CONTACTLESS PALMPRINT VERIFICATION USING SIAMESE NETWORKS

BY

NG JAN HUI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2022

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: CONTACTLESS PALMPRINT VERIFICATION USING SIAMESE NETWORKS

Academic Session: JANUARY 2022

I NG JAN HUI

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

	Verified by,
(Author's signature)	(Supervisor's signature)

Address:
15, Lorong Villa Lagenda 2,
Taman Villa Lagenda,
14000, Bukit Mertajam, Pulau Pinang

Dr. Ng Hui Fuang
Supervisor's name

Date: 21st April 2022

Date: 21st April 2022

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 21st April 2022

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ***Ng Jan Hui*** (ID No: ***1802347***) has completed this final year project/ ~~dissertation~~/ ~~thesis~~* entitled "***CONTACTLESS PALMPRINT VERIFICATION USING SIAMESE NETWORKS***" under the supervision of ***Dr. Ng Hui Fuang*** (Supervisor) from the Department of ***Computer Science***, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project /~~dissertation~~/ ~~thesis~~* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

Ng Jan Hui

(*Student Name*)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**CONTACTLESS PALMPRINT VERIFICATION USING SIAMESE NETWORKS**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ______________________________

Name : Ng Jan Hui

Date : 21st April 2022

ACKNOWLEDGEMENTS

I would like to take this opportunity to convey my sincerest gratitude to my supervisor, Dr. Ng Hui Fuang for guiding me throughout the entire FYP development. His immense experience, invaluable advice, and tutelage had paved way for me to gain new knowledge whilst being able to complete this project.

I would also like to thank my family members for their tremendous support during the entire period of doing this FYP. Without their understanding and encouragements, it would be impossible for me to keep a positive attitude moving forward.

Lastly, I would also like to thank my friends and peers for a cherished time spent together, sharing, and battling the hardships that were encountered during this entire period. My appreciation goes out to all parties for their support on making this project a reality.

ABSTRACT

In our modern world, biometric based identification has become a widely adopted standard to verify the identity of a person. Biometric based identification technology can be seen taking multiple forms. From the straightforward thumbprint authentication mechanism in mobile phones up until the intricate industrial grade biometrics fusion technology.

Due to the COVID-19 pandemic, biometric based authentication systems have become increasingly in demand due to its ease of configuration and convenience of usage. Additionally, the contactless input retrieval nature of biometric based authentication systems plays a big part in greatly reducing the risks of the virus transmission. In view of the current situation, this project aims to implement a contactless palmprint recognition system as a means to authenticate users in a more hygienic way. This contactless palmprint recognition system aims to also bring a fresh perspective to the overly saturated scene of biometric authentication that are typically based on facial features, fingerprint features and other mainstream biometric features.

The outcome of the project is to deliver a system that can perform contactless palmprint recognition in four main stages via computer vision techniques and Siamese neural networks. The four main stages are – Palmprint Image Input, Region of interest segmentation, Feature extraction and Verification.

The novelties of this project are the algorithm used to segment the feature abundant region of interest from the palm image, and also the usage of a custom-built Siamese Network utilising a state-of-the-art CNN called EfficientNet as the underlying feature extractor.

All in all, this project intends to act as an alternative check in system for users that can be used in a wide variety of scenarios such as digital contact tracing, attendance tracking, registration purposes and more.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

h	Height
l	Length

LIST OF ABBREVIATIONS

No abbreviations available

Chapter 1

Introduction

In this chapter, the problem statement and motivation; project objectives; project scope; contribution; and report organization are presented. This chapter gives a glimpse on the project outline and some background information on biometrics authentication.

1.1 Problem Statement and Motivation

Typical biometric identification systems require the user to physically touch the scanner or the surface of the input device to gather the intended biometric characteristic. For example, a fingerprint reader device requires the user's finger to be in contact for at least 2-3 seconds to retrieve the fingerprint image. A typical palmprint scanner device depicted in figure 1.2.1 requires the user to place their hands flat when the palmprint is being read. Surely these devices have their limitations, for example, the sebum smudges or dirt left by the previous person will hinder the device's ability to accurately read the next user's palmprint due to the palmprint image quality being affected. Furthermore, the divider protrusions (pegs) that spread out the hand as shown in figure 1.3.1 could also pose a problem for people who suffer from arthritis as they will have trouble keeping their hands flat [1].

Figure 1.2.1 Commercial Hand Geometry Reader

Moreover, the COVID-19 pandemic also has a part to play in pushing the need for contactless palmprint recognition technology. The pandemic has undoubtedly changed the norm of life, and the hygienic awareness of people currently is higher than ever. Many users now tend to lean more towards contactless biometric identification alternatives due to hygienic considerations.

Users currently are more reluctant of placing their hands on the same device as many people had previously placed on. The remains of germs and bacteria left on the device might even pose a health threat to the other users if they do not practise strict measures such as washing their hands after using the devices. In view of the situation of the pandemic, this project is motivated to create a contactless check-in system using computer vision techniques that utilizes the palm as the main source of input. The contactless solution can reduce the spreading of germs compared to traditional physical solutions as it is more hygienic for people to use.

To finally wrap things up, contactless recognition methods is the way how check in systems will go from now onwards. Many researchers have put in long hours of research into the problem and limitations mentioned above so to bring advancements to the field of palmprint recognition. These advancements are the key in creating a check-in system that is robust and secure at the same time.

1.2 Objectives

The implementation of the project is achieved by realizing the objectives and subobjectives listed in this section. The 5 modules that form the system had each been divided into a series of objectives (milestones) to be achieved in a systematic manner throughout the development lifetime of the project. The end-goal of the project is to develop a system that can perform contactless palmprint verification using computer vision and deep-learning techniques.

The main objectives and their corresponding subobjectives are listed below:

1. To implement a contactless palmprint image retrieval method during the input retrieval process.
    a. Setup a camera device to capture high resolution palmprint images.
    b. Standardize the image capturing process to ensure the consistency of hand images.
    c. Include a backdrop to mask out the background noises of the hand images.
2. To extract the ROI from the input palm image using MediaPipe Hands.
    a. Utilise MediaPipe Hands to localise the palm image from the input camera frame.
    b. Draw the bounding box for the ROI of the palm by connecting the landmarks that are close to the palm ROI area.
    c. Extract the ROI from the palm image by cropping the region bounded by the bounding box.
3. To train a Siamese Neural Network to perform palmprint verification using Deep Learning techniques.
    a. Implement an end-to-end Neural Network training using TensorFlow.
    b. Collect suitable palmprint datasets to be used for model training.
    c. Prepare the input data by forming palmprint pairs from the dataset and labelling them accordingly.
    d. Perform image pre-processing techniques such as Histogram Equalization on the palm image to enhance the quality of features to be learned.
    e. Design a Siamese Neural Network that will learn and extract features from the input palm images, compute the L1 distance between input pairs, which will then be used to classify whether both images match.

i. To fine-tune a state-of-the-art CNN called EfficientNet for extracting the features from the palm image inputs and generate its embeddings.
ii. To implement a custom model layer that computes the L1 distance between 2 embeddings.
iii. To train a Logistic Regression classifier to classify whether both inputs match.
iv. To evaluate the model performance by computing classification metrics.

4. To develop an application that utilizes the trained Siamese Neural Network for verifying palm images.
    a. To combine the ROI extraction module with the trained model into an application that supports user palm registration and verification.
    b. To enrol and store user's palm images as feature vectors to reduce storage space
    c. To implement administrative functions to manage user profiles.
    d. To deploy application for attendance tracking / logging purposes.

1.3 Project Scope

The outcome of this project is to develop an application that can perform end-to-end palmprint recognition. The project will be divided into 5 modules and will be implemented in a phase-by-phase manner. The modules proposed are input retrieval module, palmprint ROI extraction module, feature extraction, and followed by verification using the Siamese network.

The retrieval method of palmprint images from the user is fully contactless and is assisted by the MediaPipe Hands ML library. MediaPipe Hands not only provides a robust hand tracking solution, but it also provides the ability to mark key landmarks on the hands. Using the provided landmarks, segmentation of the ROI can be done easily by extracting the region bounded by the selected landmarks. The fully contactless palmprint image retrieval method means there will not be a requirement of physically touching scanner devices or any other surfaces.

The highlights for this project will be the usage of a custom designed Siamese neural network with EfficientNet as the feature extractor to perform verification on input palm images.

1.4 Contributions

The project puts emphasis on improving the palm image retrieval process, which is commonly overlooked by previous works. This project will feature the usage of a sophisticated DSLR camera to capture the palm images, this ensures that the accuracy of recognition is not hindered due to bad image quality.

Furthermore, the novelty of this project is the usage of MediaPipe Hands to extract consistent palm ROIs from the same person. The accurate and consistent ROI extraction would be beneficial during model training as well as during live capturing of palm images for verification.

Additionally, this project shows innovation in the design and usage of a Siamese Neural Network to learn a similarity function to perform verification. The Siamese Neural Network is a deep neural network that can perform verification on palm images without requiring too much training samples as it is learning a similarity function instead of learning the features of a particular number of classes. The Siamese Network will receive 2 palm input images and output whether if the palm images match.

Lastly, this project aims to help the community by acting as a baseline study or an alternative and more hygienic way of authenticating users based on current world norms and constraints caused by the COVID-19 pandemic. This project may also be used as a basis or steppingstone for future contactless biometric solutions.

1.5 Report Organization

This report is organized into 6 chapters, namely, Chapter 1: Introduction, Chapter 2: Literature Review, Chapter 3: System Methodology and Design, Chapter 4: System Design and Implementation, Chapter 5: System Evaluation and Discussion, and lastly Chapter 6: Conclusion and Recommendation.

Chapter 1 discusses the background of the project, goals of the project to be achieved, project contributions, and achievements of the project.

Chapter 2 presents the summary of critically reviewed papers of similar nature to this project, that were done by previous researchers.

Moving on, Chapter 3 outlines the details of the design phase, various UML diagrams will be used to supplement the explanation of this chapter.

Chapter 4 will explain the implementation details of the project from start to end in depth. This includes the technology, software, and hardware needed for the project, as well as project preplanning details which include dataset collection and preparation, etc.

Chapter 5 will mainly discuss on the flow of testing and evaluating the implemented system. The system will be evaluated based on its real-time verification, and offline verification functionalities. Besides, the evaluation of the trained model will also be included. Additionally, the objectives of this project will be evaluated, and the challenges faced during the project will also be mentioned.

Chapter 6 will mainly talk about the concluding statements for this project, knowledge gained, as well as the future work to be done for this project.

Chapter 2

Literature Review

Contactless Palmprint Recognition can be subdivided into 3 distinct stages. The literature review will analyse related works and will be sectioned based on the following: 1) Hand Image acquisition and ROI extraction 2) Feature extraction based on the extracted ROI 3) Model training and recognition.

2.1 Contactless Palmprint Image Retrieval Techniques

2.1.1 Small Sample Biometric Recognition Based on Palmprint and Face Fusion

2.1.1.1 Brief

[1] proposed the usage of an external camera for the contactless retrieval of palmprint images. They used a Logitech QuickCam Pro 9000 USB webcam as the input device for capturing the palmprint images. The camera was configured with a maximal resolution of 1600 by 1200 to acquire the palmprint images. A green backdrop was also used to mask the background noise during the hand image capturing process. No external light source was involved and the palmprint images were retrieved under natural indoor lighting conditions.

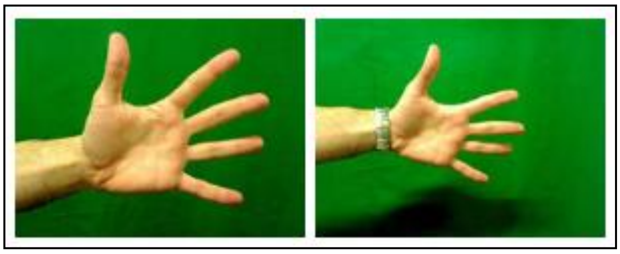


Figure 2.1.1.1.1 Examples of captured hand images with a green backdrop

2.1.1.2 Strengths

The strengths of the palmprint image retrieval method in this paper are that it is contactless and simple to set up. It is also practical enough to be used in a non-security-critical environment. The setup proposed can be used under embedded system constraints which are low-cost, low memory and low power environments. The user can simply place their hand a few centimetres away from the camera lens for this setup to capture the hand image. Furthermore, the equipment used was also easily accessible to the public.

2.1.1.3 Weaknesses/Limitations

However, the drawbacks to this setup are that the palmprint images retrieved are not well illuminated. Additional work to pre-process the images such as contrast, and brightness correction are needed to be performed to normalise all images retrieved. This is done to not hinder the performance of their algorithm. Besides, this setup is generally used for only experimentation purposes. It is quite primitive to be used in an actual security-critical application.

2.1.1.4 Recommendation

A suggestion to improve the setup is to utilise a few external light source devices. The light can be directly projected onto the hand to aid in the hand image capturing process by improving the image quality.

2.1.2 Developing a contactless palmprint authentication system by introducing a novel ROI extraction method.

2.1.2.1 Brief

[2] proposed a capturing system that features a low-cost CCD camera with a DC auto iris lens. The lens has a focal length of 28mm, a maximum aperture of f/2.8 and it also possesses a wide angle of vision. The wide angle of vision implies a larger image of the hand can be obtained. The shutter speed, focal aperture and focus adjustments have been optimally adjusted for capturing the best image quality according to the surroundings. The camera, lens, and additional LED light sources are then enclosed in a wooden box. A trapezoid shaped hole was then cut out from the top of the box to allow the lens to capture hand images that are above the hole. The system is then hooked up to a touch screen interface for controlling purposes.

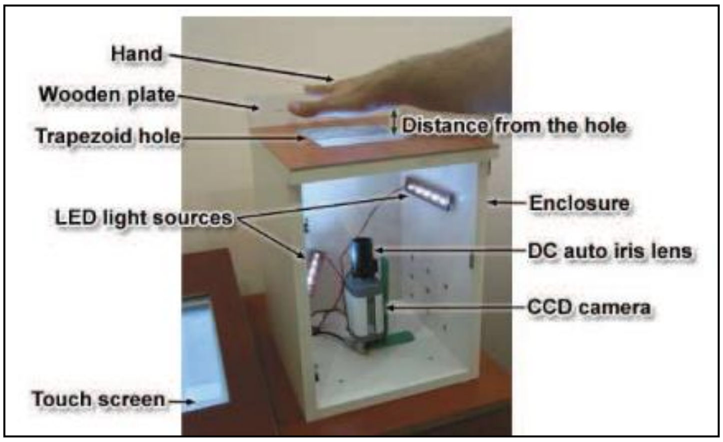


Figure 2.1.2.1.1 Capturing Setup with Enclosure

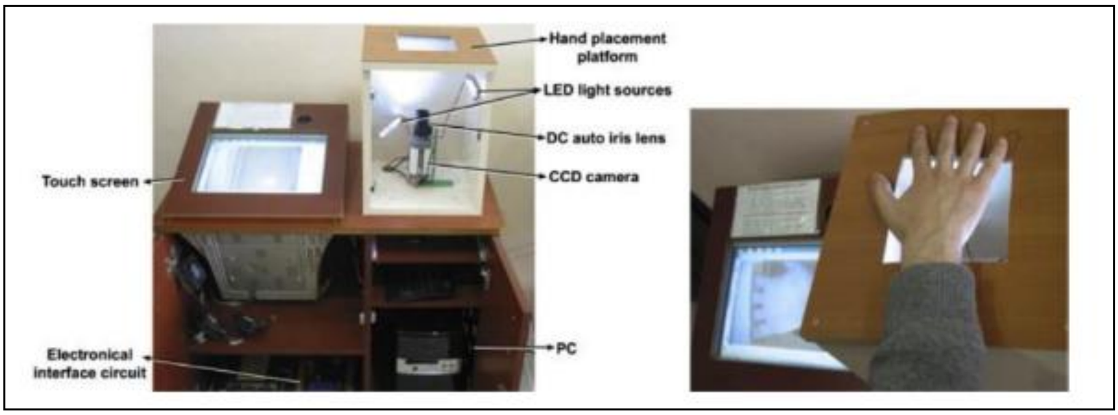


Figure 2.1.2.1.2 Usage of the setup

2.1.2.2 Strengths

The strengths of this proposed system are the usage of LED lights to increase the brightness of the capturing area and reduce the effects of surrounding light fluctuations. The usage of the lights can be a direct solution for the limitations of the palmprint capturing method in section 2.2.1. The LED lights are placed at specific locations in the enclosure. Additionally, the lights are then angled to ensure that the palmprint images are well illuminated. The intensity of the LED lights is adjusted in order to properly light up the palm instead of drowning out the subject image with intense light. More particularly, the bottom LED light was strategically placed at that specific location to curb the effects of the 3D posture variations of the hand. Furthermore, the LED light located on the upper part of the enclosure is tilted at a 45-degree angle to make the features of the palm such as the wrinkles and ridges to be more visible and clearer.

2.1.2.3 Weaknesses/Limitations

The issues of this capturing system are that the f-numbers of the lens are configured to a rather low value (Nikon | Imaging Products | DSLR Camera Basics | Aperture, 2021). Although a greater image quality of the palm and a blurry background can be captured using lower f-number values, it is still not suitable to be deployed in an actual contactless biometric system. This is because the distance between the palm and the camera lens is not regulated. The lower f-number means that if the user's palm slightly moves out of the recommended capturing distance of 20cm given by the system, then it will result in a blurry image. Besides the system is not well equipped to handle hands that will slightly shake during the capturing process. The authors proposed using a high shutter speed, however, the higher shutter speed keeps the aperture opened for a longer time and will cause the hand to be overexposed to light. The details of the features on the palm will be lost in the process.

2.1.2.4 Recommendation

A proposition to fix the issue of blurry hand images of the system is to introduce some of form physical limiter to the height where the hand can be placed. For example, a miniature ceiling can be placed no higher than 20cm on top of the trapezoid hole. Doing so restricts the maximum distance that the hand may hover above the trapezoid hole and will ensure the capture of consistent and clear palmprint images.

2.1.3 Multisampling approach applied to contactless hand biometrics

2.1.3.1 Brief

[3] proposed a contactless system that complies with the attributes of multimodal and multisampling. Multimodal in the context of this system refers to being able to extract various features from both the hand and palmprint. The recognition performance can be increased when the combination of hand and palmprint traits are used. Moving on, Multisampling refers to the usage of several images of the hand and palmprints compared to only using 1 static image in a typical setting.

The proposed system takes inspiration from swipe card readers. Two pieces of plates will be used to constraint the area which the user's hand passes through. The user will need to swipe their hands through the area between the 2 plates using a vertical motion from up to down. During the swiping motion of the hand, the palmprint images will then be collected. The capturing device being used in this system is a Logitech C600 webcam. The webcam has a resolution of 1600 by 1200 and was installed at the centre of 1 of the plates.

