**Department of Computer Science**

**BSCCS Final Year Project 2022-2023**

**Interim Report I**

**22CS035**

**Optimizing Unmanned Hotels Using Gesture Recognition**

**(Volume 1 of 1 )**

**For Official Use Only**

|  |  |  |
| --- | --- | --- |
| Student Name | **:** | **FUNG King Shun** |
| Student No. | **:** | **56659723** |
| Program Code | **:** | **BSCEGU2** |
|  |  |  |
|  |  |  |
|  |  |  |
| Supervisor | **:** | **Dr WONG, Hau San Raymond** |
| Date | **:** | **October 23, 2022** |

# FYP Declaration

# Acknowledgements

# Abstract

**Table of Contents**

[**FYP Declaration 2**](#_Toc119941290)

[**Acknowledgements 3**](#_Toc119941291)

[**Abstract 4**](#_Toc119941292)

[**1. Introduction 6**](#_Toc119941293)

[**1.1. Background 6**](#_Toc119941294)

[**1.2. Current Development Problem 6**](#_Toc119941295)

[**1.3. Solution 7**](#_Toc119941296)

[**1.4. Project Goals 8**](#_Toc119941297)

[**1.5. Project Scope 8**](#_Toc119941298)

[**2. Literature Review 12**](#_Toc119941299)

[**2.1. Public Health Crisis 12**](#_Toc119941300)

[**2.2. Hand Gestures Based on Wired Glove Approach 12**](#_Toc119941301)

[**2.3. Deep Learning Hand Gesture Recognition 13**](#_Toc119941302)

[**3. Methodology & Resources 15**](#_Toc119941303)

[**3.1. Vision-Based Hand Gesture 15**](#_Toc119941304)

[**3.1.1. Image Preprocessing 16**](#_Toc119941305)

[**3.1.2. Tracking 18**](#_Toc119941306)

[**3.1.3. Recognition 18**](#_Toc119941307)

[**3.2. Proposed Solution 22**](#_Toc119941308)

[**3.2.1. 3D Hand Skeleton Model 22**](#_Toc119941309)

[**3.2.2. BlazePalm Detector 23**](#_Toc119941310)

[**3.2.3. Hand Landmark Model 25**](#_Toc119941311)

[**3.3. Website Server Architectures 28**](#_Toc119941312)

[**3.4. Coding Language and Environment 29**](#_Toc119941313)

[**3.4.1. Programming 29**](#_Toc119941314)

[**3.4.2. Version Control and Virtual Environment 30**](#_Toc119941315)

[**4. Implementation 32**](#_Toc119941316)

[**4.1. Live Video Streaming 32**](#_Toc119941317)

[**4.1.1. Camera Detection 32**](#_Toc119941318)

[**4.1.2. Video Frame 32**](#_Toc119941319)

[**4.2. Hand Detection 33**](#_Toc119941320)

[**4.2.1. Hand Detector Object 33**](#_Toc119941321)

[**4.2.2. Hand Tracking 33**](#_Toc119941322)

[**5. Testing 36**](#_Toc119941323)

[**6. References 39**](#_Toc119941324)

[**Monthly Logs 42**](#_Toc119941325)

# 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.

## 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.

## 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. Since 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.

Chart, histogram

Description automatically generated

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

## 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.

## 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.

Last but not least, the ultimate goal is that can reduce the risk of infection when using the unmanned hotel after implementing the gesture recognition technology application.

## Project Scope

First, two kinds of registrants used the services in unmanned hotels, namely walk-in and online booking. The system will develop corresponding operating procedures for different kinds of registrants. High-level flow chart architecture is shown in Figure 2 and **Figure** 3.

