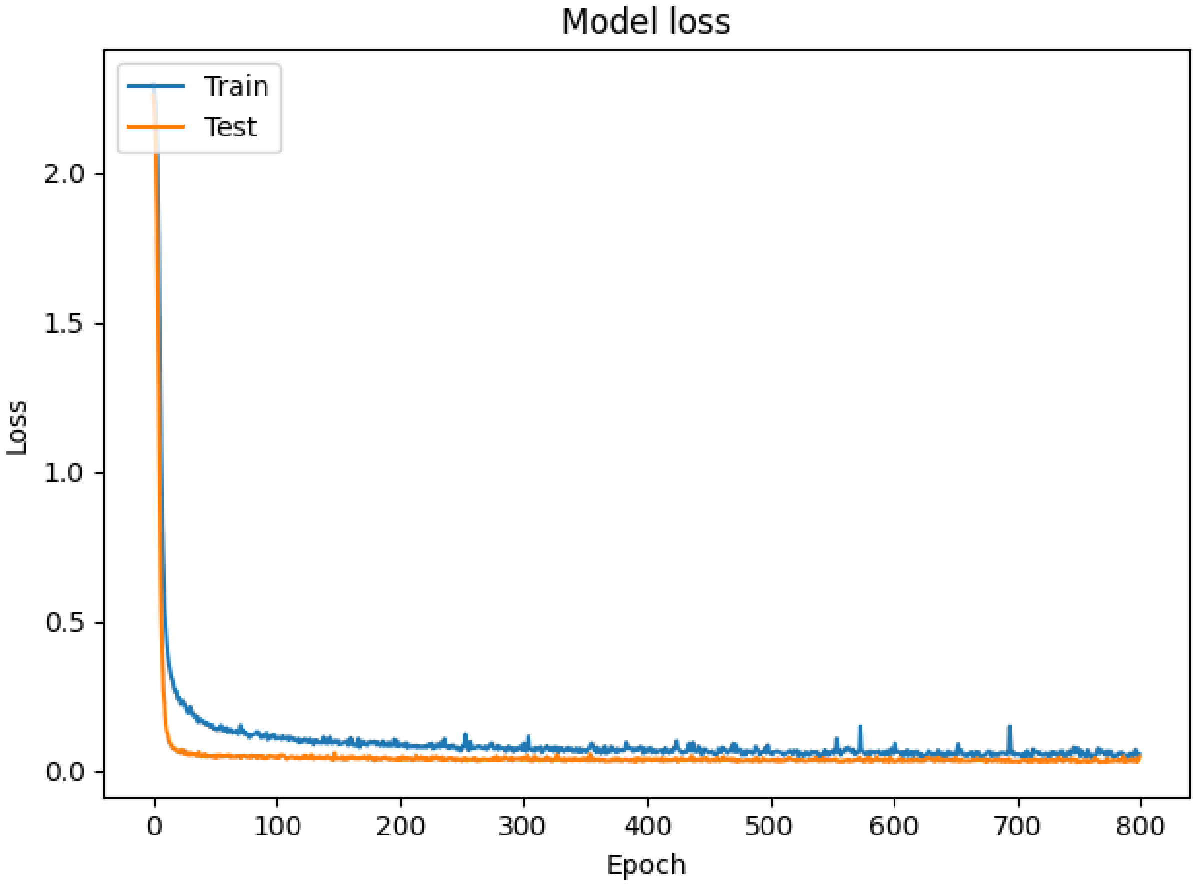


Figure 57 Model Loss Image

```
 27us/sample - loss: 0.0174 - accuracy: 0.9903
The model accuracy: 99.03
Training finished and saved the model at: ./model/DNN/gesture_model.keras

Process finished with exit code 0
```

Figure 58 Model Result

4.5. User Interface Design

Another important thing is how the UI design interacts with the user. The basic idea design is to use left and right to display two screens in parallel. On the left, it displays captured images and keeps track of your hand movements or gestures. On the right, it shows the UI layer such as buttons and text. In this left-right box design, I believe users can have a good experience using this system because left-right alignment works best when they are close to each other, their eyes do not need to move left or right too far apart. This supports the fastest completion time and fewer errors.