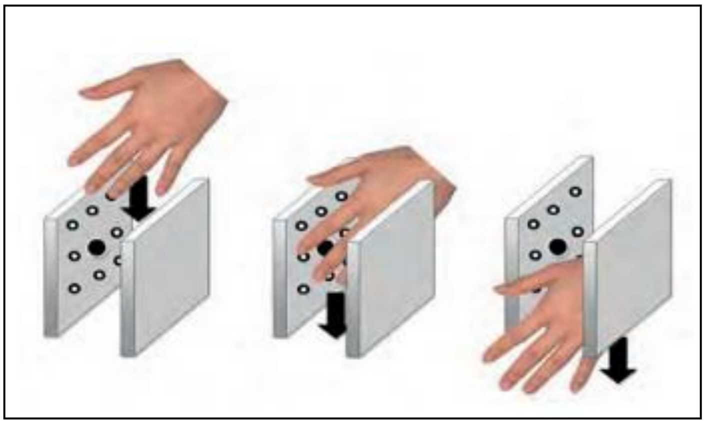


Figure 2.1.3.1.1 General depiction of the system

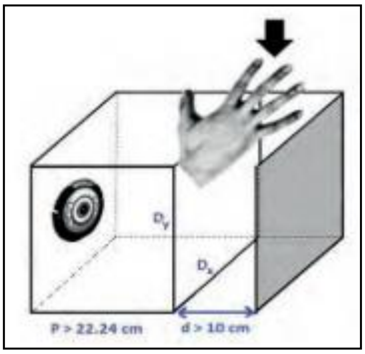


Figure 2.1.3.1.2 Prototype of the system

2.1.3.2 Strengths

The advantages of this system are that the camera was configured with low exposure values. Besides, three compact fluorescent lamps were used to regulate the illumination of the capturing area. The idea of using fluorescent lamps was quite similar to [2]. The lamps served the purpose of circumventing shadows or over lighted parts, much like the usage of LED lights proposed by [2]. The low exposure values coupled with the usage of the fluorescent lamps aids in mitigating segmentation, pose distortion and orientation issues of the hand images that had plagued other previous contactless capturing systems. Furthermore, the area size of the capturing area was calculated to be at $528cm^2$ to achieve the multisampling target of the system. With that area size, 3 hand images can be captured if they are no external disruptions.

2.1.3.3 Weaknesses/Limitations

The downside of this system is that the speed of the hand swiping motion was not regulated. The speed of the hand swiping will greatly affect the final image retrieved. People are who are not dexterous enough might swipe their hand too slowly resulting in too many images retrieved. On the other hand, some people might swipe their hands too quickly causing too few images to be retrieved. Fewer images captured will result in a loss of features, thus reducing the accuracy of the recognition algorithm.

2.1.3.4 Recommendation

A suggested solution to the weakness mentioned above is to make sure the entry and exit points are the same during the swiping motion. In other words, the same opening that the hand first enters through should be used as the exit. To achieve this, the structure of the plates should also be slightly modified such that one end is blocked. An additional plate can be used to block the other opening. Besides, an indication to pause the hands in the middle of the plates should also be given to the user so to allow the camera to focus on the hand. A brief pause for about 2-3 seconds should be sufficient for the capture device to capture the optimal number of images whilst at the same time maintaining the quality of each image captured.

2.2 Coding based hand image pre-processing, ROI extraction and feature extraction techniques

2.2.1 Multisampling approach applied to contactless hand biometrics

2.2.1.1 Segmentation and ROI Extraction

[3] proposed to use Simple Otsu's threshold to partition the hand images into fingerprint images and palmprint images. After segmentation, the ROI of the palm area is extracted by calculating the centre of the circumference that reduces the square error within the 3 valleys of the pinkie, ring, middle and index fingers, and the palm contour points roughly 200 pixels far from the exterior base of the index and pinkie finger.

The area within the calculated circumference is then extracted and is transformed to a circular shape with a diameter of 200 pixels. This is done so to counteract the problem of people placing their hands at variable lengths to the webcam during the image capturing phase.

2.2.1.2 Feature Extraction from ROI using OLOF

The authors then used Orthogonal Line Ordinal Features (OLOF) to extract the palmprint features. The authors used a combination of 2 Gaussian filters to produce an orthogonal filter. The palm features are then filtered with 3 ordinal filters that are configured to have zero average. This is done to ensure robustness against the brightness of the features.

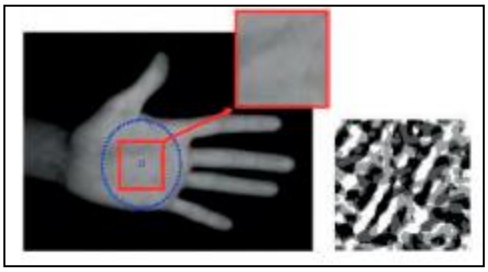


Figure 2.2.1.2.1 OLOF Extracted Features from ROI using ordinal filters of OF(0), OF(π/6) and OF(π/3)

Moving on, the results are then binarized with a threshold equal to zero. Then, the binary images obtained are then resized to 50 by 50 pixels. Finally, normalized Hamming Distance was used to compute the matching distance between the palmprint image features matrix and palmprint image feature matrix.

2.2.1.3 Drawbacks

Due to the latent intra-class differences in contactless imaging and the flaws mentioned in the pre-processing phase. The vertical and horizontal translation ordinal feature map is only used to discover the best possible matching score. The authors couldn't produce the most accurate matching score hence the matching score based on fixing the ranges of vertical, horizontal translation and rotation from -6 to 6 through empirical calculations were used instead.

2.2.1.4 Modified SIFT to curb the drawbacks

Based on the limitations mentioned in 2.2.1.3, a local descriptor technique called SIFT (Scale Invariant Feature Transform) is then introduced in this paper. Basically, SIFT can be divided into 3 major steps. Firstly, the pre-processing of the hand image is refined using Gabor Filtering. This ensures that a uniform distribution of descriptors between the main and secondary palm creases. The authors modified the Gabor Filter by tuning its parameters through the removing of average Gabor filter values to ensure robustness against brightness variation. Next, the SIFT algorithm is used to extract the features. Thirdly, pre-matching of similar features of the original method and SIFT method are then done using Euclidean distance to reduce the number of false positives.

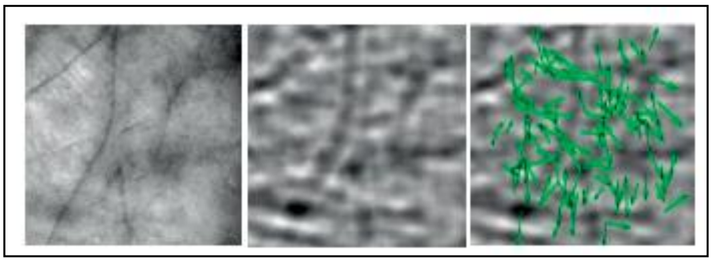


Figure 2.2.1.4.1 Key points extracted using modified SIFT from the Gabor-Filtered ROI

2.2.2 Novel Regression-Based ROI Extraction and 2 Stage Feature Extraction

2.2.2.1 ROI Extraction

According to the work of [2], they introduced a novel regression-based approach for the extraction of ROI. The strengths and novelty of this method are that instead of using pixel calculation-based approaches which can be found in other works, this approach makes use of the handshape model.

In previous iterations of palmprint recognition works, the pixel-based approach has a possibility of inaccurately choosing the hand valley points and thus leading to an inaccurate extraction of the ROI.

As stated by the authors, there are 4 main steps in this regression-based approach. The most important part which takes place during step 2 is that the authors had used an LS-SVR with RBF kernel. The reason behind doing so is that the LS-SVR is considered a non-linear regression method and can enable non-linearity for complex structures. Moreover, the LS-SVR can perform well at dealing with the small errors during landmark plotting that will affect the palm orientation.

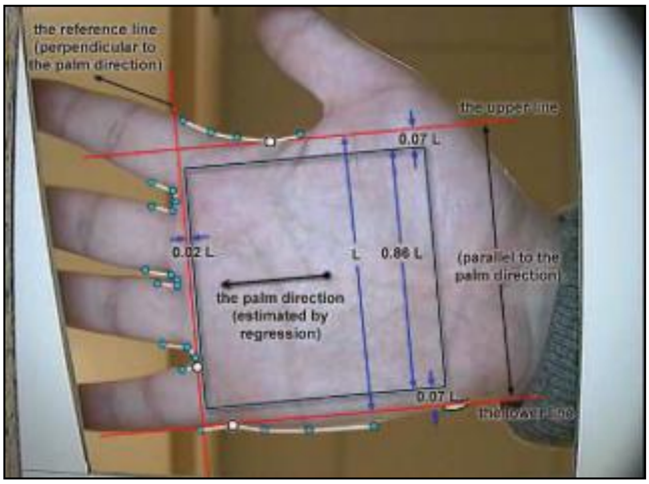


Figure 2.2.2.2.1 Overview of the ROI Extraction Technique

2.2.2.2 Feature Extraction using KFD

Similar to section 2.2.1.4, the first phase of feature extraction involves Gabor Wavelet Feature Representation. However, during the second phase of feature extraction, instead of using SIFT, the authors used KFD. KFD is used to further refine the feature set by discarding unmeaningful features which have low discriminative power.

2.2.3 Palmprint Recognition Based on Convolutional Neural Network-Alexnet

2.2.3.1 Brief

With regards to the works of [4], a convolutional neural network (CNN) was used. Particularly the CNN called AlexNet was not only utilized but was also improved in the process. AlexNet contains 8 layers of neural network, 5 convolution layers, 3 pooling layers and followed by 3 full connection layers. By default, ReLU and Dropout were used in AlexNet as the activation function to improve the speed and accuracy of the model.

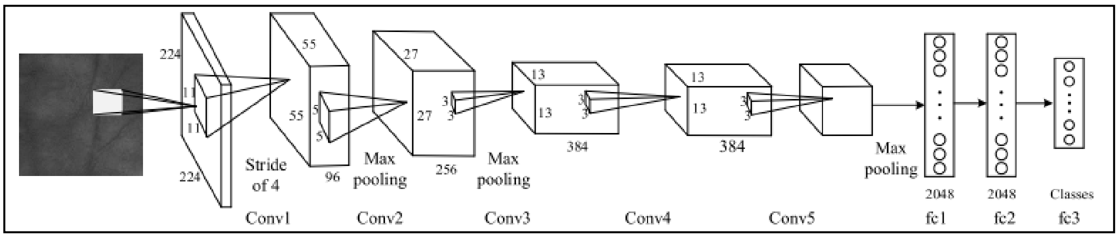


Figure 2.2.3.1.1 Depiction of the *AlexNet* Structure

2.2.3.2 Strengths

The authors pointed out that the usage of ReLU as the activation function will affect the training phase of the model. It is understood that certain neurons of the network will die and thus causing their weight to be lost and not updated. The zero weight will then be propagated forward to the next layer(s) causing the gradient to be always 0 starting from the point where the neuron dies.

To counteract the problem, the authors improved AlexNet by using a different activation called PReLU. The usage of PReLU enables the network to converge faster. Besides, the authors configured the neurons to select the best gradient in the negative region by using back propagation.

2.2.3.3 Weakness/Limitation and Recommendation

Despite all the things mentioned, there is still a weakness in using PReLU. Overfitting becomes a problem when using PReLU as the activation unit due to the additional parameter “a” it possesses. To overcome the problem of PReLU, the learning update of the “a” parameter should be achieved during back propagation.

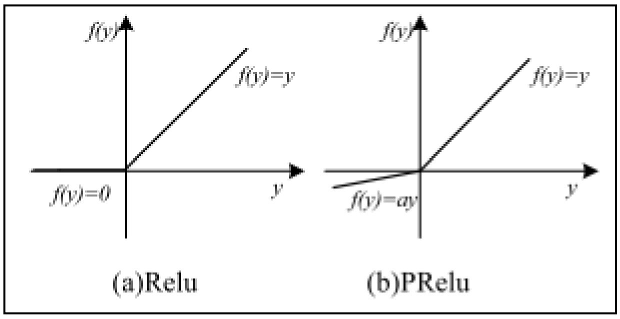


Figure 2.2.3.2 Comparison between ReLU and PReLU functions

2.2.4 Palmprint recognition system on mobile devices with double-line-single-point assistance

2.2.4.1 Brief

According to the works of [5], a novel graphic based palmprint ROI extraction technique named double-line-single-point was introduced. This methodology involved two-line segments and one point that are used to assist the user in positioning their palm correctly to ensure almost consistent and accurate palmprint ROI extraction. The authors claimed that their technique is able to reduce the computation complexity involved such as conventional segmentation and thresholding techniques applied in other similar works.

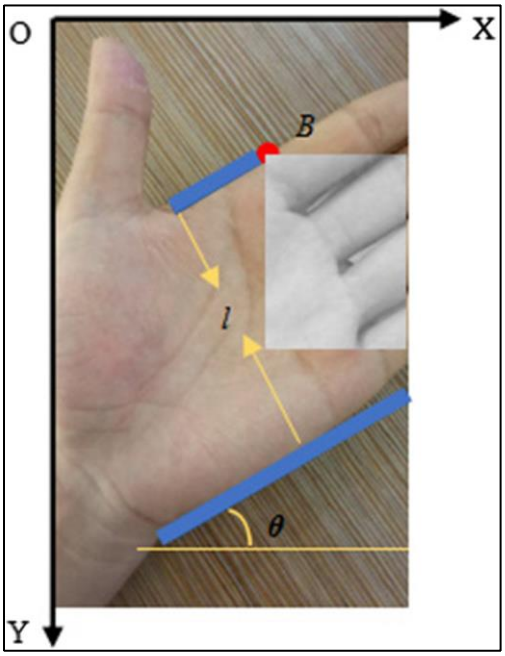


Figure 2.2.4.1.1 User-Interface and placement of the DLSP algorithm

The authors' system was configured to be mainly ran on mobile phones. The usage of a mobile platform acts alternative yet convenient check in system for the users. The users can authenticate themselves using their phone camera and palmprint should the situation requires them to do so.

Additionally, the authors ran their algorithm on palmprint databases to verify the robustness and effectiveness of the algorithm and tabulated the experimental results to act as a basis of study and comparison.

2.2.4.2 Strengths

The strengths and impact of [5]'s methodology is that the ROI can be localized and retrieved in real time by directly calculating the reference points of the region bounded by the double line single point. In other words, the ROI can be cropped from the palm directly with almost little to no image pre-processing operations involved.

Besides, the authors also proposed a judgement rule to determine the existence of the palm during the palmprint capturing process in order to prevent false enrolment of randomly segmented objects into the palmprint database. This judgement rule involves binarizing the area of the fingers above the palm ROI to retrieve the white areas.

Figure 2.2.4.2.1 Gaps between the fingers marked in white after binarization (Left); The cleaned-up gaps after performing closing operation on the white areas (Right)

This judgement rule basically makes use of Radon transformation in order to determine if the white gap above the palm fits two criteria to be a valid palm image. The results of the radon transformation are plotted in a graph as shown in figure 2.2.4.2.2.

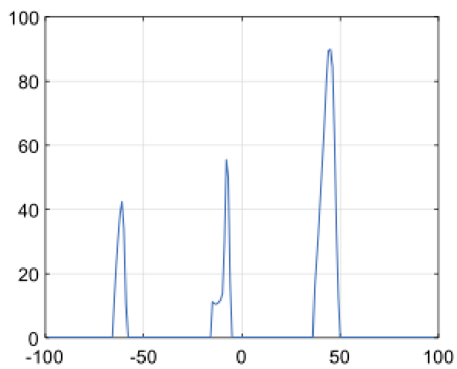


Figure 2.2.4.2.2 Shows the number of the points on the lines.

Based on the results in the graph, the two criteria to determine if a valid palmprint exists can be easily deduced. The 1st criteria checks if there are absolutely only three peaks and the distance between the peaks must be realistic enough to be the gap between the fingers. The 2nd criteria determine if the continuous peaks are increasing in height.

Overall, the usage of the judgement rule coupled with the real-time ROI extraction technique ensures low computational complexity while still capturing consistent and highly accurate ROI from the palm and still ensuring no false positives are retrieved.

2.2.4.3 Weaknesses

The weakness of their methodology lies in the illumination issues that are caused by using a mobile device and the lighting conditions of the user's current surroundings. Furthermore, despite the usage of a smart phone giving convenience to the user, it is still not mature enough to be used as an official biometric check in system to authenticate users, due to the camera quality of the majority of mobile phones are not top notch. Instead, sophisticated biometric systems require a better-quality camera that can capture the finer details and features of the biometric in question. This ensure better recognition performance because more descriptive features are captured.

Besides that, the user may need to angle and tilt their wrist at a certain angle when using the DLSP algorithm to retrieve the palmprint ROI. This may cause discomfort and soreness to the user's wrist during prolonged usage of the system.

Although the palmprint ROI retrieval method was quite revolutionary. The system's accuracy in recognising palmprints still has a lot of room for improvement.

2.2.4.4 Recommendations

A suggestion to improve their system is to address the specification of requiring the users to twist their wrists at a certain angle when retrieving the palmprint ROI. Instead of plotting the DLSP lines at a certain angle, the placement of the DLSP lines can be plotted parallel to the viewport of the device. Doing so allows the user to place their palms flat in a natural position that doesn't cause discomfort.

Additionally, a more powerful feature extraction algorithm and neural network-based classifier can be utilized to improve the poor recognition results of the author’s work.

2.2.5 Feature Extraction Methods for Palmprint Recognition: A Survey and Evaluation

[6] studied and reviewed several mainstream feature extraction methods that were commonly employed in palmprint recognition works. The authors gave their comments and critical remarks of each feature extraction method reviewed and presented their findings in their survey paper. The impact and significance of [6]’s work is that they have summarized the palmprint image types into four distinct categories and had provided an in-depth study of the feature extraction techniques of each category. Besides, the authors offered potential directions to follow as the future work for contactless palmprint recognition projects to come.

The authors mentioned that the resolution of the palm images used for feature extraction plays a big role on the number and the descriptiveness of extractable features present on the palm image. The authors pointed out that the visible features on low-resolution palm images were usually larger and texture-based, for instance, the principal lines and wrinkles are some of the bigger sized features that can be observed. Whereas on high-resolution palm images, the more intricate and miniscule features such as the minutiae points, valleys, creases, and ridge patterns can be observed. The authors then tested the feasibility and usability of the features extracted from other works by implementing popular deep-learning methods and then using the features on it.

According to the authors, the palmprint images retrieved via contactless approaches are categorized under low-resolution images. Hence, the most notable features that can be extracted from it are naturally the lines and also textures on the palm. The authors mentioned that although contactless methods allow the free and unrestricted acquisition of the palm images compared to contactless methods, but nonetheless, contactless methods are error prone and susceptible to noise due to the variations on hand pose, palm orientation, rotation, translation variances and scale (distance between camera device and the user’s palm).

The authors mentioned that due to issues such as palm misalignment and the serious intraclass differences among images acquired via contactless methods, and illumination issues, other works will often mitigate those problems by extracting additional features from the palmprint images. Besides, the authors also pointed out that subspace learning approach and collaborative representation of features are methods that can be utilized to counteract the problems mentioned above.