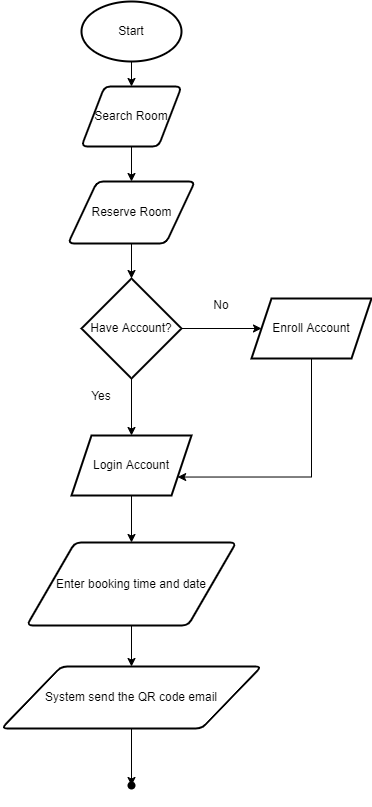


Figure 2 Simple flow chart of online booking system

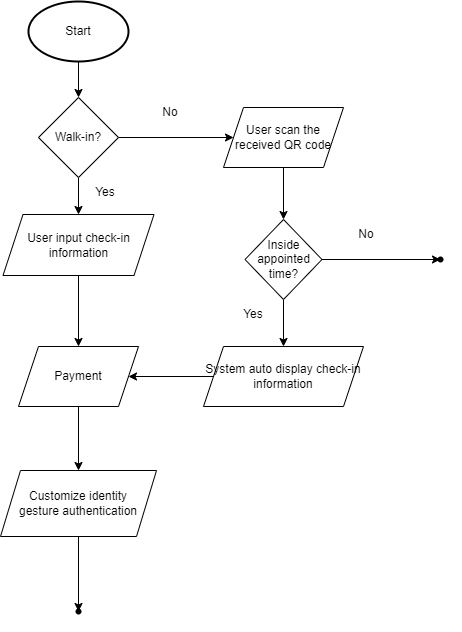


Figure 3 Simple flow chart of public machine

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 the room information and the availability of the rooms, so that users can book rooms in advance.

# 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]. Also, I will specifically explain the problem of different kinds of hand gesture technology applications and how to reform the gesture better.

## 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].

## 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 methods can collect all the data with hand position, and fingers position without loss of data or catching any noise data.

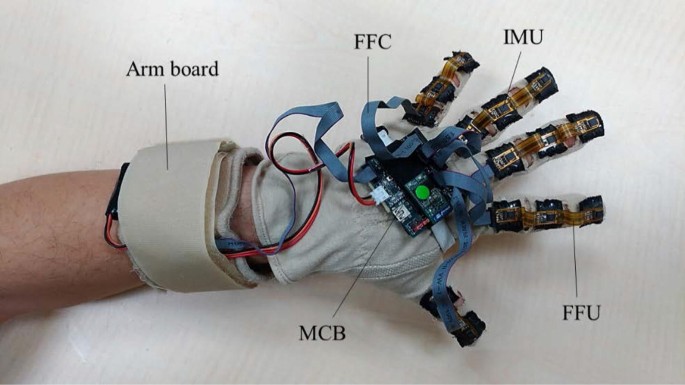


Figure 4 Modular design of the data glove

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.

## Deep Learning Hand Gesture Recognition

Deep learning recognition like Convolutional Neural Net (CNN), Long Short-Term Memory (LSTM) and Recurrent Neural Networks (RNN) is a very powerful tool. 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 as faster.

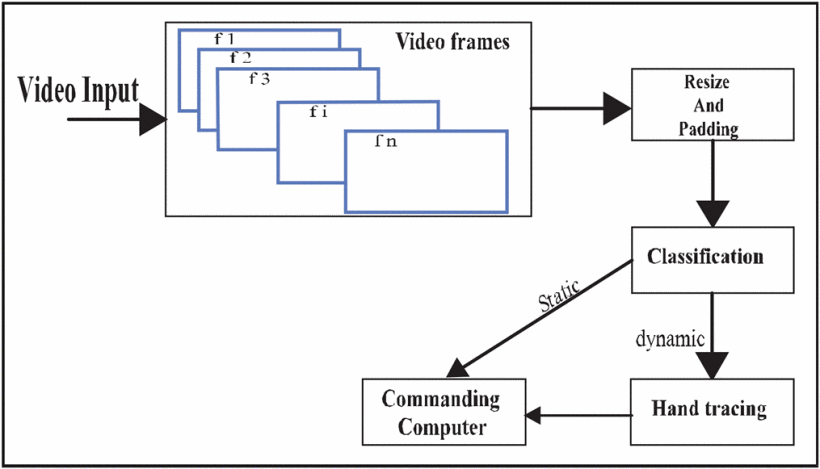


Figure 5 Workflow of CNN-based gesture recognition

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 struggle 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.

# 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.

## 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.

Diagram

Description automatically generated

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]).