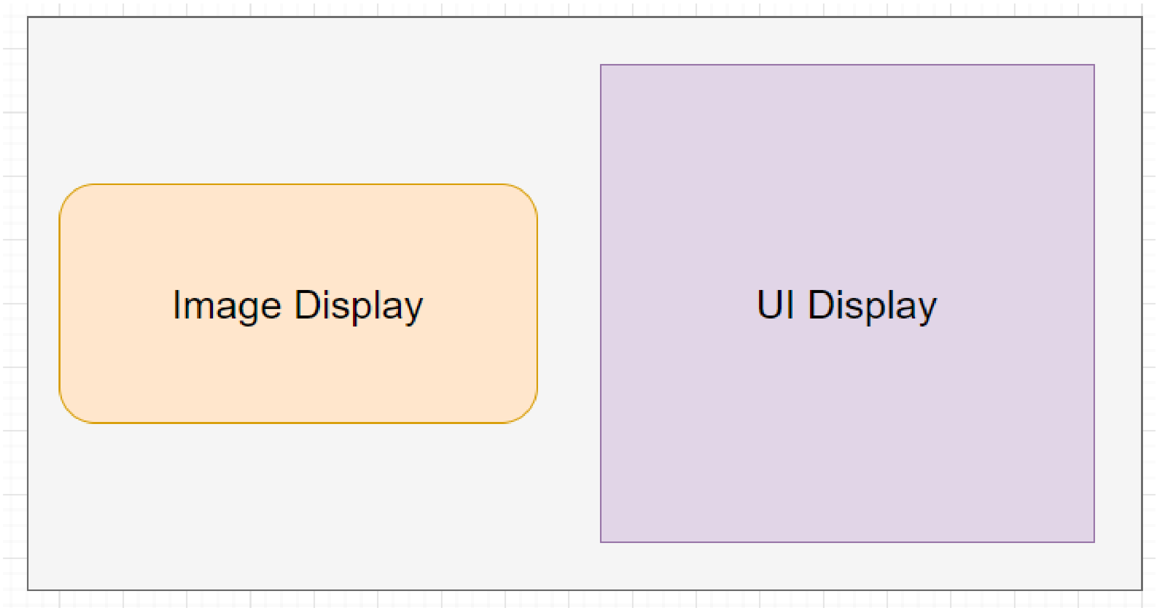


Figure 59 UI Setting

In coding, I used two Python packages called "tkwebview2" and "tkinter". First, use the TK() function to create a system window with a size of 1920x1080. Next, use the function canvas() to create a canvas, and create two objects for displaying images and UI displays. For UI display, I use load_url(), so it appears as the website. The final step is packing these two objects on the left-side canvas and right-side canvas.

```python
root = Tk()
root.title('System')
root.geometry("1800x900")

bgImg = Image.open(os.path.join(os.getcwd() + './Gesture Control/UI_img/bg.jpg'))
bg = ImageTk.PhotoImage(image = bgImg)
canvas = Canvas(root, width=1920, height=1080)
canvas.pack(fill="both", expand=True)
canvas.create_image(0, 0, image=bg,anchor="nw")

frame2 = WebView2(canvas, 500, 500)
frame2.pack(side='right', pady=80, padx=30, fill='both', expand=True)
frame2.load_url('http://localhost/project_new/index_python.php')

app = WebView2(canvas, 500, 500)
app.pack(side='left', padx=30)

label_widget = Label(app)
label_widget.pack(fill='y', side='left')
```

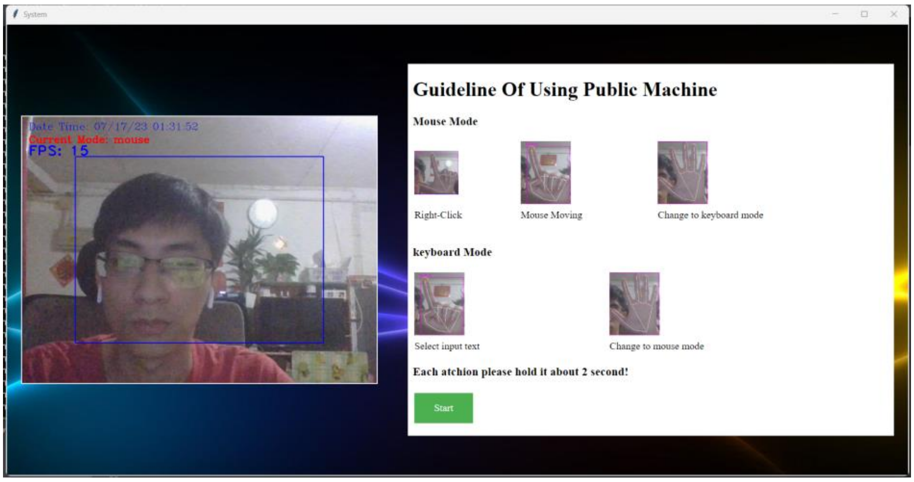
Figure 60 Coding of UI Setting

So, this is the base idea of UI design. Where the UI is displayed as a web view it can support my PHP coding and data communication with the MYSQL database. Below is each UI description:

4.5.1. Walk-in User System Flow on Public Machine

1. Index Page

By default, the machine will display guidelines and teach the user how to use this system. After clicking start it changes to the index page and prepares it for a given user. On the index page, two options are displayed for the user to choose hotel services. When the user selects the Walk-in button, the system will go to the next step.



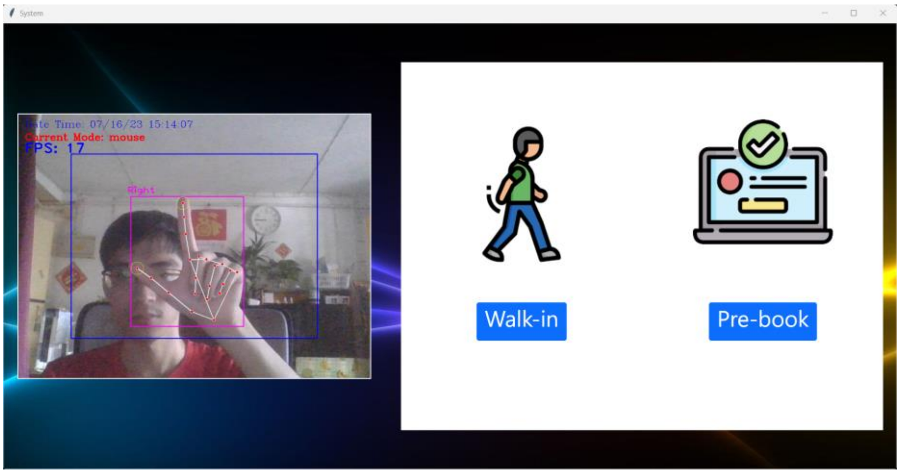


Figure 61 Walk-in UI (1)

2. Select Room Type ()

The next step is to choose the type of room you want to search for. Here I use a drop-down selection box to select. After the user selects the room type and clicks the search button, the system will enter the next step.

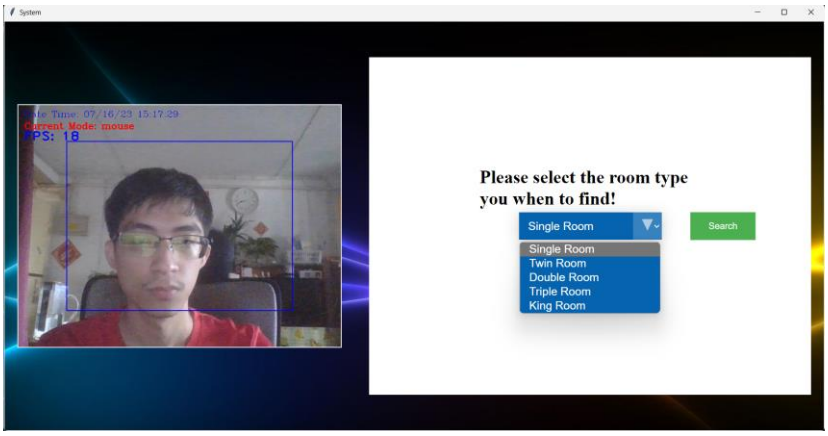


Figure 62 Walk-in UI (2)

3. Select Room for Check-in

The next step is to choose a room to stay in. This UI will display room information and what services the room provides. In addition, there are three kinds of buttons on the right side. "Previous room" and "Next room" can find different rooms. "Book this room" for users to select this room to check-in. When the user clicks the "Book this room" button, the system goes to the next step.