According to the authors, features that are highly descriptive and expressive should be considered for contactless palmprint recognition. One of the more notable examples of features extracted from the palm images is via SIFT. Next, local binary pattern or LBP is also another feature type used. Then, local line directional patterns or LLDP and OLOF are both top quality candidates to be used as features as they make full use of the rich line features on the palmprint images retrieved via contactless approaches,

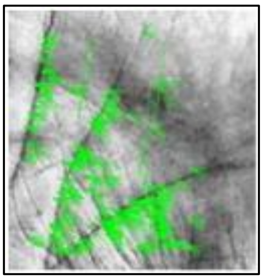


Figure 2.2.5.1 Example of SIFT Features on the ROI

The authors commented that Euclidean distance matching are often used to compute the similarities between two SIFT descriptors. The authors elaborated that LBP is considered a robust texture-based descriptor that is invariant to illumination and rotation. Basically, the authors stated that a four direction Sobel operator can be used to retrieve the LBP features on the palm, then the LBP features can be tabulated into a feature matrix by binning the histogram of LBP. The authors said line filters like Gabor filters are used to extract the lines features that will be used in LLDP and OLOF.

Finally, the authors compared and contrasted the strengths of each feature generated on publicly available contactless palmprint databases such the IITD, GPDS, and CASIA databases. Each of the databases contains images that are captured under regulated environments (controlled illumination, reduction of external noise to minimum and pose controlled) while some contain images that are captured under free conditions. The results are presented in the table below.

Table 2.2.5.2 EERs (%) obtained using different features

	IITD	GPDS	CASIA
CompCode	0.101	0.396	0.075
OrdinalCode	0.109	0.413	0.078
DOC	0.062	0.492	0.072
E-BOCV	0.089	0.510	0.064
SIFT_OLOF	0.074	0.288	0.062
DGLBP	0.132	0.340	0.071
SIFT_IRANSAC_OLOF	0.074	0.279	0.059
LLDP_MFRAT	0.031	0.294	0.030
LLDP_Gabor	0.028	0.242	0.029

Table 2.2.5.3 Error Rates (%) obtained using different features

	IITD	GPDS	CASIA
CompCode	22.21	13.97	20.73
OrdinalCode	26.68	14.47	26.74
DOC	10.01	18.92	21.49
E-BOCV	14.07	12.84	15.94
SIFT_OLOF	10.56	12.36	10.01
DGLBP	23.56	29.21	21.32
SIFT_IRANSAC_OLOF	6.72	9.83	8.54
LLDP_MFRAT	7.25	9.55	9.23
LLDP_Gabor	4.83	5.87	7.00
AlexNet	11.82	9.50	5.09
VGG-16	7.88	8.67	5.99
Inception-V3	3.78	4.69	6.15
ResNet-50	4.43	6.09	4.79

In a nutshell, the authors concluded that SIFT works generally well when dealing with feature extraction on palmprint images retrieved via contactless methods. The integration of deep learning along with contactless palmprint recognition is also a direction to explored according to the authors.

2.3 Analysis on the algorithms and classifiers used for recognition

2.3.1 Matching using Hausdorff Distance

2.3.1.1 Brief

[7] employed Hausdorff distance as the classifier in their research paper. According to their explanation, the Hausdorff distance from set A to set B is a maximin function. In simpler words, Hausdorff distance is defined as the nearest point between 2 sets. A lower Hausdorff distance implies that 2 feature sets correlate and are similar to each other.

2.3.1.2 Strengths

The strengths of using a distance-based classifier such as the Hausdorff distance is that it is a fundamental machine learning classifier. This implies that no extra model training or heavy computational overhead is required.

2.3.1.3 Weaknesses/Limitations

The weakness of using Hausdorff Distance or any distance based matching scheme (Euclidean Distance and Hamming Distance) for feature matching between 2 images is that the performance will pale in comparison with its deep learning counterparts.

2.3.1.4 Recommendations

A method to solve the issue of the weak performance of the Hausdorff Distance is to integrate the concept of distance-based matching with deep learning. With regards to the works of [8], the loss function of CNN can be replaced with a Hausdorff Distance based loss function.

$$\boxed{\mathrm{Loss}_{\mathrm{DT}}(q,p) = \frac{1}{|\,\Omega\,|}\sum_{\Omega}\Big((p-q)^2 \circ (d_p^{\alpha} + d_q^{\alpha})\Big)}$$

Figure 2.3.1.4.1 Hausdorff Distance Based Loss Function

2.3.2 Usage of optimized Residual Feature Network (RFN) for matching

2.3.2.1 Brief

Referring to the works of [9], a Residual Feature Network was used for the matching of palmprint images. This RFN was developed and carefully optimized by the authors to act as a deep learning architecture for accurately recognising the features from palmprint ROIs.

2.3.2.2 Strengths

The strengths of using the author’s self-developed RFN compared to conventional residual networks is their RFNs do not possess fully connected (FC) layers. The absence of the FC layers results in pure feature map outputs. The pure feature map outputs can preserve the spatial correspondences with the palmprint images.

Besides, the batch normalization layers of a typical residual network have been replaced with instance normalization in the RFN. This enhances the robustness of the RFN during learning low to high level features. The flaws of the pre-processing stage and image capturing processes such as hand pose or lighting issues can be mitigated with the usage of this RFN.

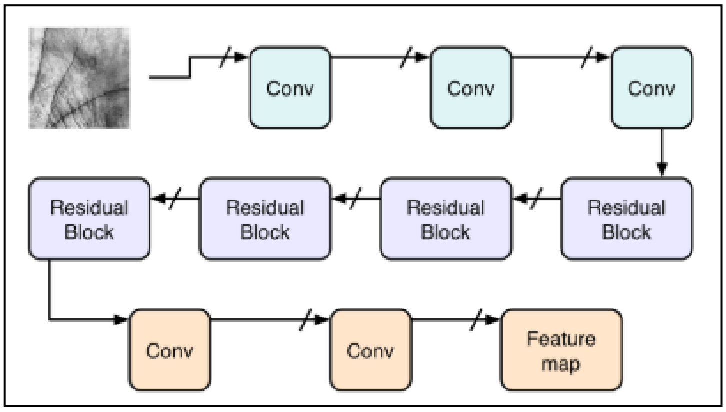


Figure 2.3.2.2.1 Overview of the RFN architecture

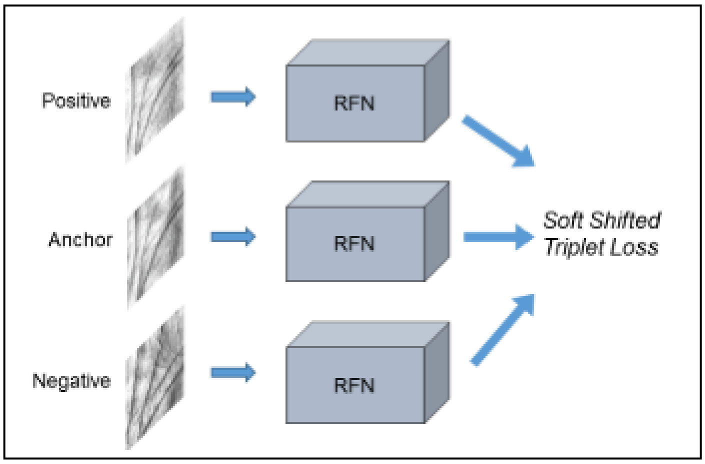


Figure 2.3.2.2.2 Triplet-based network configuration

Based on figure 2.3.2.2.2, the convolutional kernels of the RFN were trained using a triplet network. This triplet network is constituted of 3 exact RFN with uniform weights. The RFN are connected in parallel to enable the forward and backward propagation of the data and gradients for the positive, anchor and negative samples, respectively. The cost function used in the network is a triple loss function. It is used to help the network to learn in generating the feature maps that will reduce the anchor-positive distances while simultaneously increasing the anchor-negative distances. The authors softened the matching loss of each RFN pipe and called it Soft Shifted Triplet Loss (SSTL) as a mean to improve the original cost function to cope with the frequent translational changes in the palmprint images.

2.3.2.3 Weaknesses/Limitations

The issue of using this deep learning architecture is the optimization of the RFN and the tediousness of deriving the SSTL cost function.

$$SSTL = \frac{1}{N}\sum_{i=1}^{N}[\mathcal{L}(\mathcal{F}_i^a, \mathcal{F}_i^p) - \mathcal{L}(\mathcal{F}_i^a, \mathcal{F}_i^n) + m]_+$$

Figure 2.3.2.3.1 SSTL Function

According to figure 2.3.2.3.1, an additional loss function within the SSTL called the Minimum Shifted Loss (MSL) denoted as L needs to be worked out for the RFN to be differentiable along the shift directions.

Furthermore, the void generated because of the translation of the feature maps is assigned a value of 0 by default. This will hinder the training of the RFN as it requires the gradients of the SSTL between the anchor positive and anchor negative feature maps.

2.3.2.4 Recommendations

A solution for overcoming the tedious process of deriving the SSTL function is to optimise the input retrieval process instead. Ensuring the consistency of the palm images retrieved is a method to prevent the image deformation caused by pose variation. Recall that the sole purpose of the SSTL is only to cope with the trivial issue of translational changes caused by the contactless image retrieval process.

Approaches	Number of parameters	Feature extraction	Matching	Template size
CNN+Triplet loss	~449M	0.00745s	0.00140s	4096-d vector
DenseNet+SSTL	~3.1M	0.0235s	0.049s	32×32 map
DenseNet+Triplet loss	~3.1M	0.0235s	0.00040s	32×32 map
FCN+ETL	~568K	0.00142s	0.0710s	12×128 map
RFN+Triplet loss	~5.2M	0.0062s	0.00040s	32×32 map
Proposed	**~5.2M**	**0.0062s**	**0.049s**	32×32 **map**

Figure 2.3.2.4.1 Comparison between different Neural Networks and cost functions

It is proven based on figure 2.3.2.4.1 that the author's initial RFN with the original triplet loss cost function merely underperforms the optimized RFN at matching by a mere 48.6 milliseconds (almost negligible). The 48.6 milliseconds difference can still be reduced if certain measures are done to improve the image retrieval process.

2.4 Few Shot Learning techniques for Palmprint Recognition

2.4.1 N-shot Palm Vein Verification Using Siamese Networks

According to the works of [10], they had proposed the usage of a Siamese neural network to perform few-shot palm vein recognition. The dataset that acts as the basis of their study is the PolyU multi spectral palm vein database, which contains 6000 images from 500 left and right palms of 250 subjects. Their neural network performed well and was able to achieve above 90% across various classification metrics such as precision, recall, specificity, F1-Score, etc.

The highlight of their work is the utilization of a modified Siamese neural network architecture that consists of 2 sub-subnets within a subnet of the neural network for few-shot palm-vein recognition. Furthermore, the authors were able to share the palm-vein features from both palms. The concatenation of the features resulted in an increased recognition performance.

Based on the authors, the issue faced by biometric recognition tasks are the high dependence on the number of samples that are available for training the deep neural network. In other words, the neural network performs well if the number of palm vein images supplied to it increases as well. However, in the real world, suitably labelled palm vein datasets are not readily available, and as a result, the authors will have to make do with the small dataset of 6000 images by using one-shot or few-shot learning techniques.

The neural network architecture typically used in few-shot learning tasks are Siamese neural networks. The proposed network architecture of the authors is shown in figure 2.4.1.1. Contrary to the typical Siamese network architectures that consists of only 1 CNN in each subnet, the authors had proposed their own Siamese neural network by creating 2 CNN sub-subnets within each subnet of the neural network. This architecture is able to process the left and right palm images simultaneously. The CNN within the sub-subnet acts as a spatial feature extractor for the palm images, and has the architecture as depicted in figure 2.4.1.2.

The feature extractor CNN are able to generate feature embeddings of the 4 palm inputs that are received by the network. The feature embeddings generated by each sub-subnet are then concatenated into a combined 1D feature vector of length 256x1. The authors had proposed to use L1 distance as the distance function to compute the similarity between both features. To put it simply, if the value returned by the distance function returns a smaller value, this indicates that the input images are similar.

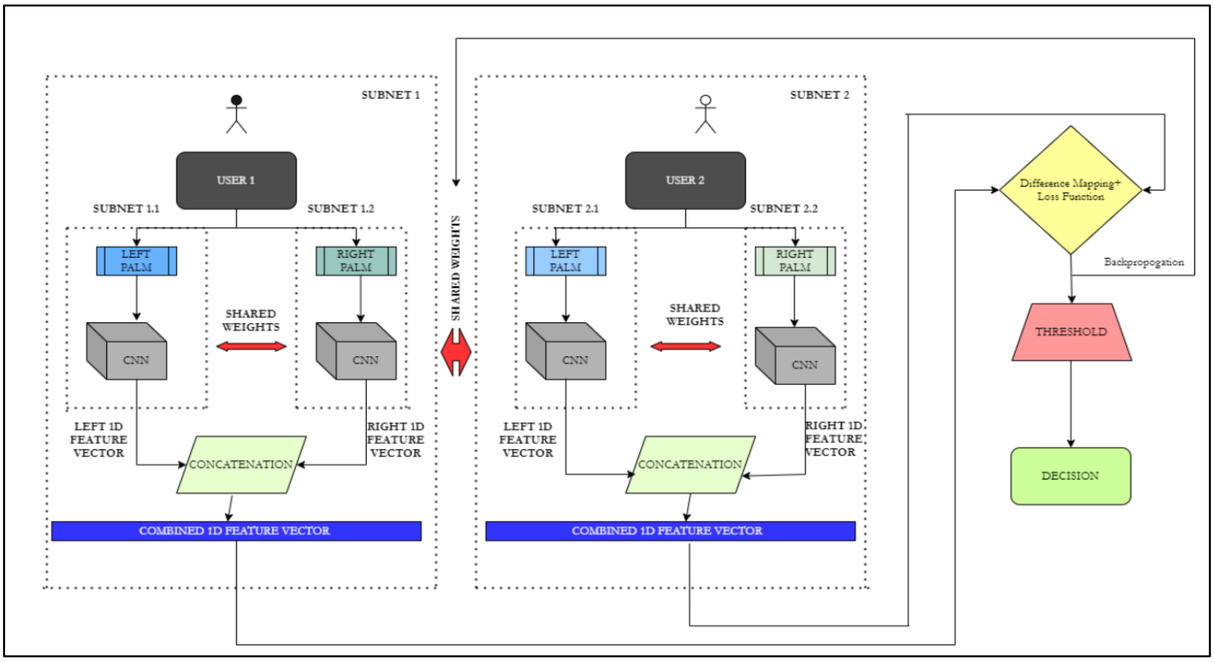


Figure 2.4.1.1 Proposed network structured based on slightly modified Siamese architecture

	Input image
Layer 1	Convolution 1, 64 x 3 x 3, Stride 1, Padding 0, ReLU
	Batch Norm+Max Pool
Layer 2	Convolution 2, 64 x 3 x 3, Stride 1, Padding 0, ReLU
	Batch Norm+Max Pool
Layer 3	Convolution 3, 64 x 3 x 3, Stride 1, Padding 1, ReLU
	Batch Norm+Max Pool
Layer 4	Convolution 4, 64 x 3 x 3, Stride 1, Padding 1, ReLU
	Batch Norm+Max Pool
Layer 5	Fully Connected, 1000 hidden units, ReLU
Layer 6	Fully Connected, 128 hidden units, Sigmoid
	Extracted Features

Figure 2.4.1.2 CNN feature extractor architecture

The distance value calculated by the function will then be used to fine-tune the network weights through back-propagation. Additionally, the authors use a sigmoid activation function to convert the distance value into a probability value. Contrastive loss and Binary cross-entropy loss were calculated and back-propagated to update the model parameters.

During implementation, the authors had split the dataset into 70% training set and 30% test set. The training set contains k x n samples, where k = 2, as the 2 classes are genuine and imposter, and n ranges from 1 to 5 depending on the number of samples. The authors had employed training batches to train their model, the model parameters were updated at the end of each epoch. During result analysis, the authors had discovered that contrastive loss performs better according to the classification metrics that they had taken. The results achieved by the authors is shown in figure 2.4.1.3.

Model	Accuracy	Recall	Precision	Specificity	F1-Score
k=2, n=2	0.862	0.867	0.874	0.881	0.871
k=2, n=3	0.881	0.885	0.892	0.899	0.889
k=2, n=4	0.892	0.897	0.906	0.911	0.903
k=2, n=5	0.905	0.911	0.919	0.922	0.915

Figure 2.4.1.3: Results acquired using contrastive loss for 2-way, with n ranging from 2 to 5 for both palms

2.5 Critical Remark of Previous Works

This section will summarize the methodology, novelties/impact/strengths, and shortcomings of all works reviewed. In addition, a short remark will be given to compare between the approach used in each work and the approach that will be used during the project development.

To recap, all total of 9 works had been analysed and reviewed. Each paper has its own form of uniqueness that sets it apart from other works such as the image acquisition method, ROI localization technique, feature extraction method and classifier used.

Table 2.5.1 Summarization of reviewed works

Author	Novelty	Strengths/Impact	Weaknesses	Remark
[2]	• Built a palm image capturing system with good lighting and sophisticated camera	• Image capturing system well lit • Reduce the negative effects of lighting issues	• F-numbers configured on the lens were too low	• A DSLR camera will be used in the project, therefore the configuration of the f-number must be considered
[6]	• Categorized palmprint image types into 4 distinct categories • Implemented deep learning methods to compare the performance of each feature extracted by many feature extraction techniques	• Offered potential directions to follow for future work		• This paper reviewed and produced an almost exhaustive list of mainstream feature extraction techniques used for palmprint images • Fusion of the feature extraction techniques in the paper can be considered during project development to produce more meaningful features
[4]	• Usage of a special CNN – AlexNet • Modified AlexNet for palmprint recognition	• Usage of PReLU activation function instead of default ReLU • Back propagation to select best gradient in negative region – allow network to converge faster	• Overfitting might occur due the extra “a” parameter of PReLU	• Take note of overfitting issue if AlexNet is to be utilized for recognition in the project

[5]	• ROI localization method without complex image processing techniques by using their line assisted method	• Consistent and accurate ROI segmentation using low computational resources	• Requires the user to twist their wrists during capturing – cause discomfort • Weak feature extraction techniques were used	• Assistive graphics will also be used in the project. The improvement of the shortcomings of the DLSP algorithm can be treated as a novelty for the project • Utilize better feature extraction techniques like SIFT to act as improvement to their work
[9]	• Usage of RFN for recognition • Introduction of Soft Shifted Triplet Loss (SSTL)	• RFN can preserve the spatial correspondences of the palmprint images due to the absence of the fully connected layer • Replacement of batch normalization layer to instance normalization – better feature learning performance	• Need extra work to figure out Minimum Shifted Loss (MSL) to ensure the RFN can be differentiable along the shift directions • Gradient between the anchor positive and anchor negative feature map will be voided because translation of the feature maps is assigned to 0 by default	• Ensure the input retrieval process can acquire higher quality palm images as the authors state that it is tedious to modify and optimize the loss functions within the RFN
[3]	• Multimodal and multisampling image acquisition approach • Built a hand image acquisition system inspired by card swipe readers • Otsu method for thresholding and plotted a custom coordinate system to localize a round shaped ROI	• Fusion of fingerprint and palmprint features – better recognition results • Low exposure value was used – reduce pose distortion & orientation issues • Capturing system well illuminated • OLOF features were used	• The speed of hand swiping motion was not regulated – ununiform number of images will be taken • Inaccurate matching score when using OLOF	• Can study the proposed OLOF feature extraction technique and use alongside SIFT in the project
[7]	• Usage of Hausdorff distance as the classifier	• Low computational complexity compared to deep learning methods	• Non-satisfactory and inconsistent recognition results	• Can incorporate a Hausdorff distance-based loss function along with AlexNet

[1]	• One of the earlier and pioneer low cost contactless palmprint image acquisition systems	• Low resource requirements – capturing system made of easy to acquire components	• Palmprint image acquired is lacking in resolution and quality	• Can utilize a backdrop in own project to mask out the background noise
[10]	• Modified Siamese Network architecture with 2 sub-subnets within a subnet	• Network able to learn left and right palm features at the same to increase model predictive performance	• Self-built model is not deep and output features are only of length 128, might affect classifier performance.	• Model architecture is built from scratch. Suggest to fine-tune pretrained CNN for better feature extraction performance.