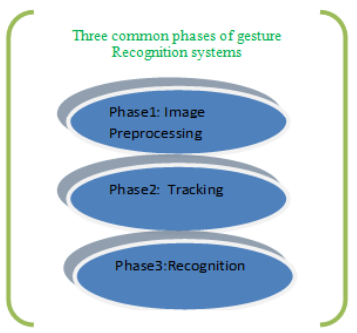


Figure 7 Three common stages of gesture recognition system

* + 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]).

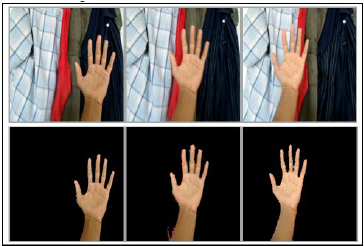


Figure 8 Color-skin segmentation of hand detection

* 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]).

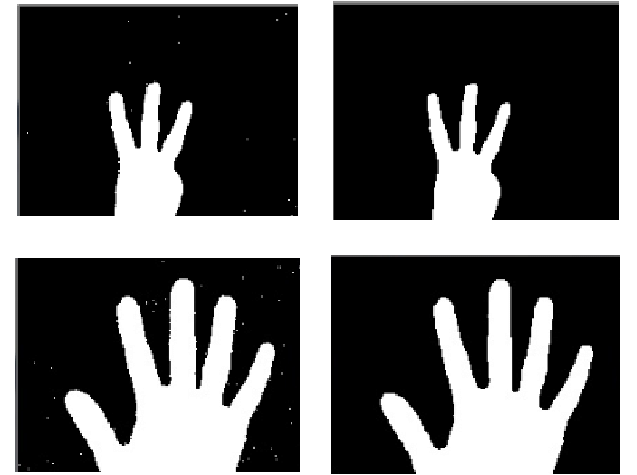


Figure 9 Sample of noise image and after noise reduction

* + 1. 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

* + 1. 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.

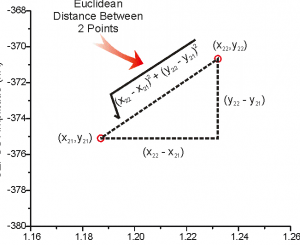


Figure Euclidean Matching Algorithm

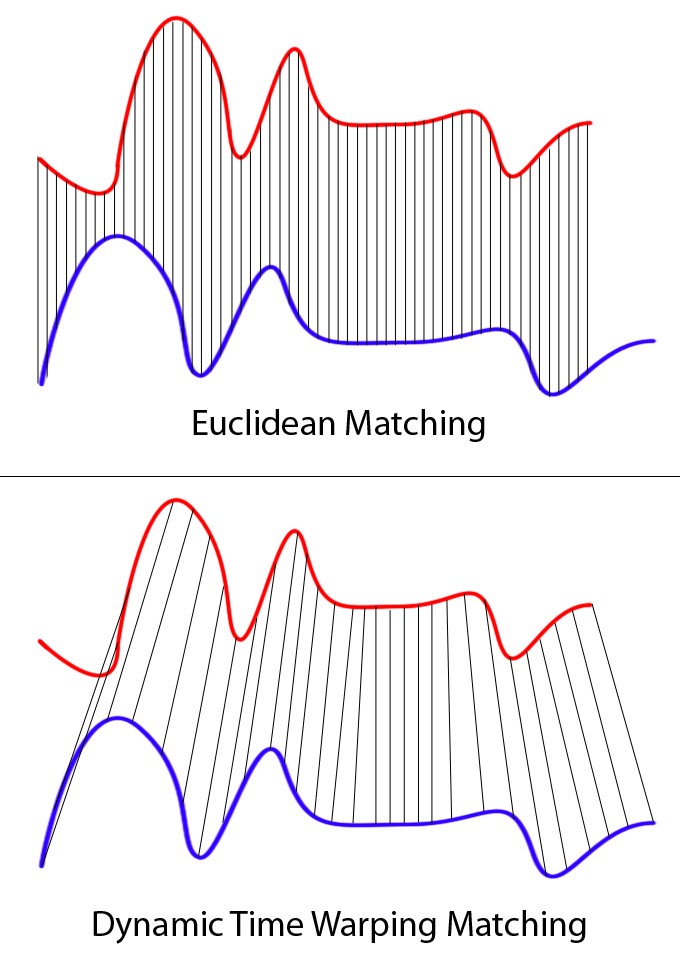


Figure Dynamic Time Warping Algorithm

* 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.