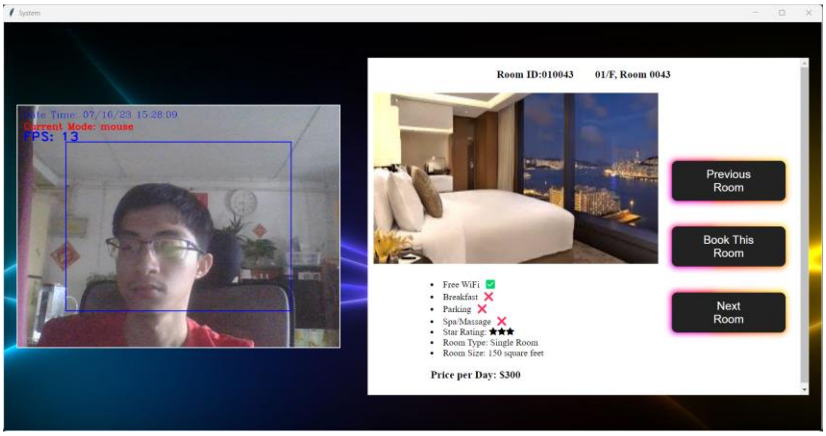


Figure 63 Walk-in UI (3)

4. Room Registration

The next step is that the user needs to enter customer information for registration. Users need to enter their name, phone number, check-in date and check-out date. After the input is complete, click the "Book" button, and the system will enter the next step.

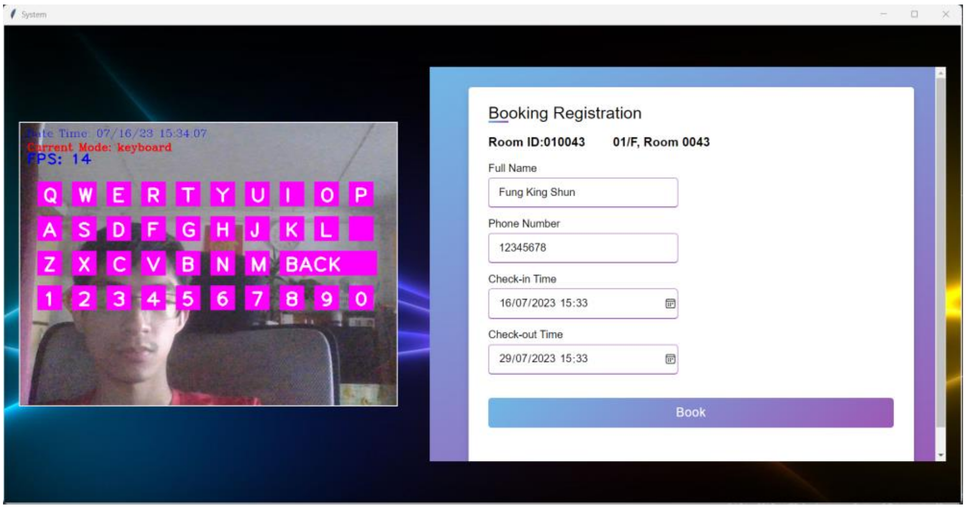


Figure 64 Walk-in UI (4)

5. Guideline of PING

In the next step, the system will display a guide page to teach users how to customize their own PING and how to unlock the room door. When the user clicks the "Start" button, the system will go to the next step.

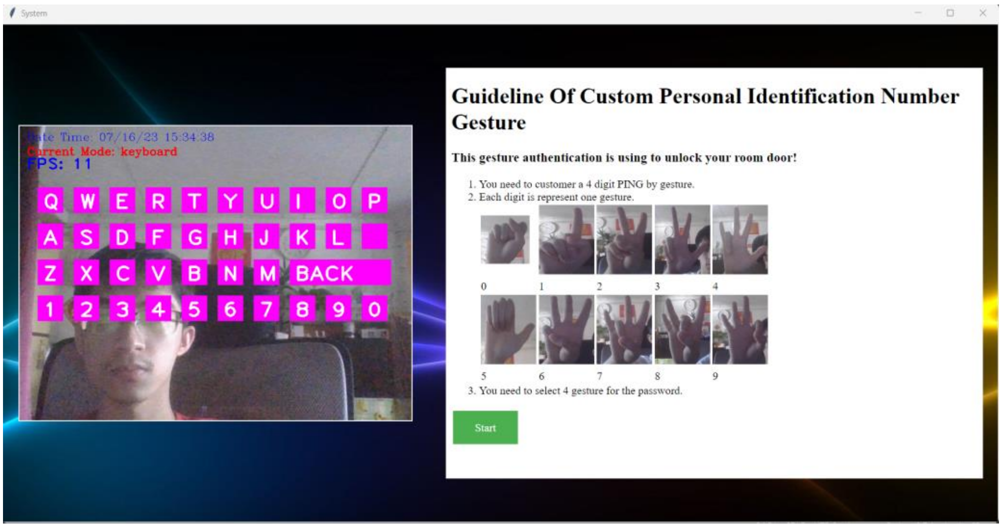


Figure 65 Walk-in UI (5)

6. Customize PING

In the next step, the system will show steps to teach users how to customize their own PING. When the user completes the customization, the system will automatically detect and proceed to the next step.

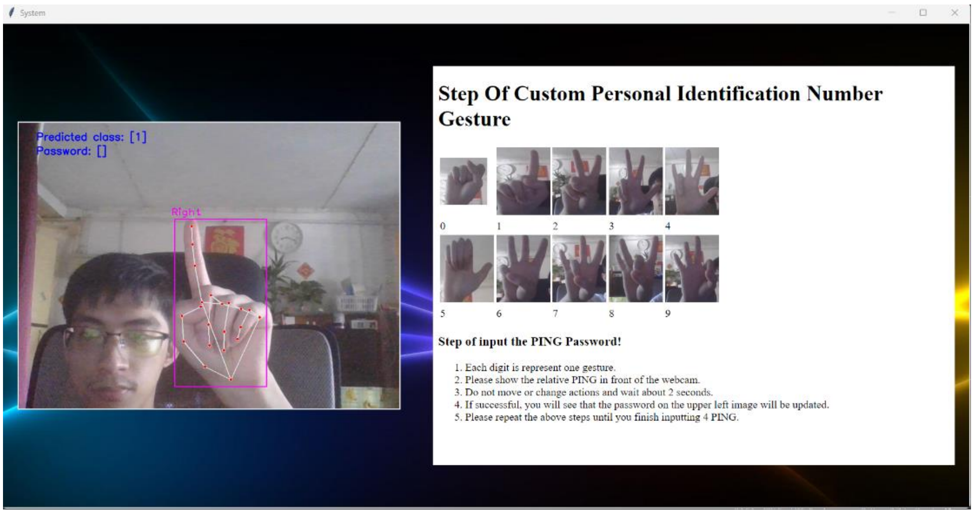


Figure 66 Walk-in UI (6)

7. Finish Check-in

In the last step, the system will display the completion page, and the user can select the "Reset" button to return to the previous step or the "Finish" button to complete the check-in.