Chapter 3

System Methodology and Design

In this chapter, the overall process and design phases of the project will be outlined. Section 3.1 will provide a top-level view of implementing the core function of the system, which is to verify palms. The further subsections will focus on the design of the final application named VeriPalm, that integrates the trained model to perform verification. UML diagrams and other relevant diagrams will be used to assist the explanation of this chapter.

3.1 System Design Diagram

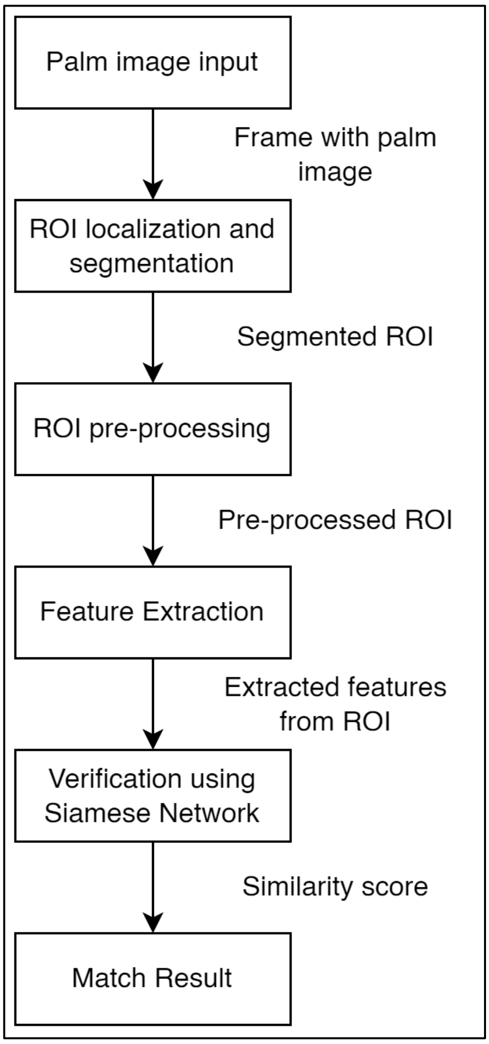


Figure 3.1.1 High Level View of System

Figure 3.1.1 above depicts the high-level overview of the palmprint verification system.

The input palm image captured from the camera device undergoes 4 phases before the verification results is outputted. During the ROI localization and segmentation stage, the video frame containing the palm will be passed to the various ML models of MediaPipe Hands. MediaPipe Hands will then track and localise the palm on the video frame and then plot the key landmarks located on the palm. Using the landmarks information obtained, the ROI of the palm can be easily segmented by connecting points 1, 5, 17 and an extrapolated point perpendicular to 5 and 17 as shown in the diagram below.

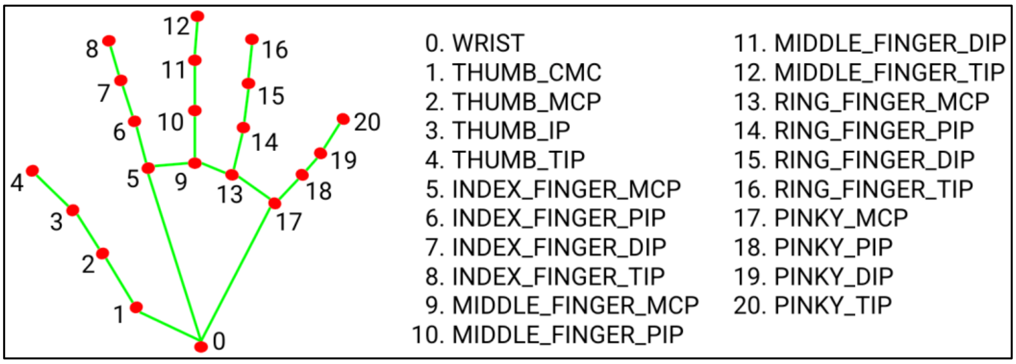


Figure 3.1.2 MediaPipe Hands 21 Landmarks

Then, the extracted ROI of the palm is subjected to image pre-processing techniques to enhance the image quality. Better image quality allows more descriptive features to be learned and extracted by the CNN during the verification phase. The image pre-processing techniques include Histogram equalization, normalization, etc.

Moving on, the pre-processed ROI will then be passed into a state-of-the-art convolutional neural network called EfficientNet for feature extraction. The extracted feature vector (referred to as embedding) can then be used for verification or for storage purposes.

During the verification stage, 2 embeddings extracted from 2 palm ROIs are then inputted to the Siamese neural network. The Siamese neural network will then compute the L1 distance between the embeddings and output a similarity score. A higher similarity score indicates both palms match, whereas a lower similarity score indicates both palms do not match.

Finally, the match results can be obtained by subjecting the similarity score to a sigmoid function to obtain a matching probability. The 2 palms can then be verified by using this probability along with a threshold value.

3.1.1 System Architecture Diagram

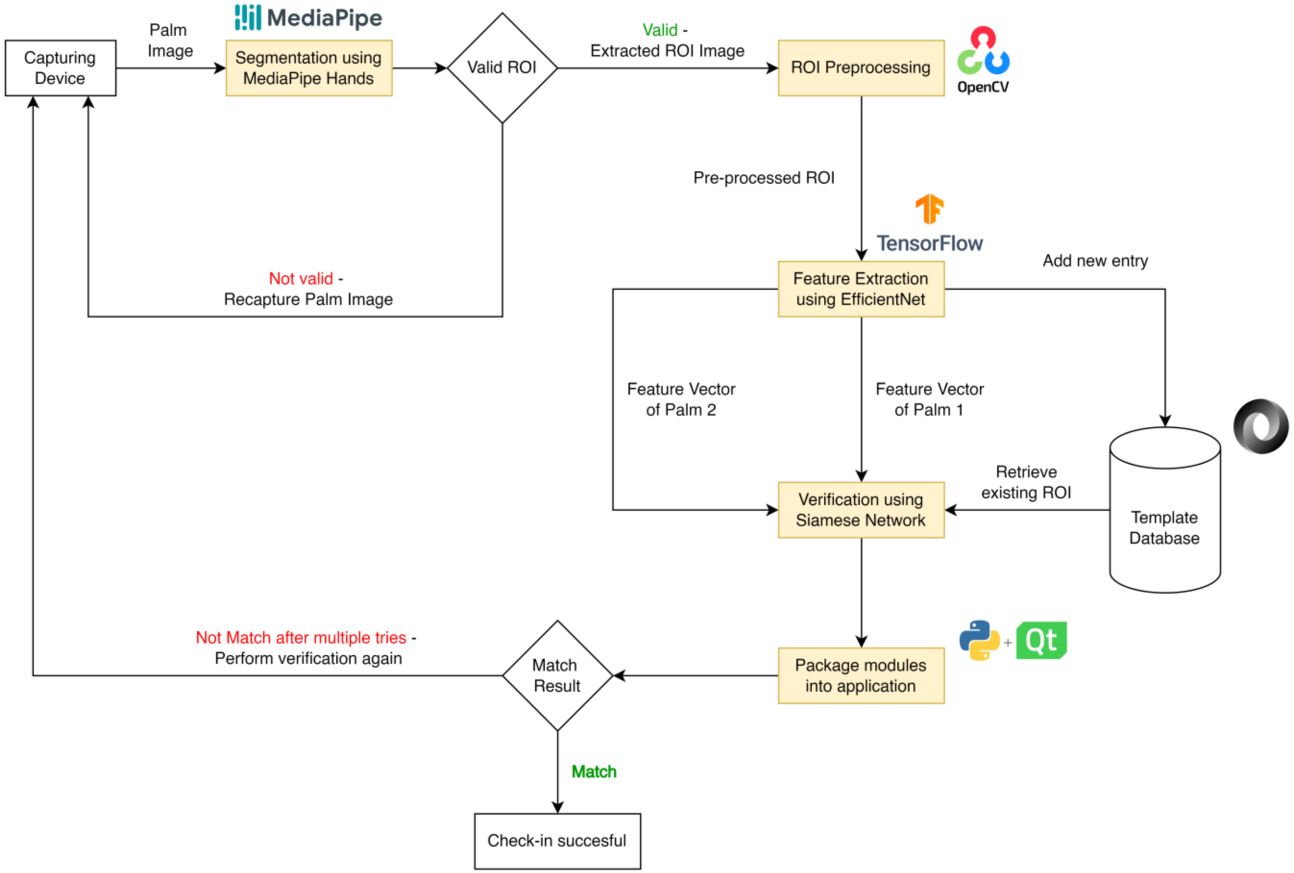


Figure 3.1.1.1 Architecture Diagram for VeriPalm System

The structure and logical flow of the proposed system is depicted in the block diagram below. The 4 most important core modules that serves as the backbone of the system is highlighted in yellow. In this section, each block within the block diagram will be discussed in detail.

The hand images will be taken with a DSLR camera for better image quality. The usage of the DSLR Camera is an improvement to the image retrieval process of other related works because most of them usually only employ the usage of webcams that have relatively low resolution. A backdrop will be used during the capturing process to eliminate external noise in the background and ease the image pre-processing stage later in the pipeline. OpenCV will be used to interface the system with the camera.

The ROI will be extracted from the palmprint via MediaPipe Hands by utilizing the landmark information obtained. Then, image pre-processing techniques such as image size resizing and normalization, noise removal (denoise) via Histogram Equalization, will all be performed on the ROI as a means to clean it up to make sure it is at its most ideal condition for feature extraction. The output for the ROI Preprocessing block would be a cleaned and prepped ROI that is to be used for feature extraction.

Moving on, the ROI is then passed into the feature extraction module. Feature extraction will do using a deep learning approach using the EfficientNet CNN. The goal after this stage of the module is to represent the distinctive features of the palm ROI in a 1-dimensional feature vector. After the feature extraction stage, the features can then be stored into the template database along with the user's other identifiers and information for future matching. Apart from that, the feature vector could also be passed into a custom-built Siamese Network for verification.

During the verification stage of the system, two feature vectors extracted from two ROIs will be classified to determine its similarity. Two possible outcomes can be presented after the verification phase. The first outcome is if the feature vector from the user's ROI doesn't match any feature vector in the template database. The user would then be prompted to retry the verification process.

The second and correct outcome is the feature vector of the user's ROI matches one of the feature vectors of another ROI in the template database. The system will then authenticate the user and grants the user a successful check in.

3.1.2 Use Case Diagram

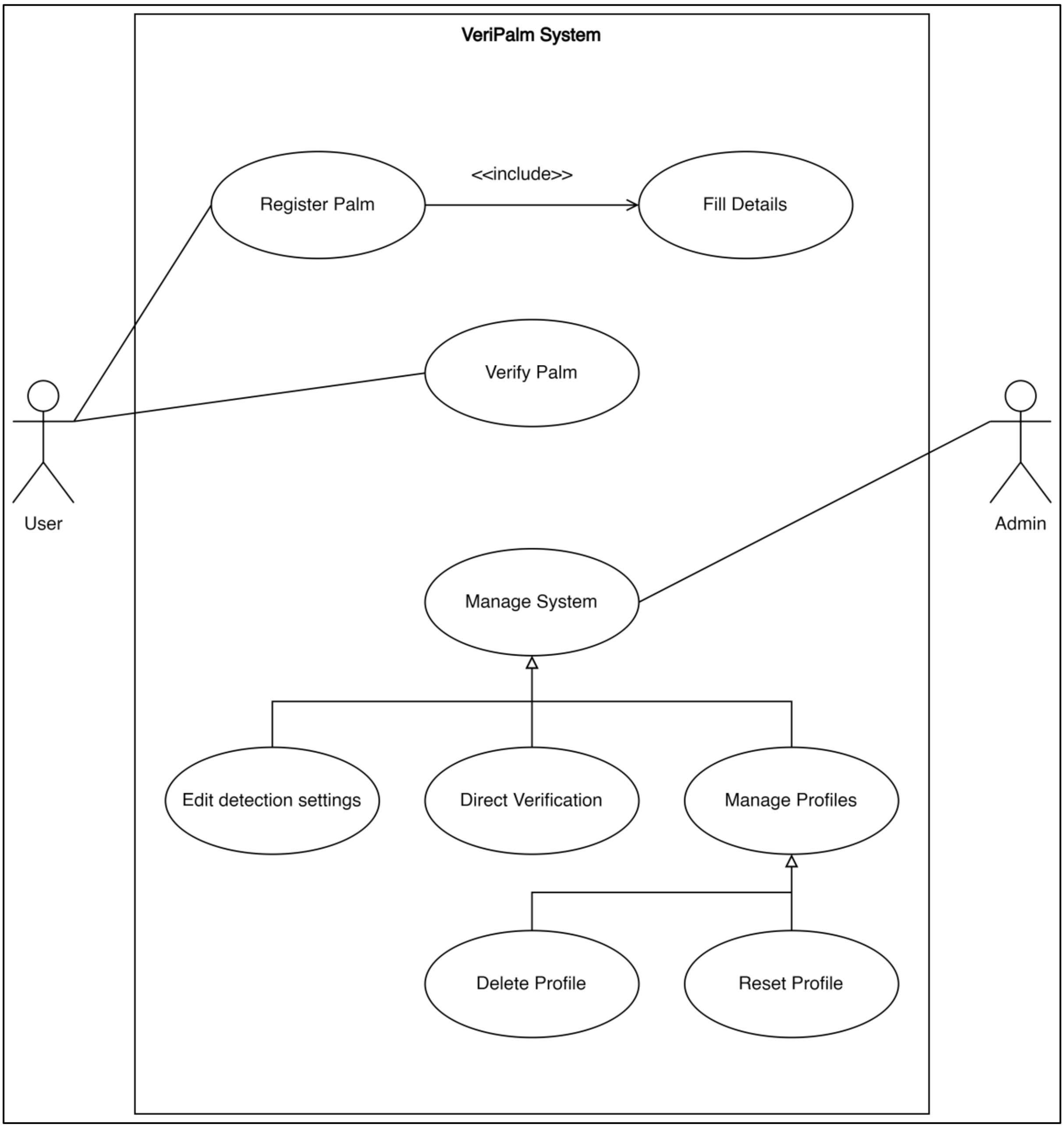


Figure 3.1.2.1 Use Case Diagram for VeriPalm App

The VeriPalm system caters to 2 different types of users which are the user and admin. The highlight of the system is the real-time verification and registration of palmprints using the trained Siamese Network. Additionally, administration function for profile management and model tuning are also available.

3.1.3 Use Case Description

Use Case Name: Register Palm	**ID:** 1	**Importance Level:** Critical
Primary Actor: User		**Use Case Type:** Detail, Essential
Stakeholders and Interests: • User – for registering their palm into the system for future authentications.		
Brief Description: This use case describes the process for registering a palm for new/existing users		
Trigger: Users who wish to register their palm to the system **Type:** External		
<u>Relationships</u> **Association:** User **Include:** Fill Details **Extend:** **Generalization:**		
Normal Flow of Events: 1. User aligns their palm in front of the camera device. 2. User previews the captured ROI on the interface. 3. User clicks on register palm button. 4. System prompts user for their name. 5. System extracts the features from the ROI using the model. 6. System appends the embedding into the user's profile. **7.** System prompts user registration success.		
Sub Flows: None		
Alternate/Exceptional Flows: 4a. System prompts user for their password if the entered name corresponds to an existing profile. 4b. System redirects back to the registration menu if user cancels. 7a. System prompts user a random generated password. Password will be required for future registration to that profile for security purposes.		

Use Case Name: Fill Details	**ID:** 2	**Importance Level:** High
Primary Actor: User		**Use Case Type:** Detail, Essential
Stakeholders and Interests: • User – for users who wish to add their palm to a new profile or existing profile.		
Brief Description: This use case describes the case when user is required to key in their profile names for profile retrieval or registering a new profile.		
Trigger: Users who has confirm to register the captured ROI to the system. **Type:** Internal		
<u>Relationships</u> **Association:** User **Include:** **Extend:** **Generalization:**		
Normal Flow of Events: 1. User enters a name when prompted. If name exists, the S-1: Name exists sub flow will be performed. If name does not exist, S-2: New profile sub flow will be performed.		
Sub Flows: S-1: Name exists 1. System will prompt user for the password associated to their profile. 2. System loads the existing profile. 3. System will append the embedding to the existing profile. S-2: New profile 1. System creates a profile. 2. System generates a password for the new profile. 3. System displays the password for the user.		
Alternate/Exceptional Flows: S-1 1a. User enters wrong password and user redirects back to registration menu		

Use Case Name: Verify Palm	**ID:** 3	**Importance Level:** Critical
Primary Actor: User		**Use Case Type:** Detail, Essential
Stakeholders and Interests: • User – who wishes to verify their palm with the system to retrieve their profile/clock-in		
Brief Description: This use case describes the process for palm print verification in real-time		
Trigger: User who wants to verify their palm **Type:** External		
<u>**Relationships**</u> **Association:** User **Include:** **Extend:** **Generalization:**		
Normal Flow of Events: 1. User aligns their palm in front of the camera device. 2. User previews the extracted ROI on the interface. 3. User clicks on verify palm. 4. System will extract the feature from the ROI and generate the embedding vector. 5. System will compare the embeddings with all the other stored embeddings of other profiles. 6. System will find a match based on the detection threshold and verification confidence set.		
Sub Flows: None		
Alternate/Exceptional Flows: 6a. System will display "Unrecognised Palm" if the palm does not match any embeddings of the profile. 6b. System will display "Welcome user {name}" and display the clock-in time for the user.		

Use Case Name: Manage System	**ID:** 4	**Importance Level:** Critical
Primary Actor: Admin		**Use Case Type:** Detail, Essential
Stakeholders and Interests: • Admin – for managing user profiles and editing the system settings		
Brief Description: This use case describes the case when the admin intends to perform administration work on the system which covers editing detection thresholds, and profile management.		
Trigger: Admin who wishes to perform administration work **Type:** External		
<u>**Relationships**</u> **Association:** Admin **Include:** **Extend:** **Generalization:**		
Normal Flow of Events: 1. Admin chooses the "Admin Console" setting. 2. System prompts for the admin password.		
Sub Flows: None		
Alternate/Exceptional Flows: 2a. The password entered is correct, and system allows admin to access the console and displays the admin settings. 2b. The password entered is wrong, and the system prevents access to the console.		