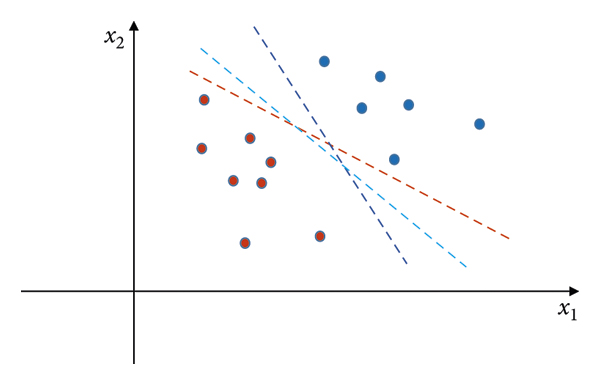


Figure Example of classifying in 2D dimension

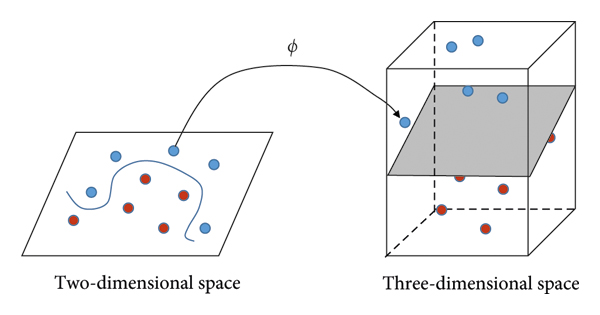


Figure Mapping data from two-dimensional space to three-dimensional space

## Proposed 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 if we 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.

* + 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]).

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure Hand perception pipeline overview

* + 1. 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]).

A picture containing text, way, road, scene

Description automatically generated

Figure Sample image before NMS and after NMS

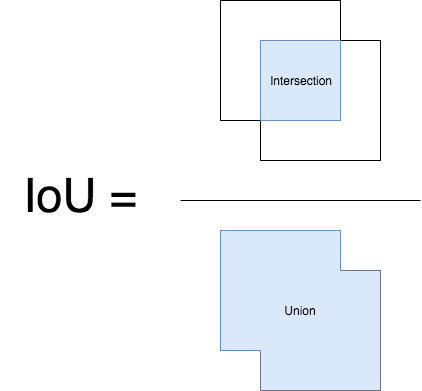


Figure Intersection over Union algorithm

Text

Description automatically generated

Figure 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.

Chart

Description automatically generated

Figure 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].

* + 1. 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]).

Text

Description automatically generated

Figure 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]).

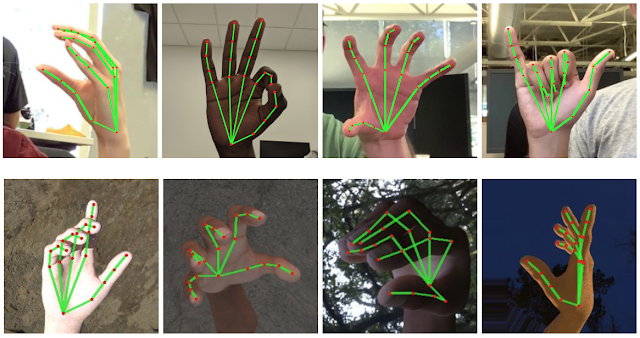


Figure Sample dataset of real-world images and synthetic hand images

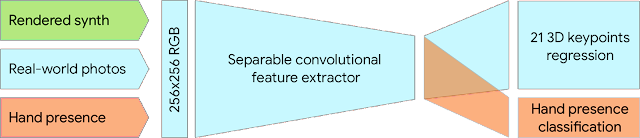


Figure A high-level model training diagram

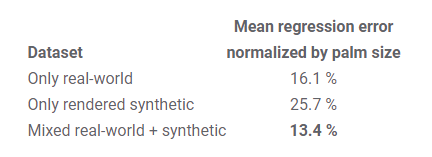


Figure Summarizes of regression accuracy

## Website Server Architectures

The following is the server architecture required by the system:

|  |  |
| --- | --- |
| 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 24** [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.