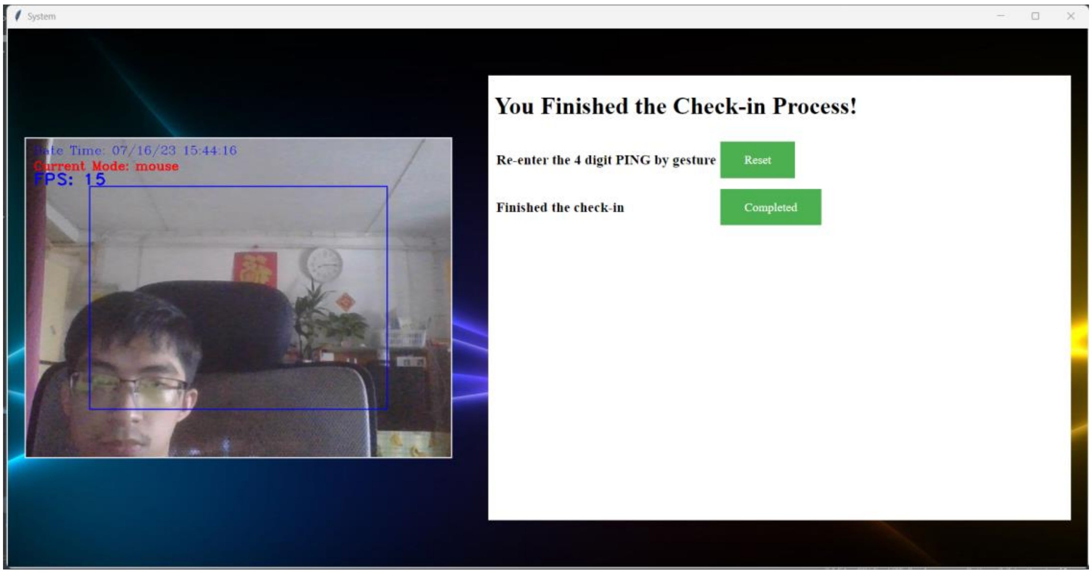


Figure 67 Walk-in UI (7)

4.5.2. Pre-order User System Flow on PHP Website

1. Hotel Website

To complete a pre-order booking, the user needs to use the website. The first is the index page to display information about the room and what services it provides. When the user selects the reserved room, he will enter the next step.

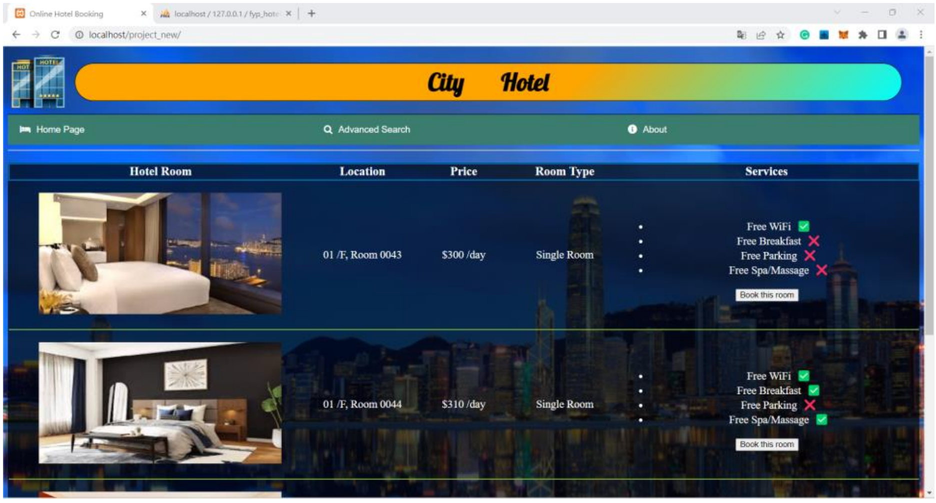


Figure 68 PHP Website UI (1)

2. Advanced Search

Sometimes, users may need to search further to find suitable rooms more quickly. So, there is an advanced search function on the navigation bar. Users can use it to quickly search.

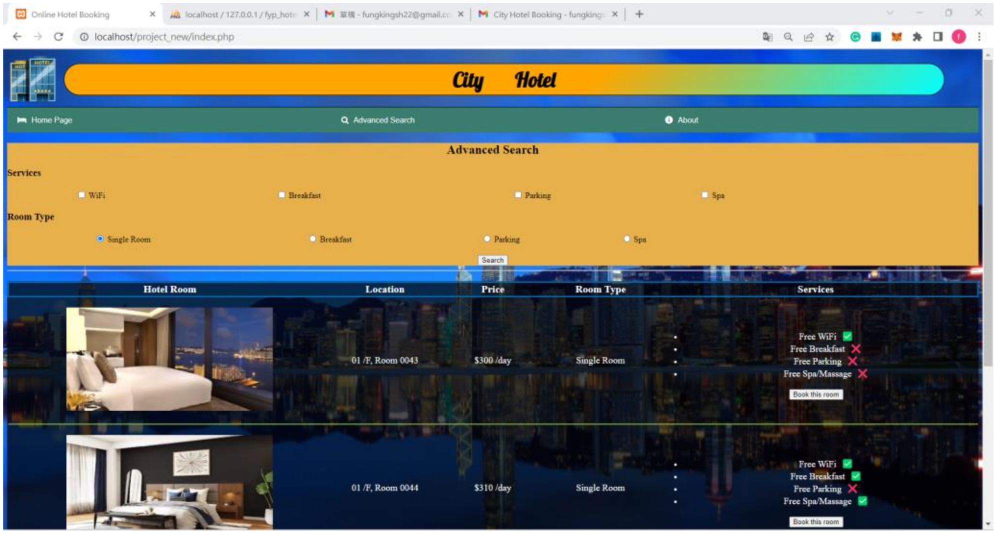


Figure 69 PHP Website UI (2)

3. Booking Registration

The next step is to make an appointment. On this page, the information on room details will be displayed on the left, and there will be a form on the right. When the user completes the form and clicks on the 'book' button, it goes to the next step.