Use Case Name: Edit Detection Settings	**ID:** 5	**Importance Level:** High
Primary Actor: Admin		**Use Case Type:** Detail, Essential
Stakeholders and Interests: • Admin – when admin decides to edit detection settings.		
Brief Description: This use case describes the case when the admin wishes to edit the detection settings which include profile detection threshold, average confidence threshold, matching threshold, and strict registration mode.		
Trigger: Admin who wants to edit the detection settings **Type:** External		
<u>Relationships</u> **Association:** Admin **Include:** **Extend:** **Generalization:** Manage System		
Normal Flow of Events: 1. System displays the threshold sliders and strict registration mode checkbox for the admin 2. Admin edits the settings 3. Admin clicks the close button after completing edit		
Sub Flows: None		
Alternate/Exceptional Flows: None		

Use Case Name: Direct Verification	**ID:** 6	**Importance Level:** High
Primary Actor: Admin		**Use Case Type:** Detail, Essential
Stakeholders and Interests: • Admin – for directly verifying 2 palm ROI images via direct upload		
Brief Description: This use case describes the case when the admin wishes to directly verify 2 palm ROIs and bypassing the registration and verification steps.		
Trigger: Admin who wishes to verify 2 palm ROIs **Type:** External		
<u>Relationships</u> **Association:** Admin **Include:** **Extend:** **Generalization:** Manage System		
Normal Flow of Events: 1. Admin clicks on direct verification. 2. Admin uploads image 1 and image 2. 3. Admin clicks the verify palm images button. 4. System displays to user the match status and confidence value.		
Sub Flows: None		
Alternate/Exceptional Flows: None		

Use Case Name: Manage Profiles	**ID:** 7	**Importance Level:** High
Primary Actor: Admin		**Use Case Type:** Detail, Essential
Stakeholders and Interests: • Admin – for managing profiles of the registered users in the system		
Brief Description: This use case describes the case when the admin wants to perform profile management.		
Trigger: Admin wants to manage the profiles of the users **Type:** External		
<u>Relationships</u> **Association:** Admin **Include:** **Extend:** **Generalization:**		
Normal Flow of Events: 1. Admin clicks on the manage registered profiles button. 2. System displays all user's profiles in a form of a table with reset and delete buttons associated to each profile.		
Sub Flows: None		
Alternate/Exceptional Flows: None		

Use Case Name: Delete Profile	ID: 8	Importance Level: High
Primary Actor: Admin		**Use Case Type:** Detail, Essential
Stakeholders and Interests: • Admin – deleting a profile from the system		
Brief Description: This use case describes the process of deleting a user's profile		
Trigger: Admin who wants to remove an existing profile from the system **Type:** External		
Relationships **Association:** Admin **Include:** **Extend:** **Generalization:** Manage Profiles		
Normal Flow of Events: 1. Admin clicks on the delete button associated to that user's profile. 2. System prompts user for confirmation on deleting the profile 3. System fetches the updated list of profiles and updates the table view.		
Sub Flows: None		
Alternate/Exceptional Flows: 2a. Admin declines the action, and is redirected to the table view 2b. Admin confirms and proceeds with the profile deletion.		

Use Case Name: Reset Profile	**ID: 9**	**Importance Level: High**
Primary Actor: Admin		**Use Case Type:** Detail, Essential
Stakeholders and Interests: • Admin – reset a profile's embeddings in the system		
Brief Description: This use case describes the process of resetting a user's profile		
Trigger: Admin who wishes to reset a user's profile **Type:** External		
<u>**Relationships**</u> **Association:** Admin **Include:** **Extend:** **Generalization:** Manage Profiles		
Normal Flow of Events: 1. Admin clicks on the reset button associated to that user's profile. 2. System prompts admin for confirmation. 3. System fetches the updated list of profiles and updates the table view.		
Sub Flows: None		
Alternate/Exceptional Flows: 2a. Admin declines and will be redirected to the table view. 2b. Admin confirms, and the profile will be reset		

3.1.4 Class Diagram

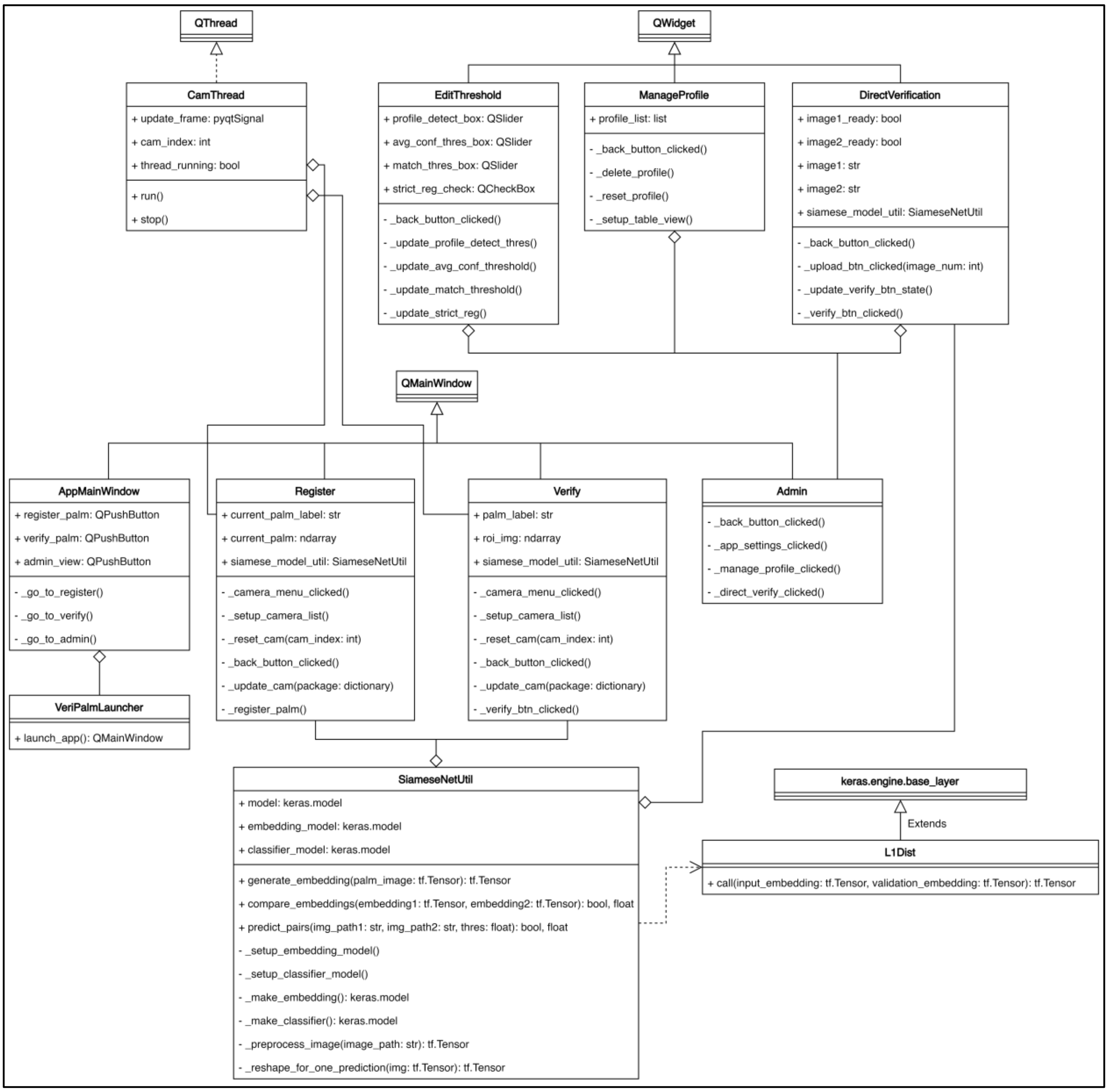


Figure 3.1.4.1 UML Class Diagram for VeriPalm System

Figure 3.1.4.1 shows the UML class diagram for the VeriPalm system. The application will mainly be developed using the Python GUI framework called PyQT5. Object-Oriented principles will be used to facilitate the design of the system.

The SiameseNetUtil Class will be used to encapsulate the trained Siamese Network model and provide functions that ranges from image preprocessing to model prediction related functions like compare_embeddings() and predict_pairs(), which will be called and used by application classes. The SiameseNetUtil will use a class called L1Dist, which inherits the keras.engine.base_layer class. The L1Dist is a custom class that is implemented as a layer in the custom-built Siamese Network, this class will calculate the L1 distance between 2 feature vectors.

The other classes such as the Register, Verify, Admin, AppMainWindow, are classes that represent each view of the system. These classes inherit the QMainWindow class and contain functions for event handling. The elements such as the buttons, dropdown lists on the view classes are given functionally via event-driven design.

Furthermore, the CamThread class which inherits the QThread class, provides the functionality to access the camera using another thread. Additionally, several popup windows that are contained by the Admin class such as the EditThreshold, ManageProfile, and DirectVerification classes will inherit the QWidget class. These classes represent popup views under the Admin class which serve different functionalities.

In short, the functionalities and views of the system are encapsulated in a Class to provide ease of implementation and modularity to the system.

Chapter 4

System Implementation

In this chapter, the implementation details of all modules of the system will be discussed in detail. The development of the system is divided into 6 distinct phases as shown below.

i. Project pre-planning and camera setup.
ii. Localization and segmentation of ROI from palm image.
iii. Dataset collection and preparation.
iv. Model design, training, and fine-tuning.
v. Model testing and evaluation process.
vi. Application development and model integration

4.1 System Block Diagram

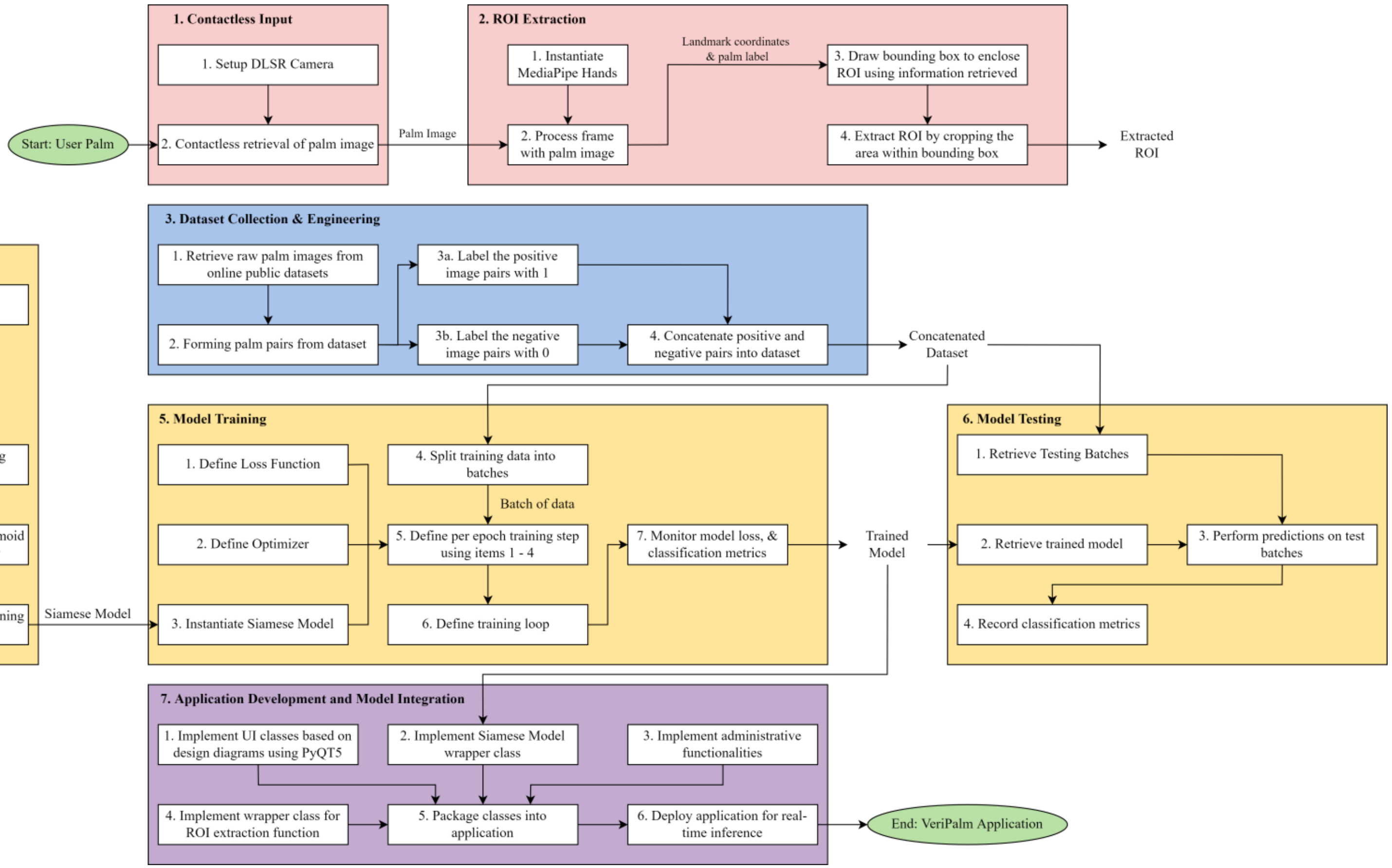


Figure 4.1.1 End-to-end System Block Diagram

Based on the block diagram above, the system is comprised of 7 main modules namely, Contactless Input, ROI Extraction, Dataset Collection and Engineering, Siamese Model Design, Model Training, Model Testing, and finally, Application Development and Model Integration. Each module is colour coded and grouped according to the nature of their functionalities. For example, model training and deep learning related modules are colour coded with yellow.

Each module consists of several tasks to be performed in sequence in order to achieve its functionality. The outputs of modules no. 1 to no. 6 will then be integrated into module no. 7 and packaged as an application that consists of the completed functions of all modules. Each module's roles and underlying implementation details will be discussed in the upcoming subchapters.

4.2 Tools and Technologies

This section will discuss on the tools and technologies involved in implementing the system.

4.2.1 Hardware Setup

Device	Specification	Description
Laptop	Operating System	Windows 10 (64-bit)
	Memory	16GB
	CPU	Intel Core i7-7700HQ
	GPU	NVIDIA GTX1050M
	Storage	500GB Samsung SSD
DLSR Camera	Brand	CANON
	Model	EOS 70D
	Photo Resolution	20 Mega Pixels
	Sensor Type	CMOS
	ISO Sensitivity	100-12800
	Auto Focus Configuration	Phase & Contrast Detection
	Video Resolution (Framerate)	1920 x 1080 (24 – 30)
	Lens (MM)	Canon EF or EF-S Mount (15-85)

A DLSR Camera was chosen as the camera input device for the system as the palm photos that are captured have a higher quality and are less prone to lighting issues or image distortions.

4.2.2 Required Software and Setup

Name	Type	Description
Python	Programming Language	The main programming language for the project will be Python. The motivation behind selecting Python is because it is a high-level programming language that has a syntax that is relatively simple to understand. Furthermore, it also contains robust machine learning and deep learning APIs, libraries, and frameworks. Additionally, there is wide community support for python machine learning and deep learning works.
OpenCV	Python Library	OpenCV will be used for image processing related operations within the project. The main usage of OpenCV will be seen during the segmentation phase, when drawing the ROI bounding box and for cropping the ROI from the palm.
NumPy	Python Library	NumPy will be used alongside OpenCV for representing the images captured in a format of an NumPy array. The mathematical functions found within NumPy will also be used during the ROI segmentation phase for calculating the dimensions of the bounding box.
TensorFlow	Python Library	TensorFlow will be the main deep learning library to be used in this project. Particularly, the TensorFlow functions will be used via the Keras API for ease of usage and implementation. TensorFlow will be used for dataset preparation, image processing operations, data loading, model design, training, testing and evaluation.
MediaPipe Hands	Python Library	MediaPipe Hands is used for tracking and localizing the palm on the video frame captured by the camera device. MediaPipe Hands is a ML library by Google that contains models to extract the key landmarks on palm images. The landmark information will be used for extracting the ROI from the palm.
PyQT5	Python Library	The PyQT5 GUI library will be used to package the different modules of the system into a complete application with a user interface. PyQT5 features a designer tool to create user-interfaces and allows for adding functionalities to the UI via event-driven programming.
QtDesigner	Software	QtDesigner is an application that allows for the development of Qt UI pages via drag-and-drop. The pages can be exported as components with .ui extension and can be loaded into code.
PyCharm	Python IDE	The main IDE for developing the application will be carried out using the PyCharm IDE. PyCharm is chosen as it has a built-in debugger, which will be useful for identifying and fixing errors during implementation.

4.3 Project Timeline

Project Tasks	Duration (Weeks)	Y3S3 Weeks (24th Jan 2022 – 1st May 2022)													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
FYP2 Report – Chapter 1 & 2	1														
FYP2 Report – Chapter 3	2														
FYP2 Report – Chapter 4 Onwards	4														
Dataset Collection and Engineering	2														
Model Design and Training	2														
Application Design and Model Integration	2														
Finalizing and packaging of the application	4														
Application testing phase	2														
FYP2 Draft Report Submission	1														
FYP2 Report – Chapter 6	1														
FYP2 Report Finalization and System Refactoring	4														
FYP2 Presentation	2														

Based on the timeline of FYP2, the remaining modules of the system such as dataset preparation, model design and training, and application development will be done. All development is expected to be completed by Week 10 of the trimester. Testing phase will soon commence after all implementation has been completed.

4.4 Project Preplanning and Camera Setup

4.4.1 Overview

In the project preplanning phase, the tasks to be done are the preparation of the DLSR camera and also implementing the codes for interfacing the camera for capturing images using OpenCV.

4.4.2 Block Diagram

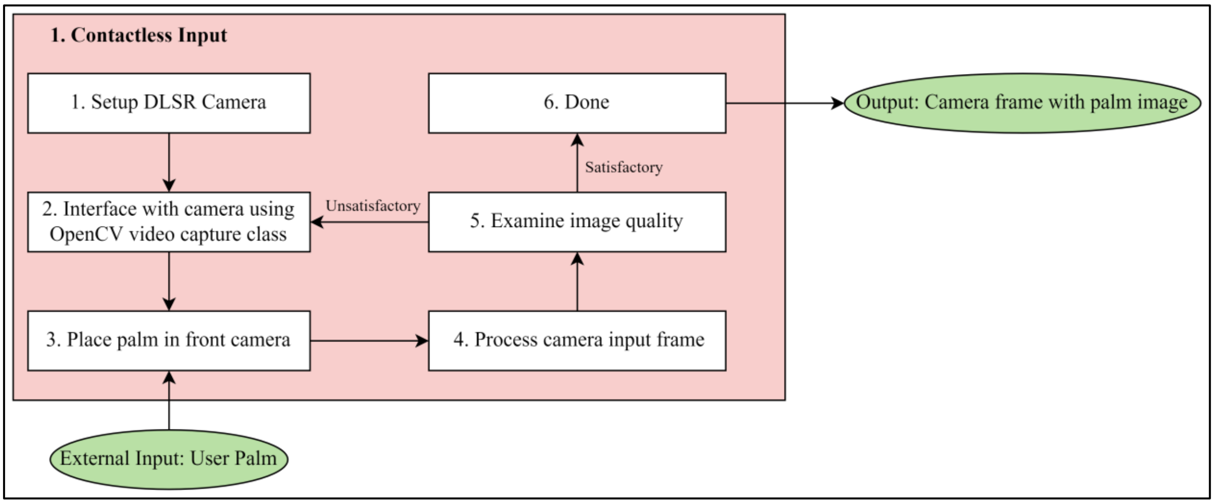


Figure 4.4.2.1 Project Preplanning and Camera Setup Block Diagram

4.4.3 Requirements

The outcome of this module is to be able to retrieve the real-time captured video frames from the camera using OpenCV. The video frames that are represented as NumPy arrays can be passed into the ROI extraction module to undergo ROI extraction in real-time.