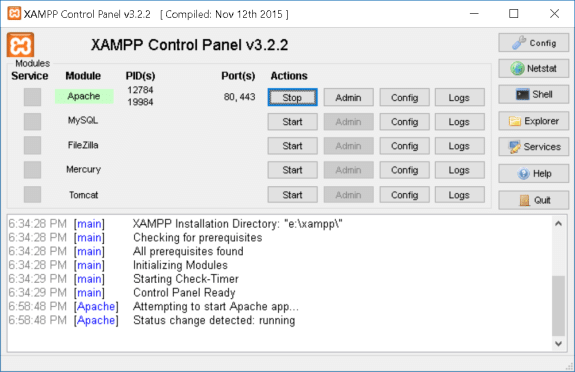


Figure 24 The main graphic user interface of XAMPP

## 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:

* + 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.

* + 1. 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

|  |  |
| --- | --- |
| Python version | Anaconda |
| Library |

# Implementation

## 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.

* + 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.

* + 1. 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).

## 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 need to use HandTrackingModule class to assist in hand detection. The following are the steps:

* + 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.

* + 1. 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 25**) 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'

}

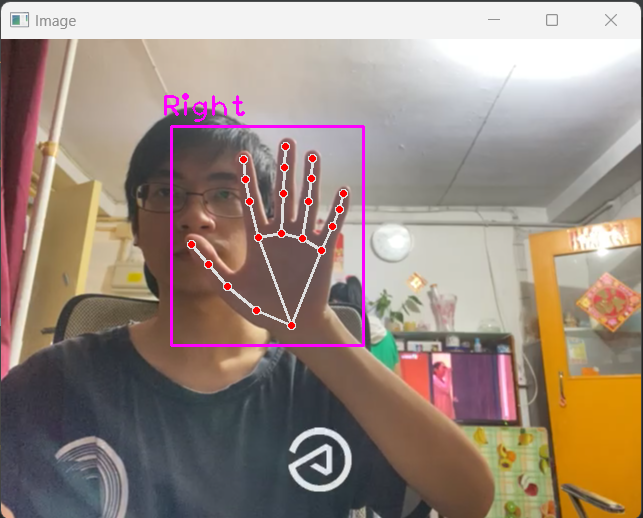


Figure 25 The 3D hand skeleton images

# Testing

A picture containing text, person, indoor

Description automatically generated

Figure 26 3D hand skeleton on right hand

A picture containing text, person, indoor

Description automatically generated

Figure 27 3D hand skeleton on left hand

A picture containing text, person, indoor

Description automatically generated

Figure 28 3D hand skeleton on different hand pose (1)

A picture containing text, person, indoor

Description automatically generated

Figure 29 3D hand skeleton on different hand pose (2)

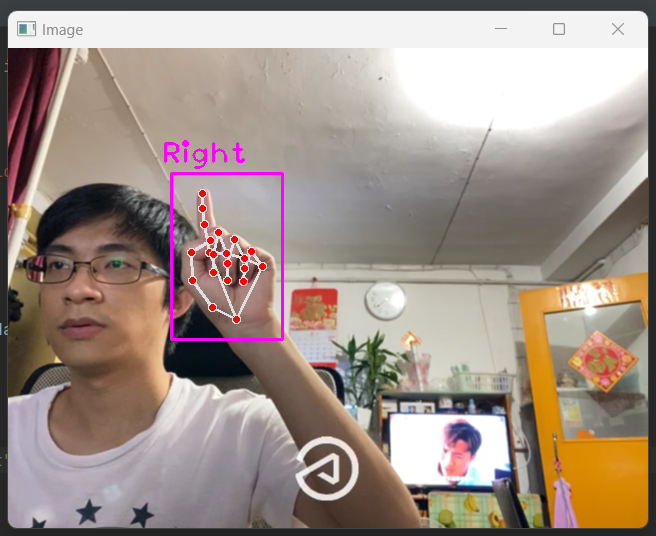


Figure 30 3D hand skeleton on different hand pose (3)

A picture containing text, person, indoor

Description automatically generated

Figure 31 3D hand skeleton on different hand pose (3)

A picture containing text, person, indoor

Description automatically generated

Figure 32 3D hand skeleton on two hand (1)

A picture containing text, person

Description automatically generated

Figure 33 3D hand skeleton on two hand (2)

# 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.

# 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 (work until to 15/11/2022) | Writing Interim Reports I:  1. Methodology & Resources  2. Implementation  3. Testing |