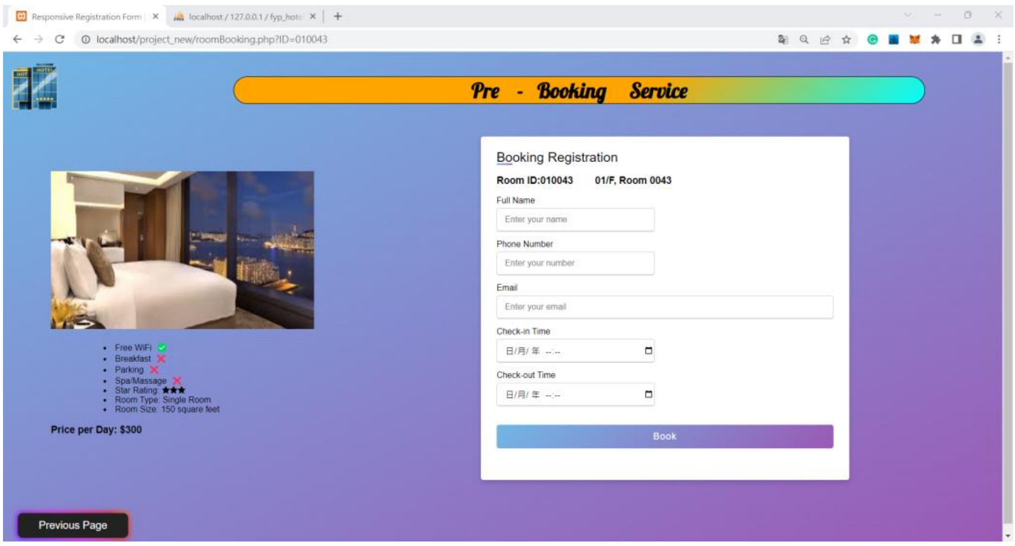


Figure 70 PHP Website UI (3)

4. QR Code

After finishing the per-order booking, the user will receive an email containing a QR code. This QR code is easy to use in the check-in process of public machines.

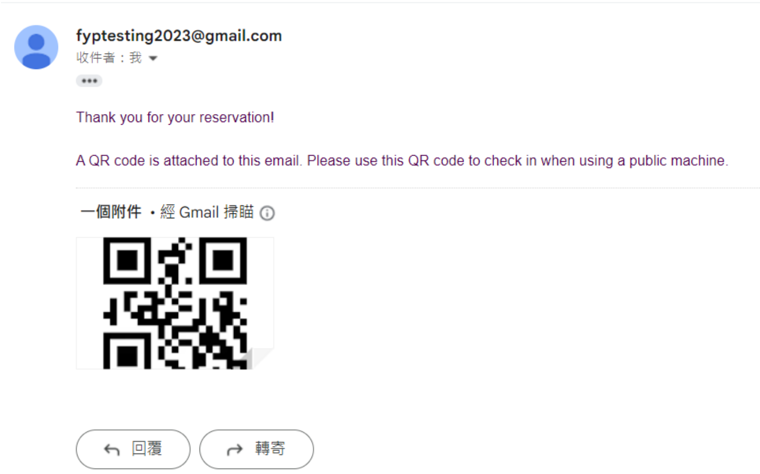


Figure 71 Email of QR Code

4.5.3. Pre-order User System Flow on Public Machine

1. Index Page

Similarly, this time the user selects the pre-order button on the index UI of the public machine.

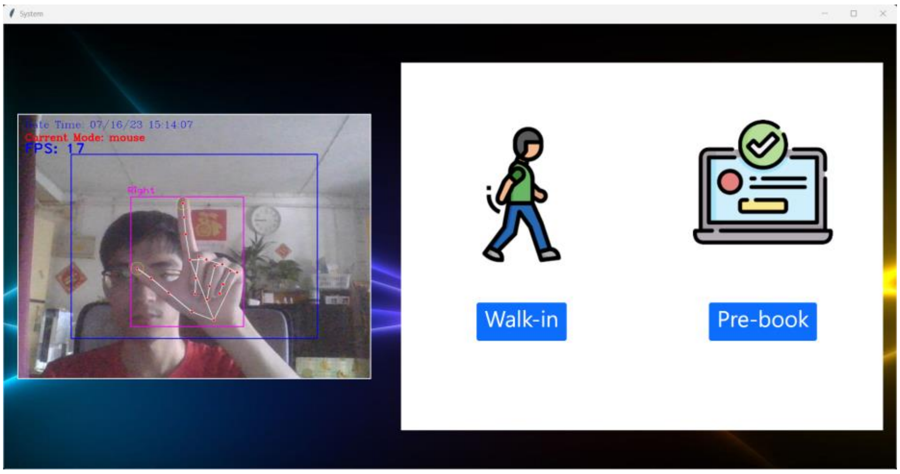


Figure 72 Pre-order UI (1)

2. Scan QR Code

This step will teach the user how to identify the reservation using the QR code obtained from the booking website. When the system detected a QR code then will find the corresponding appointment record in the database.

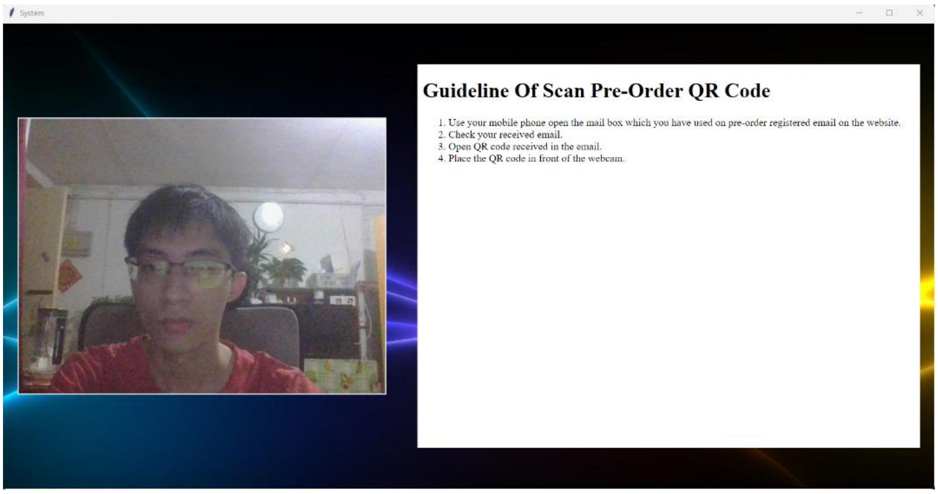


Figure 73 Pre-order UI (2)

3. Confirm Check-in

After scanning the QR code will display the booking record for the user to double-check. Users need to click the 'Confirm' button to confirm the check-in.