4.4.4 Implementation Details

The palm print images will be captured using the camera device mentioned in section 4.2.1. The camera is mounted to a tripod for easier repositioning.

Figure 4.4.4.1 Camera Setup

The figure above depicts the setup of the camera. The lens of the camera features auto focusing technology and will be helpful during the capturing process as it will be able to lock on to the palm.

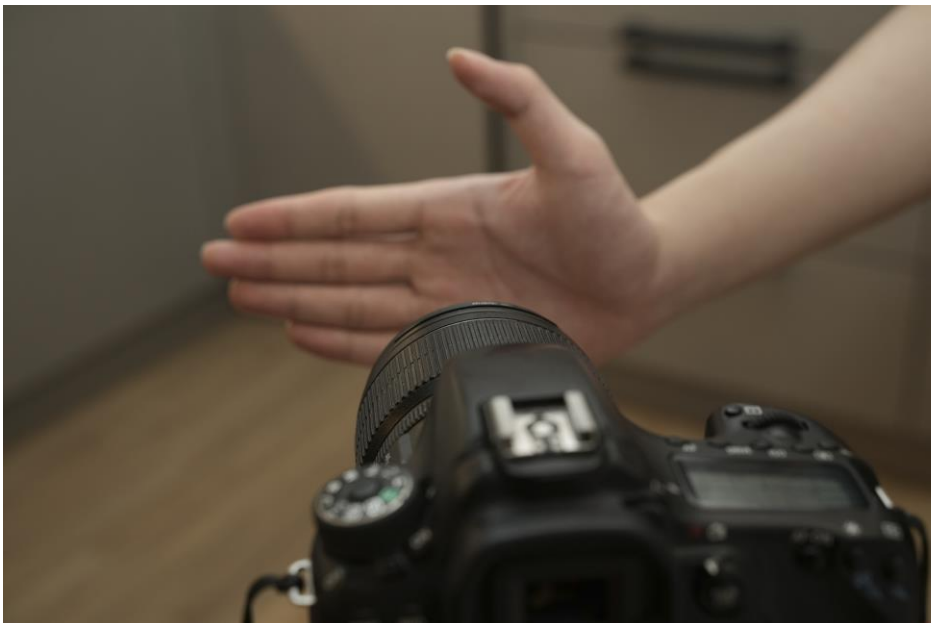


Figure 4.4.4.2 Capturing Process for Right Palm

Figure 4.4.4.3 Capturing Process for Left Palm

Both figures above depict the method of placing the user’s left and right palm in front of the camera respectively.

The camera is connected using USB. In order to interface with OpenCV, a driver provided by the camera’s manufacturer has to be downloaded and installed in order for it to be recognised as a webcam device. The camera will then be discoverable by the OS after the software has been installed. Doing so will allow the camera to be accessed as a webcam device in Python code through the OpenCV VideoCapture() class.

Once the camera can be successfully accessed by OpenCV, the next step will be to retrieve the video frames from the connected camera. The OpenCV VideoCapture() class provides a method called .read() which returns the a Boolean indicating the frame is successfully read and the live video frame represented as an NumPy array. At this point, the user may place their palm in front of the camera and allow the camera to capture the live frames with the palm in it.

Then 3 main image processing operations are needed to be done before passing it for ROI extraction. Firstly, the frame will have to be converted to the RGB colour space, this step is needed as the frame will be displayed on the interface for the user to verify, as by default, the frames read by OpenCV are in the BGR colour space. Then, the frame will be flipped along the y-axis so that the video frame is seen as a mirror image for the user. Afterwards, the video frame will be resized to a shape of 640 by 480.

After completing the image processing operations, the video frame will then be displayed for examining the image quality. If the image quality is unsatisfactory, certain tweaking of the camera lens will be required. Otherwise, the video frame can then be passed into the ROI extraction module for segmentation.

4.5 ROI Segmentation and Localization

4.5.1 Overview

The tasks needed to be performed during this stage is mainly on segmenting the feature abundant ROI from the palm. MediaPipe Hands will be used to assist the ROI extraction process. The input to this module is a video frame with a palm image which is be supplied by the previous contactless input module. The video frame will be subjected to various MediaPipe Hands functions to extract the landmark information on the hand for segmenting the ROI.

4.5.2 Block Diagram

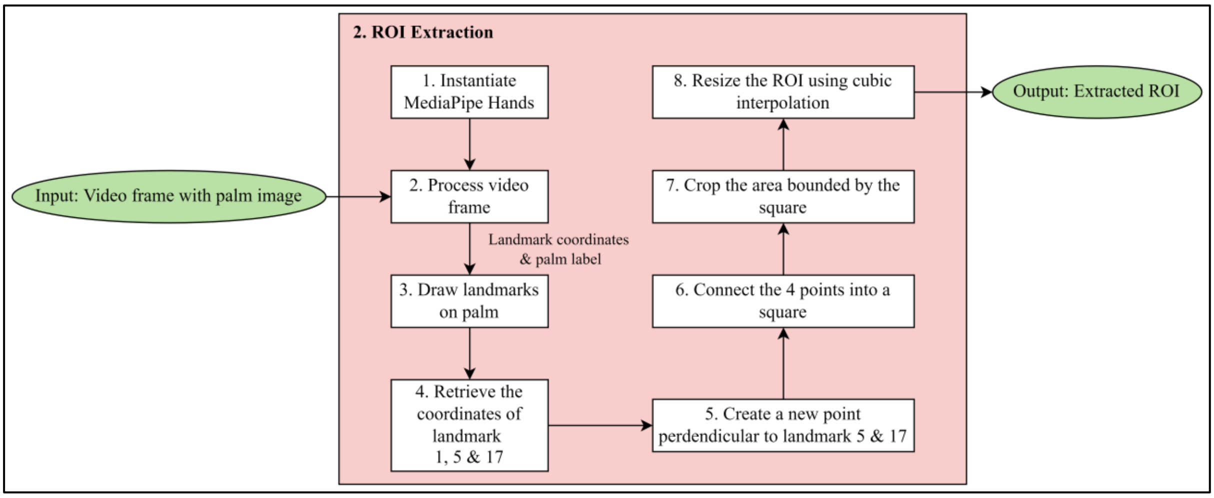


Figure 4.5.2.1 ROI Extraction Block Diagram

4.5.3 Requirements

The outcome for this module is the segmented ROI from the video frame. A successful model prediction relies on the proper segmentation of the ROI, as it contains features which the CNN models like EfficientNet can learn and extract well.

4.5.4 Implementation Details

The ROI of the palm is depicted in the diagrams below.

Figure 4.5.4.1 Sample palm image acquired from the Tongji Database

Figure 4.5.4.2 ROI extracted and aligned

Referring to the sample above, the ROI (Region of interest) is a square section in the middle of the palm which contains most of the wrinkles, principal lines, and other textural features. The palm print verification process operates on the ROI instead of on the whole palm print, hence a proper segmentation needs to be performed to extract the ROI.

Previous works proposed the usage of traditional image processing techniques such as calculating image centroid and maximum inscribed circle to extract the ROI from the palm [11]. In this project, MediaPipe Hands will be used to extract the ROI from the palm. This deep learning approach of ROI segmentation is seen as a novelty to other works, as no other related works had used this approach of segmentation.

The ROI extraction is modularized into a single process that accepts a video frame containing a palm, and then outputs the ROI extracted from the video. The ROI segmentation can be done in real-time as MediaPipe Hands is a lightweight model that can perform inference on palm images to obtain its landmark information quite quickly.

The first step in this module is to instantiate the MediPipe Hands class. The class contains a slew of functions which include processing hand images, retrieving the hand landmarks, and also drawing the landmarks on the palm. MediaPipe Hands can be instantiated by calling mediapipe.Hands() and also specifying the min_detection_confidence and min_tracking_confidence parameters. The former parameter specifies the detection confidence that MediaPipe will use to search a palm in the supplied frame, a higher value indicates a stricter detection. The latter parameter specifies the confidence in tracking the moving palm across consecutive video frames. In this case, the values were set to 0.8 and 0.5 respectively.

After instantiating the class, the process() method is called, and the current captured video frame is passed as an argument. The output returned by the process() method is an object that contains a few attributes. The attributes that were used are multi_hand_landmarks and multi_handedness. The multi_hand_landmarks contain a collection of landmark information and its coordinates for each detected hand, whereas the multi_handedness contains the information about the detected hand and also its detection confidence.

The landmarks for the detected hand is then used along the draw_landmarks() function draw the landmarks on the hand. Figure 4.5.4.3 shows the landmarks plotted.

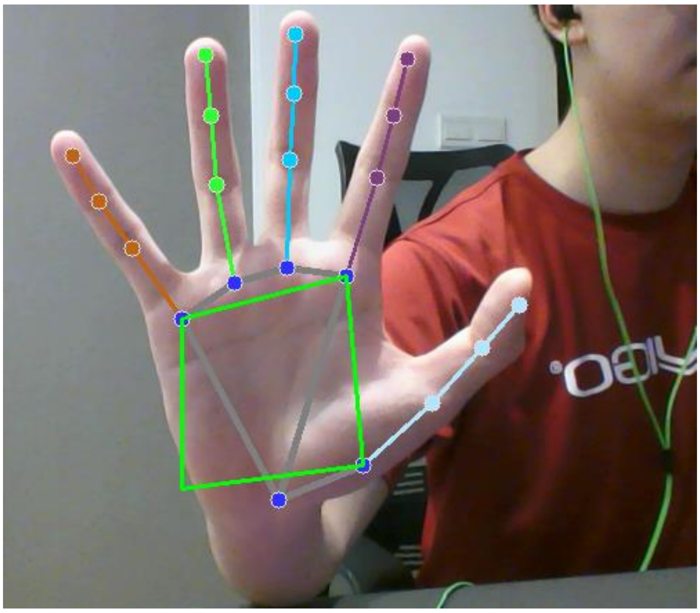


Figure 4.5.4.3 Plotted landmarks on the left hand

Afterwards, the landmark information of the detected hand is then passed into a self-defined function. The function will translate the landmark points that are needed to draw the bounding box of the ROI into the actual coordinates relative to the video frame. The points that are needed to enclose the ROI are shown below.

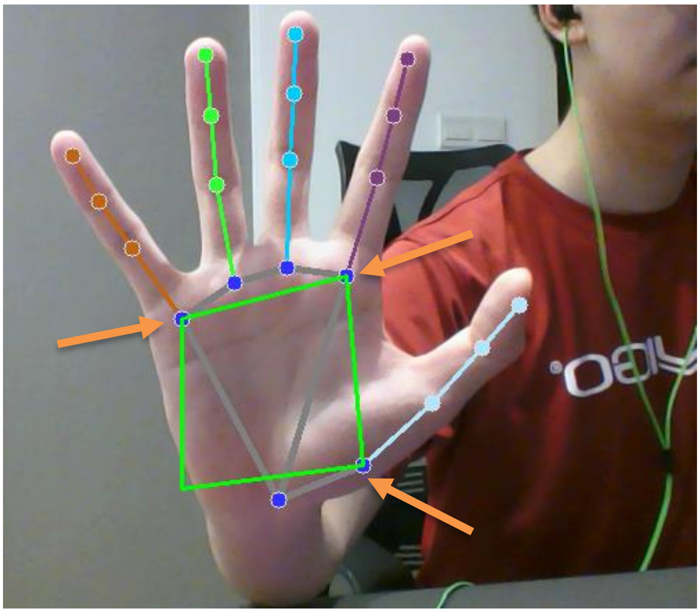


Figure 4.5.4.4 Landmarks points that encloses the ROI

Based on the hand landmark diagram of MediaPipe Hands, the names of the 3 points are the PINKY_MCP, INDEX_FINGER_MCP, and THUMB_CMC. The real-world x & y coordinates are then retrieved from the aforementioned points. To ease the process of working with the coordinates, a wrapper class named Point is written to encapsulate the details of the point and its coordinates. The class diagram for Point is shown below.

Point
+ x: float + y: float
+ distance_between_point(other_point: Point): float

Figure 4.5.4.5 Point Class Diagram

Before drawing the bounding box, and additional point will be required to form the 4 corner points of the bounding box. Hence, an additional point (referred as "point 4") that is located below the PINKY_MCP will be calculated.

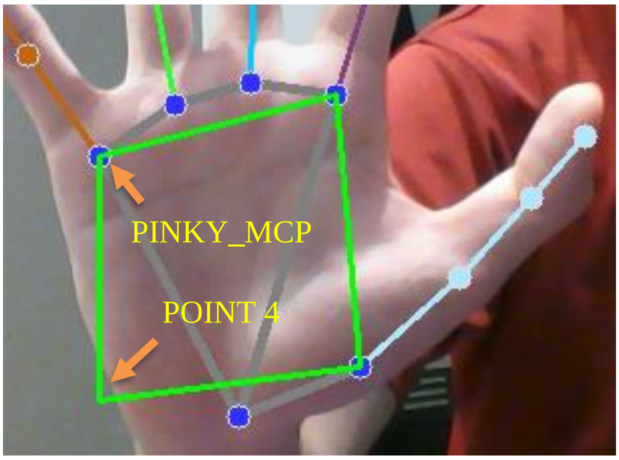


Figure 4.5.4.6 New point – Point 4

Point 4 is calculated using the following formula:

1. Let p1 be the INDEX_FINGER_MCP and p3 be the PINKY_MCP
2. Point 4's x and y coordinates are then computed as follows

$$Length = \sqrt{(p2.x - p3.x)^2 + (p1.y - p3.y)^2}$$

$$Point4.x = p3.x + length \times cos\left(\frac{\pi}{2}\right)$$

$$Point4.y = p3.y + length \times sin\left(\frac{\pi}{2}\right)$$

After calculating point 4, the bounding box can then be plotted using the 4 points. The function to draw the bounding box is using the OpenCV line() method. The resulting ROI can then be extracted by cropping the region bounded by the 4 points. The cropped ROI will be resized to a standard size of 300 by 300 pixels via cubic interpolation. The figure below shows the extracted ROI from the palm.

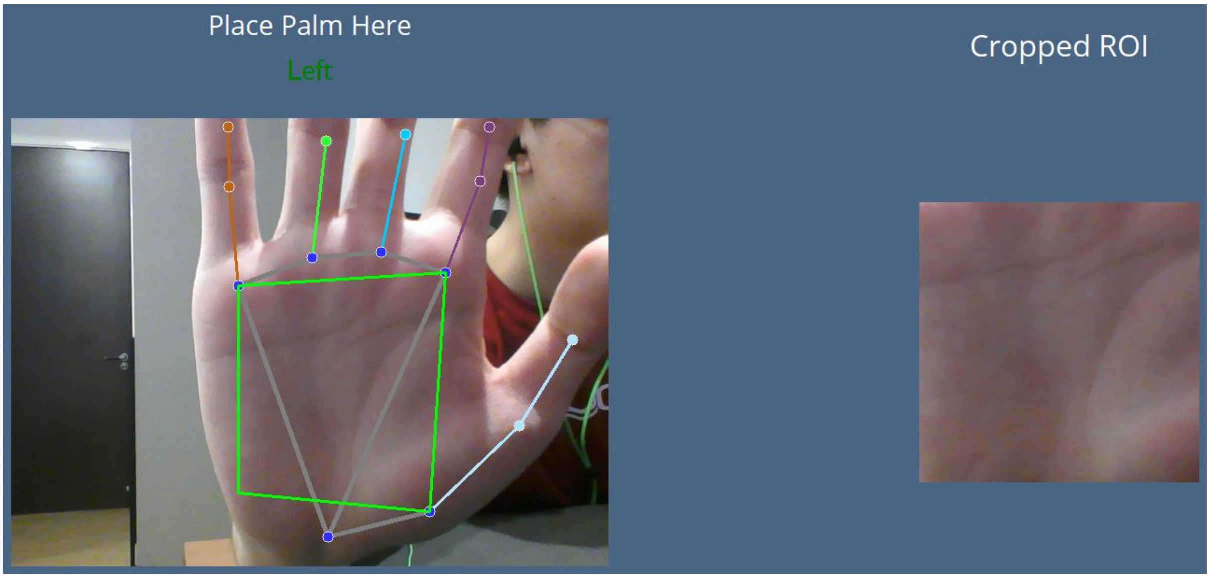


Figure 4.5.4.7 ROI extraction performed in real-time

As seen from the figure above, the image quality of the extracted ROI is high, and does not lose to the well-known methods done in previous works. In fact, this method of ROI extraction is less tedious compared to traditional manually designed segmentation schemes, but still produces a high-quality ROI.

The ROI extraction module can then be used during the application development to segment ROI from real-time camera inputs. Besides, this module can also be used to extract from unprocessed palm images and generate additional training samples, as seen below.

Figure 4.5.4.8 ROI extracted from raw images

4.6 Dataset Collection and Reengineering

4.6.1 Overview

In order to differentiate between 2 presented palm images, a Siamese Neural Network will have to be trained using palm image pairs. Several publicly available palmprint datasets were acquired for this purpose. The datasets were reengineered as images pairs and are labelled as 1 if they match and labelled as 0 otherwise.

4.6.2 Block Diagram

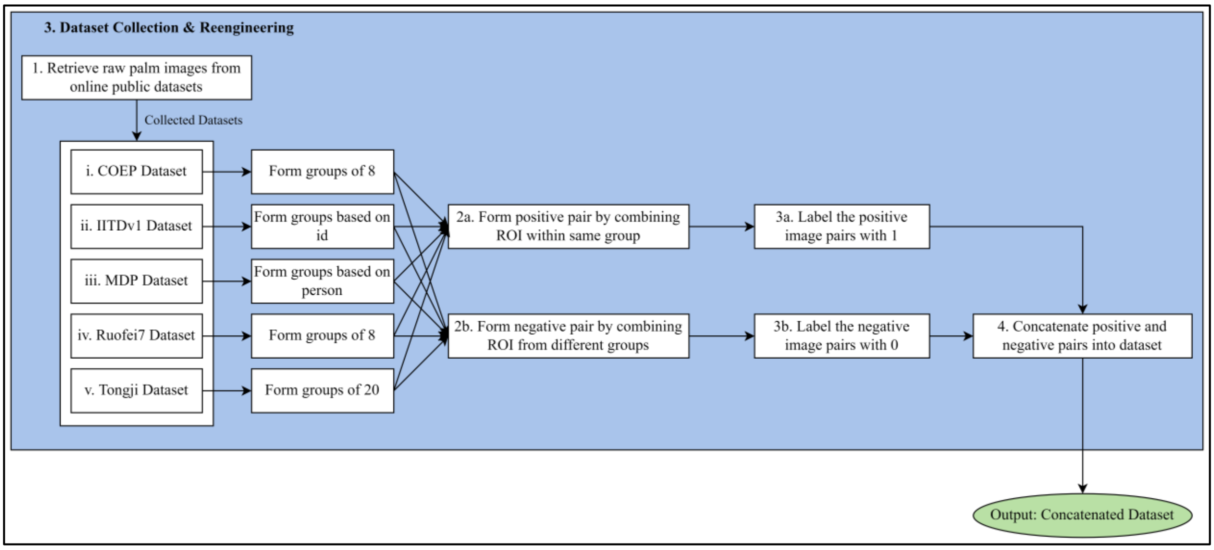


Figure 4.6.2.1 Dataset Collection and Reengineering Block Diagram

4.6.3 Requirements

The outcome of this module is to generate a dataset of palm image pairs which are reengineered from raw public palm datasets with their appropriate labels. The dataset of palm image pairs will then be used as the training data for the Siamese Neural Network in the upcoming module. In short, similar image pairs are denoted as positive pairs, while dissimilar image pairs are denoted as negative pairs.

4.6.4 Implementation Details

Five public palmprint databases were acquired for model training purposes. The names of the databases are the COEP database, IITDv1, MediaPipe ROI, ruofei7 database, and the Tongji database. Each database provides a large sample of raw and unlabelled palmprint ROI images. The following figures shows the samples from each database.

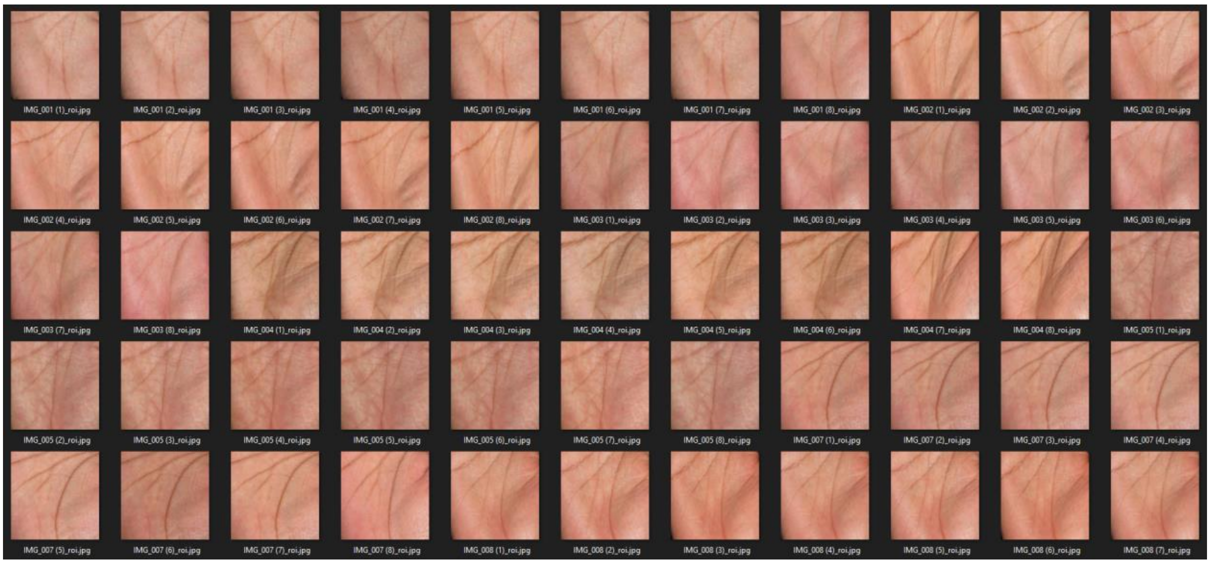


Figure 4.6.4.1 COEP Right Palm ROIs

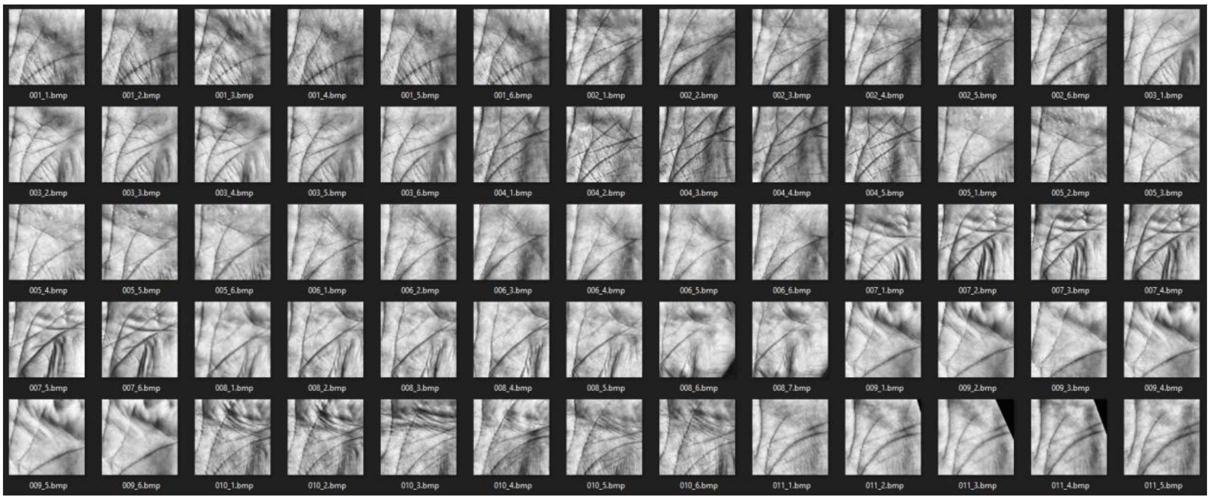


Figure 4.6.4.2 IITDv1 Left Palm ROI

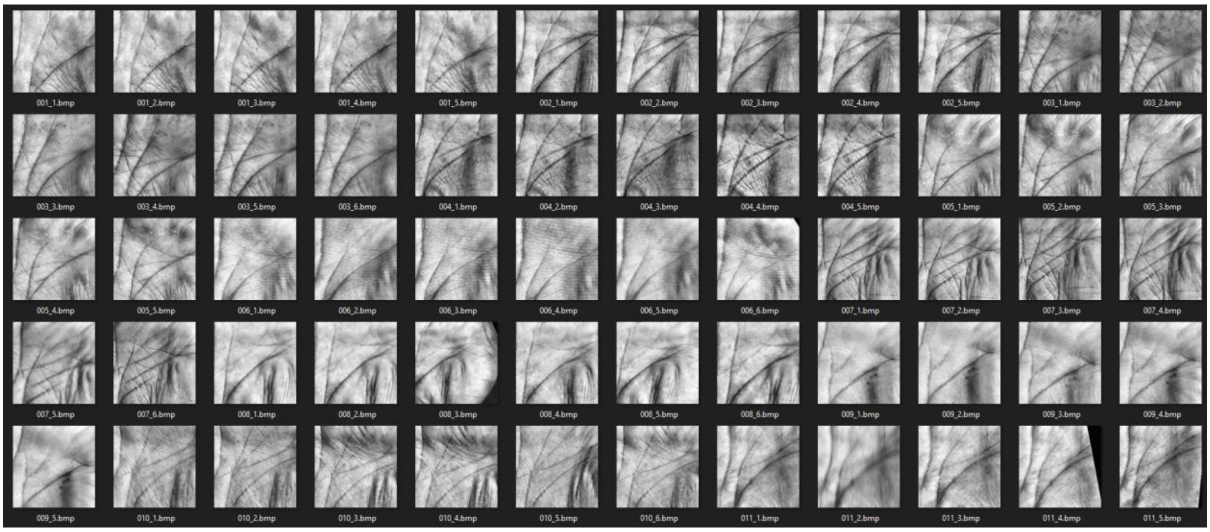


Figure 4.6.4.3 IITDv1 Right Palm ROI

Figure 4.6.4.4 MediaPipe ROI Samples

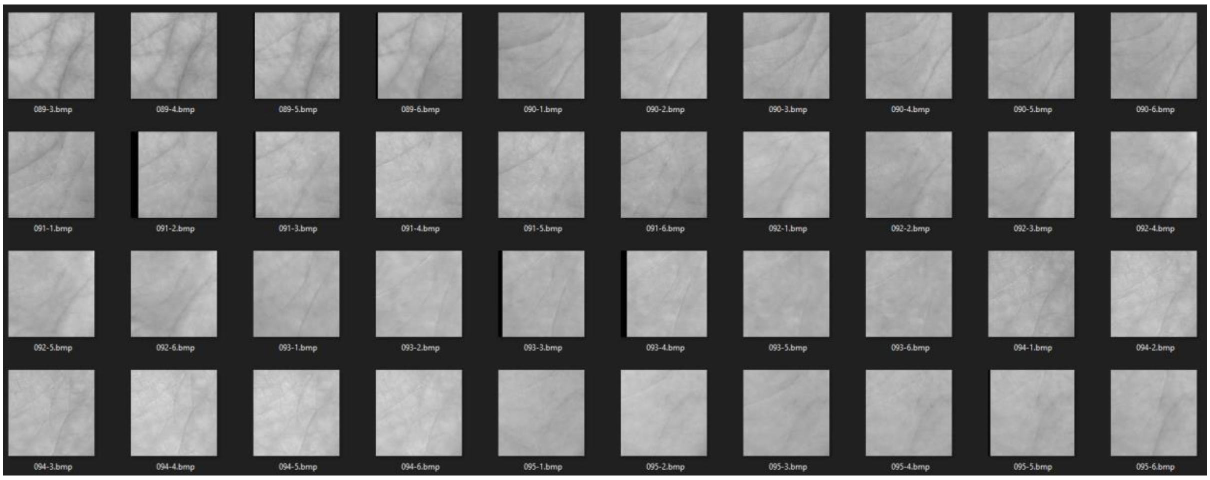


Figure 4.6.4.5 Ruofei7 ROI

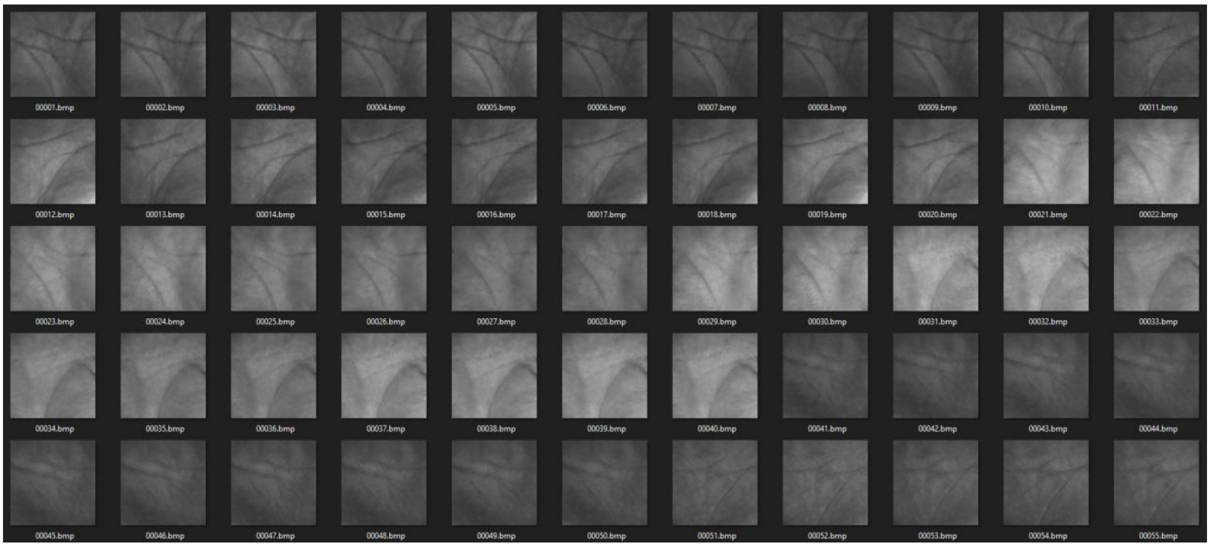


Figure 4.6.4.6 Tongji ROI Session 1

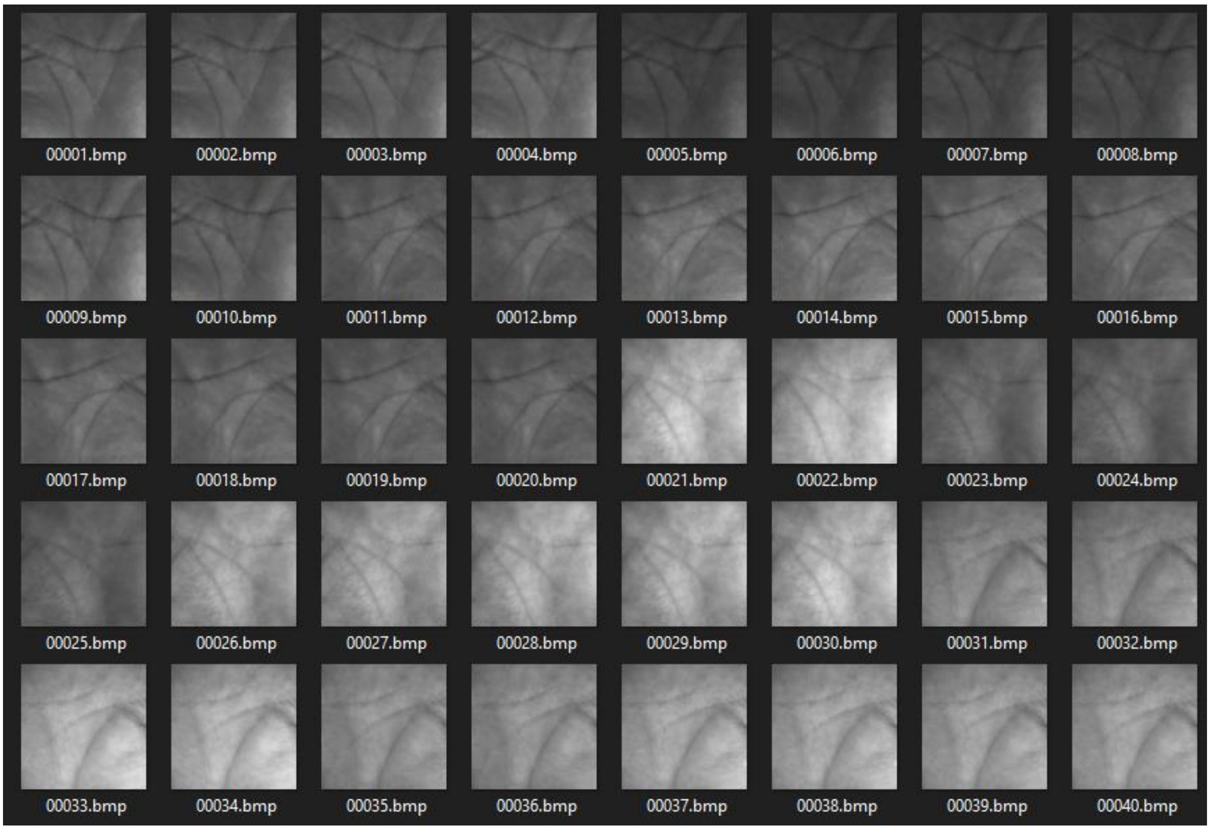


Figure 4.6.4.7 Tongji ROI Session 2

The reason of sourcing ROI images from 5 different databases is to introduce heterogeneity to the final training dataset, as the samples from the 5 databases are distinct and different from one another. For example, images from the COEP database and MediaPipe database are coloured images, while the other databases are grayscale images. The properties of each dataset are listed in the table below.

Table 4.6.4.8 Summary of datasets

Database	ROI Image Properties	Number of images	Number of palms	Image Size	Capture Method	Acquisition
COEP	Coloured	1305	167	300x300	Contactless	Extracted from raw image using ROI extraction module
MediaPipe ROI	Coloured	4000	5	250x250	Contactless	Self-collected
IITD	Grayscale	2601	230	150x150	Contactless	Official ROI provided
ruofei7	Grayscale	594	99	128x128	Contactless	Official ROI provided
Tongji	Grayscale	12,000	1200	128x128	Contactless	Official ROI provided

Based on the table above, the IITD, ruofei7 and Tongji databases had provided the official ROIs for usage. Two databases namely, COEP and MediaPipe are self-extracted and compiled using the ROI extraction module. The COEP ROI is extracted from raw palm images given in the COEP dataset.

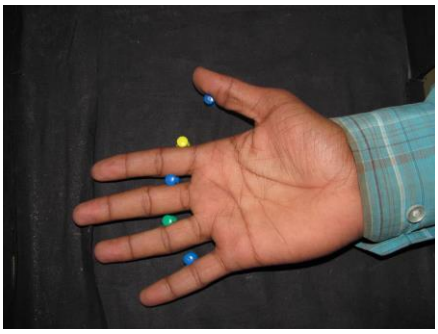


Figure 4.6.4.9 ROI Extracted from raw palms images of COEP database

On the other hand, the MediaPipe ROI images were acquired by extracting the ROI from the palms of a few volunteers to this project. The dataset was named MediaPipe ROI, because the main algorithm used to segment the ROI was done via the ROI extraction module. The setup of the capturing process is depicted as below.

Figure 4.6.4.10 MediaPipe ROI capturing process

After gathering the images from each database, the next process is to divide the images groups based on the palms. Referring to the documentation provided by the databases, the images are separated into groups according to each distinct user. For instance, the Tongji database has 1200 unique palms, each contributing 10 images each, leading to 12000 images in total.

A helper function called chunks() was defined to divide n-sized groups from elements in a list. After dividing images into their respective groups based on the palm, the combination() function provided by the itertools library was called on each group of palm images.

The combination() function works by forming pairs based on the elements in each group. For example, take a list of images [img1, img2, img3, img4, img5, img6, img7, img8], the combinations function will return a list of tuples as such, [(img1, img1), (img1, img2) … (img8, img8)]. The following scheme is then repeated for each database to form the positive pairs.

To form the negative pairs, the product() function was used instead of the combination() function. The product() function works by forming a cartesian product between 2 lists. For example, given 2 lists of images of different palms, the product() function returns [(img1, img9), (img1, img10) … (img8, img16)]. Due to product() forming a cross product between all elements of 2 lists, downsampling is done to prevent class imbalance.

After forming the image pairs, the positive_pairs list and negative_pairs list were exported to a csv file of image path strings for future use. Then, the positive_pairs list and negative_pairs list are loaded as tensorflow Dataset class. The positive class are labelled with a 1 and negative class are labelled with a 0 using tf.ones and tf.zeros respectively.

The final dataset is obtained by concatenating the positive pairs and negative pairs vertically. The final dataset will be of shape (972893, 3) where 972893 indicates the number of samples and 3 indicates the images pairs and their corresponding labels. The dataset is then passed into the model training module as training data.

4.7 Model Design, Training and Fine-Tuning

4.7.1 Overview

This section serves as one of the highlights of the project. This module will train a special deep learning neural network called a Siamese Network that aims to differentiate between 2 input images. The motivation behind the usage of a Siamese Network is to learn a similarity function instead of learning to classify a predetermined number of classes.