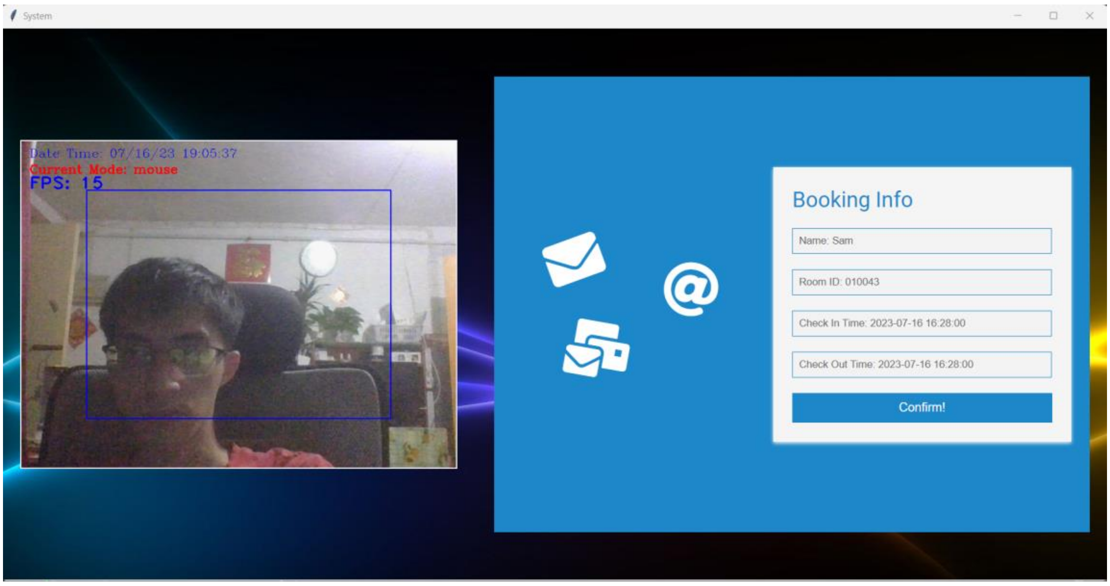


Figure 74 Pre-order UI (3)

4. Guideline of PING

In the next step, the system will display a guide page to teach users how to customize their own PING and how to unlock the room door. When the user clicks the "Start" button, the system will go to the next step.

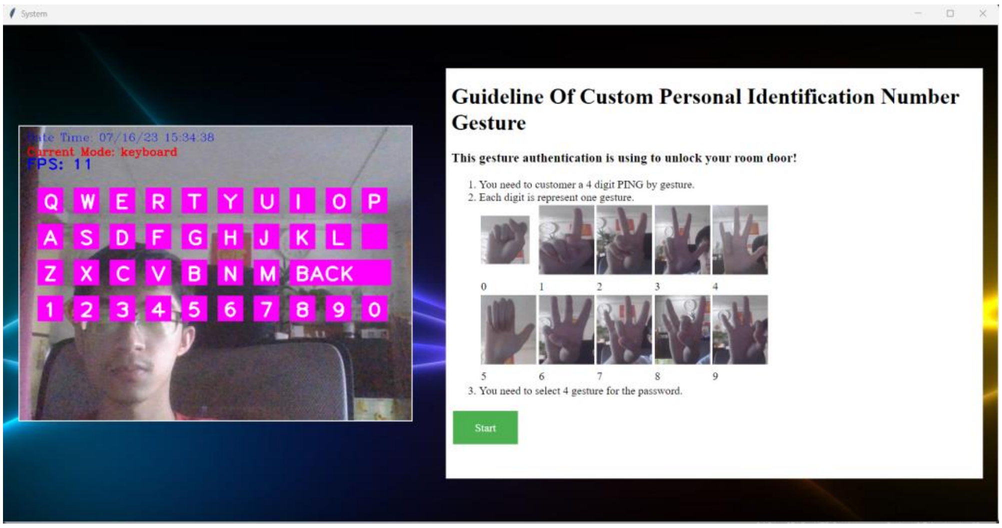


Figure 75 Pre-order UI (4)

5. Customize PING

In the next step, the system will show steps to teach users how to customize their own PING. When the user completes the customization, the system will automatically detect and proceed to the next step.

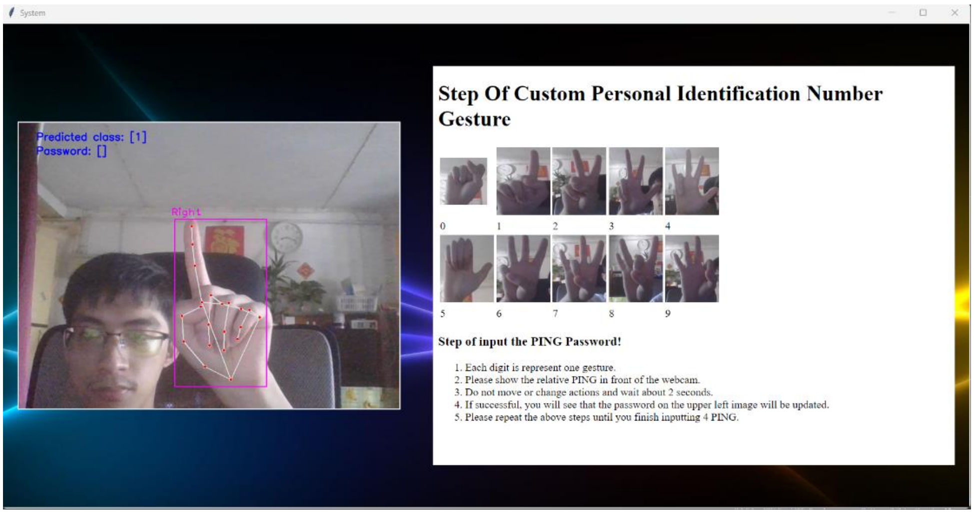


Figure 76 Pre-order UI (5)

6. Finish Check-in

In the last step, the system will display the completion page, and the user can select the "Reset" button to return to the previous step or the "Finish" button to complete the check-in.

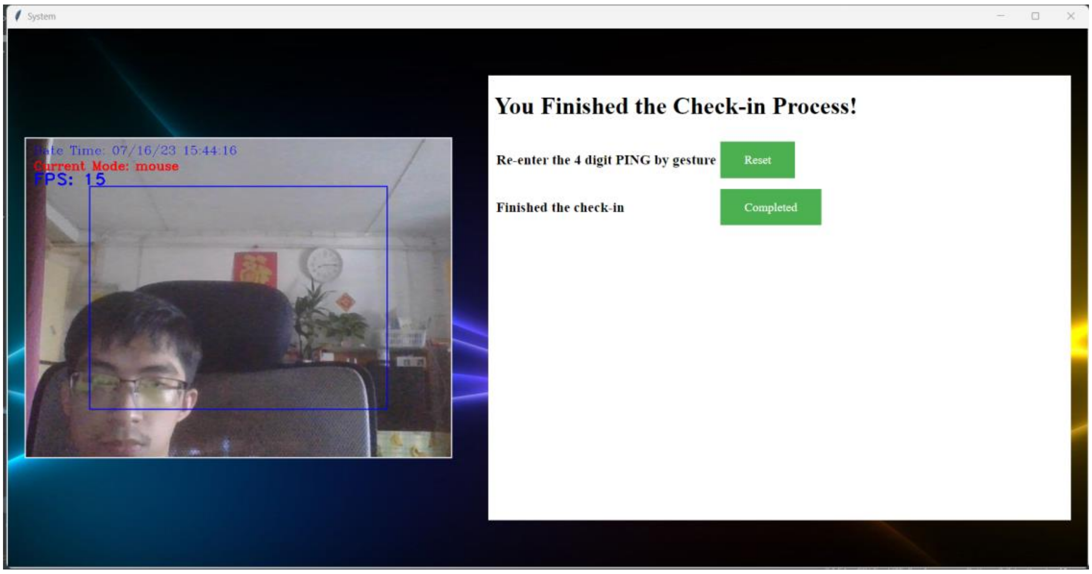


Figure 77 Pre-order UI (6)

4.6. Database Connection

Since all UI designs use the webview format. So, in PHP coding I need to do SQL update, create and query information. And below is the PHP coding on how to connect the database and do the SQL. Here I create 4 variables, and connect to the database with the server name, user name, password and database name.

```php
<?php
        $servername = "localhost";
        $username = "root";
        $password = "";
        $dbname = "fyp_hotel";

        // Create connection
        $conn = new mysqli($servername, $username, $password, $dbname);

        // Check connection
        if ($conn->connect_error) {
          die("Connection failed: " . $conn->connect_error);
        }
        echo "<script>console.log('DB Connected successfully.')</script>";
?>
```

Figure 78 Database Connection

If the database is connected successfully, then it can run the SQL query to get the data. And combined with the HTML elements like: <table>, <h1>, <form>, <td>… to do the display. Also using CSS to do the design. Finally, can get the website UI and public machine UI.

```php
if (!isset($_GET['type'])){
  $sql = "SELECT * FROM room_information, room_service where room_information.room_id = room_service.room_id;";
}else{
  $sql = "SELECT * FROM room_information, room_service where room_information.room_id = room_service.room_id and
          room_information.type = '".$_GET['type']."' and ".
          checkServices('wifi')." and ".
          checkServices('breakfast')." and ".
          checkServices('parking')." and ".
          checkServices('spa').
          ";";
}
$roomInfo = $conn->query($sql);

$conn->close();
```

Figure 79 SQL Query

5. Conclusion and Future Improvement

This project, successfully addressed the challenges faced by traditional hotels by implementing a gesture-based control system. Through the use of a virtual mouse and virtual keyboard, guests can conveniently operate public machines within the hotel premises. Additionally, I have developed an online booking website that accurately simulates the entire booking process, ensuring a seamless experience for users. To enhance security, I have incorporated a highly accurate AI model for authentication purposes, thereby establishing a reliable system for unlocking room doors. Overall, this project offers innovative solutions to improve guest convenience and security in the hotel industry.

After completing my final year project, I have gained a wealth of knowledge and experience that has greatly contributed to my personal and professional growth. Here are some key learnings from this project:

1. Technical skills: Throughout the project, I have honed my technical skills in the specific area of study. Whether it was programming, data analysis, or design, I have gained a deep understanding of the tools and techniques required to successfully execute a project in this field.

2. Project management: Managing a final-year project has taught me valuable lessons in project planning, organization, and execution. I have learned how to set realistic goals, create timelines, allocate resources effectively, and adapt to unforeseen challenges.

3. Problem-solving abilities: Undertaking a complex project has allowed me to develop strong problem-solving skills. I have learned how to identify issues, analyze them critically, and devise

creative solutions. This ability will be invaluable in my future career as it is applicable across various domains.

4. Research skills: Conducting research for my final year project has enhanced my ability to gather relevant information from various sources such as academic papers, books, online databases, and interviews with experts. This skill will be beneficial in any future academic or professional endeavors that require extensive research.

5. Time management: Completing a final year project requires efficient time management skills as it involves juggling multiple tasks simultaneously while meeting deadlines. Through this experience, I have learned how to prioritize tasks, manage my time effectively, and maintain a healthy work-life balance.

For future improvement, I suggested training one more AI model for authentication to detect left-hand action and do the left-hand detection of virtual control. An online booking website can provide more functionality such as mobile compatibility. The website should be mobile-friendly or offer a dedicated mobile app to cater to the increasing number of users who prefer booking hotels on their smartphones or tablets.

Overall, my final year project has been a transformative experience that has equipped me with valuable skills and knowledge. It has prepared me to tackle real-world challenges in my chosen field and has laid a strong foundation for my future career.

6. References

[1]: The Government of the Hong Kong Special Administrative Region. (2022). *Inbound Travel*. Retrieved October 19, 2022, from https://www.coronavirus.gov.hk/eng/inbound-travel.html

[2]: Hong Kong Tourism Board PartnerNet (2005). Research & Statistics, Retrieved October 19, 2022, from https://partnernet.hktb.com/en/research_statistics/tourism_statistics_database/index.html?pageMode=1

[3]: X. Wang, Z. Li and J. Bai, "Non-contact human-computer interaction system based on gesture recognition," Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, 2011, pp. 125-128, doi: 10.1109/EMEIT.2011.6022878.

[4]: R. Matlani, R. Dadlani, S. Dumbre, S. Mishra and A. Tewari, "Virtual Mouse using Hand Gestures," 2021 International Conference on Technological Advancements and Innovations (ICTAI), 2021, pp. 340-345, doi: 10.1109/ICTAI53825.2021.9673251.

[5]: Chang, Y.-S., Cheah, J.-H., Lim, X.-J., Morrison, A. M., & Kennell, J. S. (2022). Are unmanned smart hotels du jour or are they here forever? Experiential pathway analysis of antecedents of satisfaction and loyalty. International Journal of Hospitality Management, 104, 103249. https://doi.org/10.1016/j.ijhm.2022.103249

[6]: Global travel and tourism industry - statistics & facts, 2018, [online] Available: https://www.statista.com!topics/962/global-tourism/.

[7]: Shapoval V, Hägglund P, Pizam A, Abraham V, Carlbäck M, Nygren T, Smith RM. The COVID-19 pandemic effects on the hospitality industry using social systems theory: A multi-

country comparison. Int J Hosp Manag. 2021 Apr;94:102813. doi: 10.1016/j.ijhm.2020.102813. Epub 2020 Dec 28. PMID: 34866741; PMCID: PMC8631802.

[8]: Kim, J. J., & Han, H. (2022). Saving the hotel industry: Strategic response to the COVID-19 pandemic, hotel selection analysis, and customer retention. International Journal of Hospitality Management, 102, 103163. https://doi.org/10.1016/j.ijhm.2022.103163

[9]: Dan, R.B., & Mohod, P.S. (2014). Survey on Hand Gesture Recognition Approaches. https://cgvr.informatik.uni-bremen.de/teaching/studentprojects/nui4cars/wp-content/uploads/2013/06/survey_Agrawal_AI2012_handRecod.pdf

[10]: Salman Shaikh, Raghav Gupta, Imran Shaikh, Jay Borade (2016). Hand Gesture Recognition Using Open CV. International Journal of Advanced Research in Computer and Communication Engineering. https://www.ijarcce.com/upload/2016/march-16/IJARCCE%2090.pdf

[11]: Bhuyan, M.K., Neog, D.R., & Kar, M.K. (2012). Fingertip Detection for Hand Pose Recognition.

[12]: Truong Quang, Vinh & Tri, Nguyen. (2015). Hand gesture recognition based on depth image using kinect sensor. 34-39. 10.1109/NICS.2015.7302218.

[13]: Dey, Sumit Kumar & Anand, Shubham. (2014). Algorithm For Multi-Hand Finger Counting: An Easy Approach. https://arxiv.org/ftp/arxiv/papers/1404/1404.2742.pdf

[14]: Lin, BS., Lee, IJ., Chiang, PY. et al. A Modular Data Glove System for Finger and Hand Motion Capture Based on Inertial Sensors. J. Med. Biol. Eng. 39, 532–540 (2019). https://doi.org/10.1007/s40846-018-0434-6.

[15]: Huang, Ching-Yu. (2019). Integrated Curriculum of Multi-tier Client/Server Web-Based Database Applications. International Journal of Information and Education Technology. 9. 318-323. 10.18178/ijiet.2019.9.5.1220.

[16]: S. Hussain, R. Saxena, X. Han, J. A. Khan and H. Shin, "Hand gesture recognition using deep learning," 2017 International SoC Design Conference (ISOCC), 2017, pp. 48-49, doi: 10.1109/ISOCC.2017.8368821.

[17]: Valentin Bazarevsky and Fan Zhang, Research Engineers, Google Research. On-Device, Real-Time Hand Tracking with MediaPipe. Retrieved MONDAY, AUGUST 19, 2019, from https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html

[18]: Sambasivarao. K. Non-maximum Suppression (NMS). Retrieved Oct 1, 2019, from https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c

[19]: Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C., & Grundmann, M. (2020). MediaPipe Hands: On-device Real-time Hand Tracking. *ArXiv, abs/2006.10214*. https://www.semanticscholar.org/reader/84b19524609ad75f309be7f87bcea783e6ecd337

[20]: Kavana KM, Suma NR. RECOGNIZATION OF HAND GESTURES USING MEDIAPIPE HANDS. Retrieved 06/June-2022 from https://www.irjmets.com/uploadedfiles/paper//issue_6_june_2022/27011/final/fin_irjmets1656344520.pdf

[21]: Ajitesh Kumar. Different Types of Distance Measures in Machine Learning. Retrieved December 26, 2020 from https://vitalflux.com/different-types-of-distance-measures-in-machine-learning/

[22]: Jeremy Zhang. Dynamic Time Warping. Retrieved Feb 1, 2020 from https://towardsdatascience.com/dynamic-time-warping-3933f25fcdd

[23]: Nguyen, Phat & Ngoc, Tan. (2021). Hand Gesture Recognition Algorithm Using SVM and HOG Model for Control of Robotic System. Journal of Robotics. 2021. 1-13. 10.1155/2021/3986497.

[24]: Innovation and Technology Bureau, 2023, from https://www.smartcity.gov.hk/tc.html

[25]: Hong Kong Smart City Blueprint 2.0 from https://www.smartcity.gov.hk/modules/custom/custom_global_js_css/assets/files/HKSmartCityBlueprint(ENG)v2.pdf

[26]: Airport Authority Hong Kong. From https://www.hongkongairport.com/en/shop-dine/shopping/travelwell

Monthly Logs

June to August	Project topic feasibility study
September	Writing Project Plan: 1. Motivation & background information 2. Problem statement, project objectives & scope 3. Technical Consider 4. Non-Technical Consider 5. Major technical components 6. Description of Each Component 7. Expected results & deliverables 8. Project schedule
October	Writing Interim Reports I: 1. Introduction 2. Literature review Python Coding: 1. Webcam live streaming 2. Hand detection 3. Hand tracking
November	Writing Interim Reports I: 1. Methodology & Resources 2. Implementation 3. Testing

December	Analytical Study: 1. Database analyze 2. Web UI analyze Python Coding: 1. Identify a set of gestures actions
January	Writing Interim Reports II: 1. Database Structure 2. Gesture Actions Representation 3. Mouse Functionality 4. Keyboard Functionality
February	Coding: 1. Database Connection Testing 2. PHP coding of hotel website UI
March	Coding: 1. Gesture Actions 2. Mouse Actions
April	Coding: 1. Gesture Authentication - Dataset collection - Data training
May	Coding: 1. Gesture Authentication - Model Evaluation

	- Model Training Writing Interim Reports II: - Gesture Authentication - UI Design
June	Writing Interim Reports II: Testing Implementation
July	Writing Interim Final Report: Implementation Coding: UI Demo Video Presentation Slides