Unlike conventional CNNs that are trained to classify image classes via the softmax loss function. A Siamese network will receive 2 input images, extracts its features using a weight sharing CNN, and then calculate the distance between 2 feature vectors using a distance metric. Finally, the distance vector is then passed into a sigmoid classifier that outputs the matching probability.

4.7.2 Block Diagram

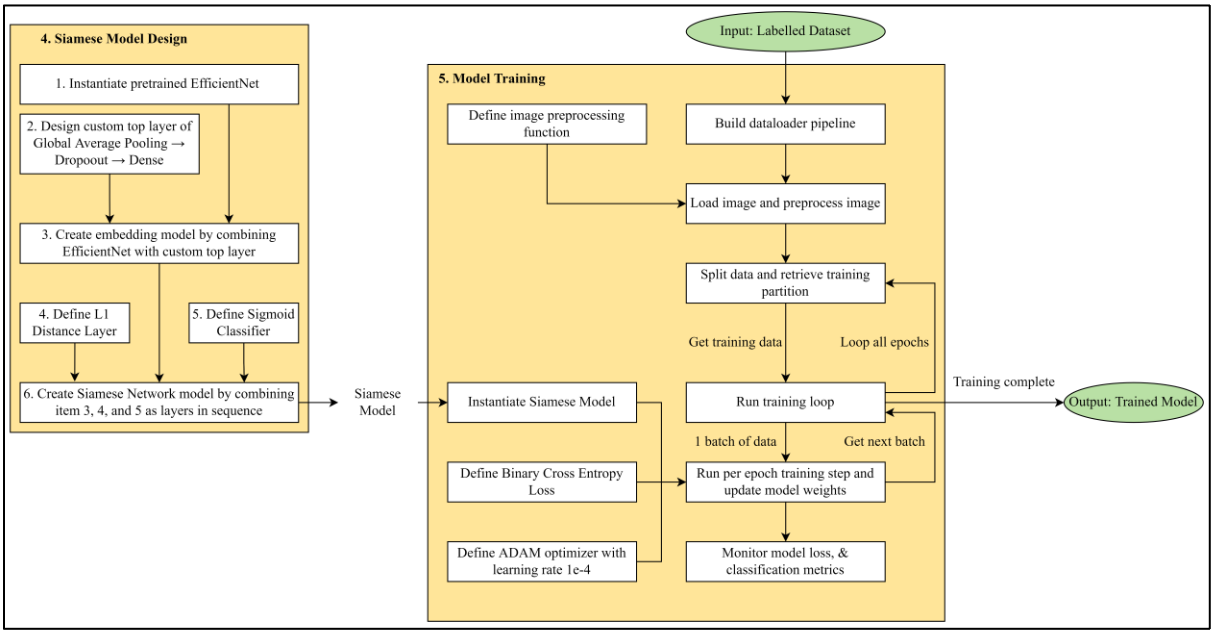


Figure 4.7.2.1 Bloc Diagram for Model Design and Training

4.7.3 Siamese Model Architecture

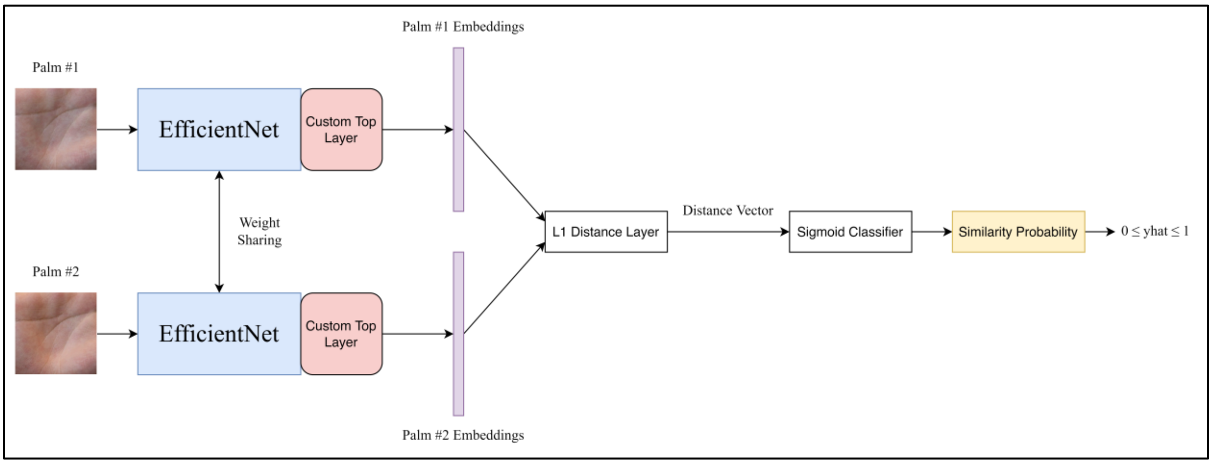


Figure 4.7.3.1 Siamese Model Architecture

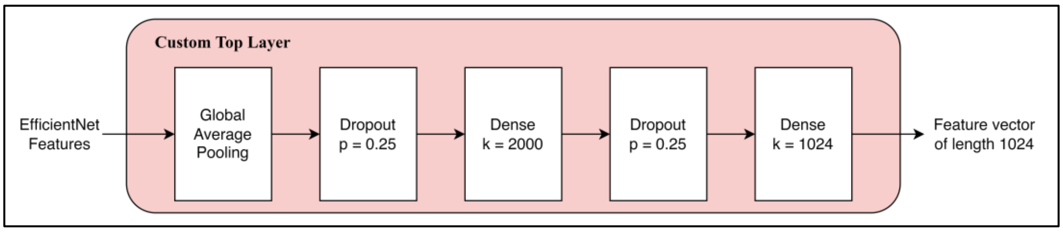


Figure 4.7.3.2 Custom Top Layer

4.7.4 Requirements

The outcome of this module is a trained Siamese Network that is able to determine whether both input palm images same or do not match. Besides, the embedding model can also be extracted from the Siamese model to extract features from the palm image. The model will then be saved as a single artifact which can be then deployed in the application.

4.7.5 Implementation Details

Firstly, the Siamese Network is constructed based on the network architecture defined in figure 4.7.3.1 and 4.7.3.2. EfficientNet was chosen as the CNN feature extractor as it was a lightweight model that is able to perform inference in real-time. Besides, EfficientNet is the state-of-the-art CNN that is able to achieve above 90% classification accuracy with the benefit of having lesser parameters to train. The variant of EfficientNet chosen for this project is EfficientNetB1.

The EfficientNetB1 model was acquired from TensorFlow. The top layer of the original EfficientNet was removed as transfer Learning will be adopted as the methodology of training. Hence, the weights of EfficientNet were initialized with imagenet weights.

In order to fine-tune the EfficientNet to extract features that are specific to palmprints, a custom top layer was built for this purpose. The architecture of the custom top layer is created as per figure 4.7.3.2. The custom top layer receives the features that are extracted from EfficientNet. Since the features are 3 high dimensional, global average pooling is used to summarise the features of each channel into a single value. The output of this summarization is a feature vector of length 1280. The reason behind the usage of global average pooling instead of a flatten layer is because the former has significantly lesser parameters to train, having lesser number of parameters means that the model is less prone to overfitting.

Afterwards, the features pass through a series dropout and dense layers to learn higher level features that are specific to palmprints. Dropout layers are used for regularization purposes to prevent overfitting from occurring, due to the large number of features learned. The custom top layer is then appended to the end of EfficientNet as a custom feature extractor model.

The defining part of a Siamese Network is the distance layer that connects and joins both embedding models together. L1 distance is used as the distance layer of the Siamese Network, the L1 distance computes the absolute difference between 2 feature vectors and returns a distance vector of the same length as the feature vector. The L1 distance is defined as a class that inherits the keras.layer class, this enables the class to be added to the model as a custom layer.

Moving on, a dense layer with 1 output neuron using sigmoid activation function is defined as the classifier. The Siamese model is then created by combining the embedding model, L1 distance layer and classifier. The figure below shows the Siamese Model summary implemented in Keras.

```
Model: "SiameseNetwork"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_img (InputLayer)         [(None, 224, 224, 3  0           []
                                )]

 validation_img (InputLayer)    [(None, 224, 224, 3  0           []
                                )]

 eff_net_embed (Functional)     (None, 1024)         11186263    ['input_img[0][0]',
                                                                  'validation_img[0][0]']

 distance (L1Dist)              (None, 1024)         0           ['eff_net_embed[0][0]',
                                                                  'eff_net_embed[1][0]']

 dense_2 (Dense)                (None, 1)            1025        ['distance[0][0]']

==================================================================================================
Total params: 11,187,288
Trainable params: 4,612,049
Non-trainable params: 6,575,239
__________________________________________________________________________________________________
```

Figure 4.7.5.1 Siamese Model Summary

Before beginnning model training, the dataset prepared earlier is loaded into a data loader pipeline. In the pipeline, the image undergoes several pre-processing operations such as image resizing and histogram equalization. Histogram equalization is done to smoothen and elevate the features of the palmprint by eliminating lighting distortions and clearing shadows on the palmprint image. Pixel normalization was not done as EfficientNet contains a normalization layer at the start of the network. The dataset was subsequently shuffled and divided into training and testing partitions with ratio of 7:3. The testing partition is set aside and awaited to be used in the model testing module. Then, the data is divided into batches of 20 as Mini-Batch Gradient Descent will used for training the model.

A low learning rate was chosen, as the goal of training was to fine-tune the custom top layer. Since the output of the model is to predict whether both input palm images match, hence this is considered as a binary classification task. Binary Cross Entropy loss was chosen as the loss function as it is most appropriate to be used for this task.

The optimizer used for model training is the Adam optimizer with a learning rate of 0.0001. The training ran for 50 epochs. In order to monitor the training metrics and the training memory footprint, a custom training loop was implemented instead using the .fit() method provided by Keras.

During training, the precision, recall, and loss were recorded each epoch. The model loss shows a steady decrease while precision and recall values were increasing, this indicated that the model was learning and was able to fit to the dataset. Figure below shows the metrics plotted as graphs.

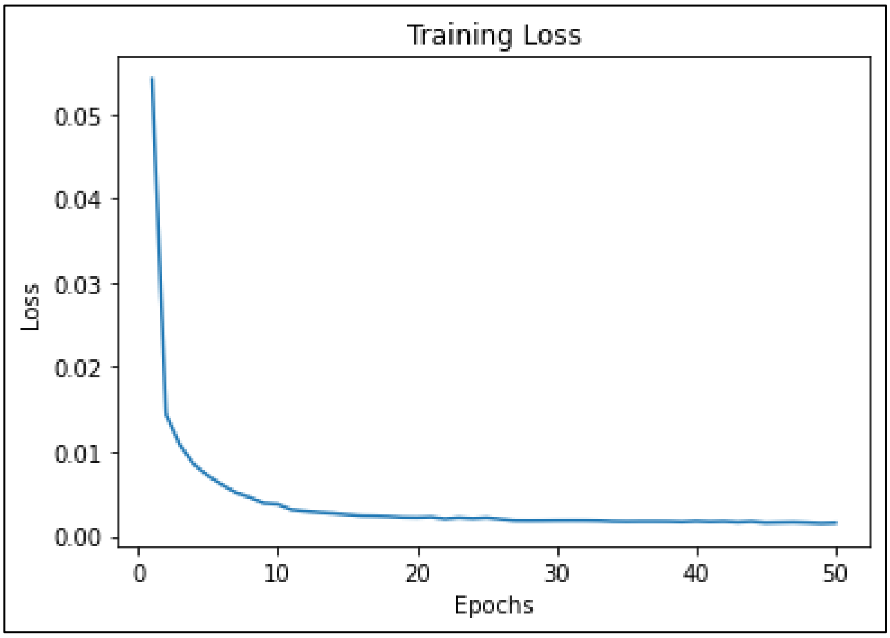


Figure 4.7.5.2 Model Training Loss

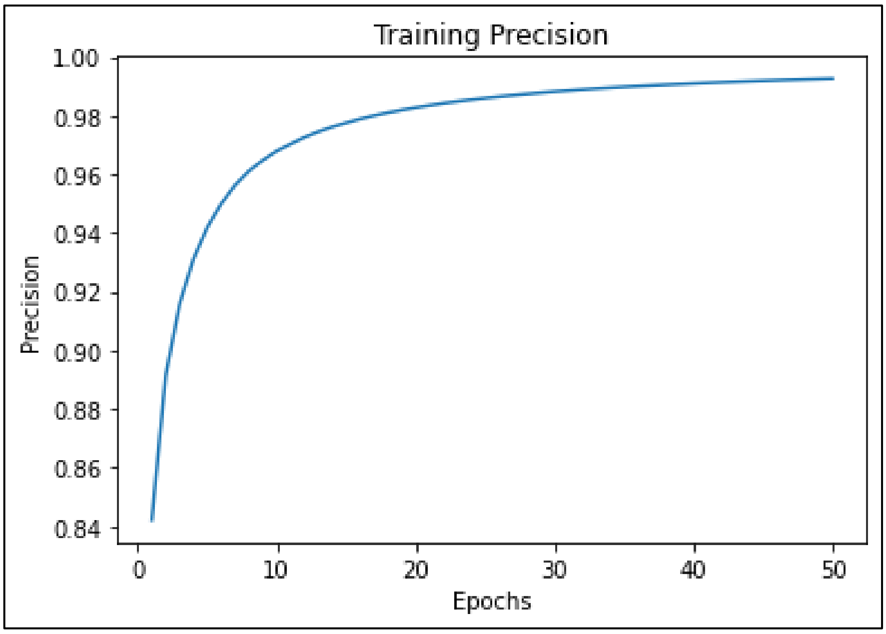


Figure 4.7.5.3 Model Training Precision

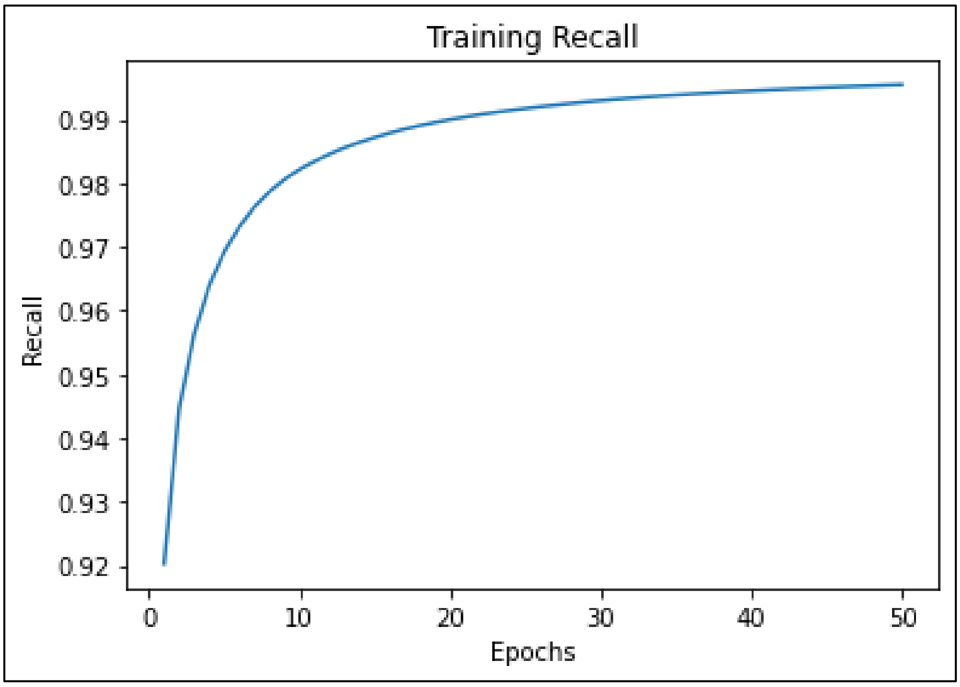


Figure 4.7.5.4 Model Training Recall

The model landed on a loss of 0.00006 after the 4-day training session. The trained model was then used for testing and evaluation purposes, which will be discussed in the upcoming chapter.

4.8 Model Testing and Evaluation Process

4.8.1 Overview

This section analyses the performance of the trained Siamese Model on unseen samples. The model will be evaluated using classification metrics which include precision and recall. The evaluated model will then be saved as a single h5 file and can be reloaded for other usage.

4.8.2 Block Diagram

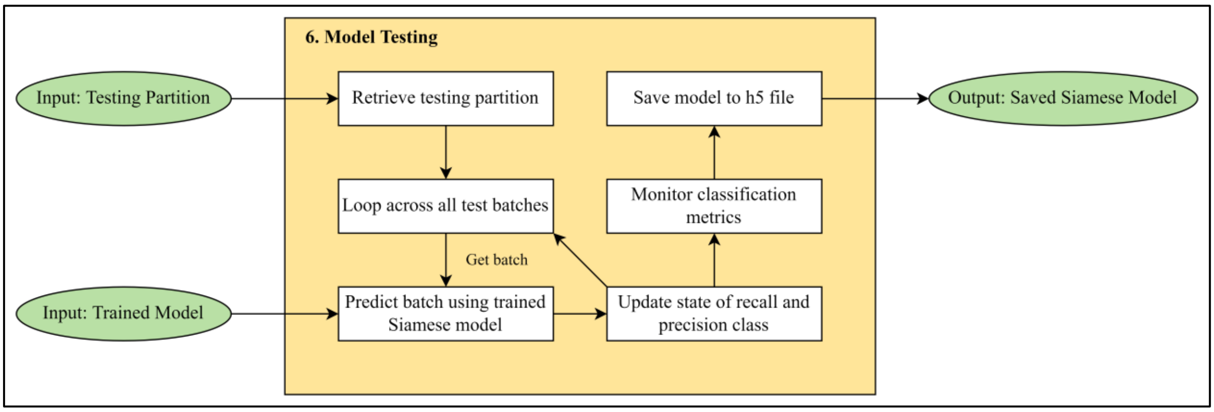


Figure 4.8.2.1 Block Diagram for Model Testing

4.8.3 Requirements

The outcome of this module is to save the trained and evaluated model into a single artifact file which can be deployed in the application to perform inference. Besides, the model may also be reloaded for future usage such as model retraining, etc.

4.8.4 Implementation Details

The inputs to this module are the testing partition of the dataset and also the trained Siamese model. The goal of this module is to evaluate whether the trained model is able to generalize well to unseen samples and whether it is fit for performing inference when deployed for real-world usage.

First off, the testing partition was retrieved. Since the dataset was previously batched during model training, hence the testing partition was also divided into batches of 20 samples. Subsequently, the testing batches was looped through, and each batch was predicted using the trained model. The prediction of each batch along with their ground truth labels are then used for calculating the precision and recall. The precision and recall retrieved were above 0.9 respectively, this indicated the model did not face any overfitting and was able to generalize well to untrained samples. The evaluation metrics are shown below.

Table 4.8.4.1 Classification Metrics Summary

	precision	recall	f1-score	support
Match	0.93	0.92	0.92	148968
Not Match	0.92	0.92	0.92	137892
Accuracy			0.92	286860
Macro Average	0.92	0.92	0.92	286860
Weighted Average	0.92	0.92	0.92	286860

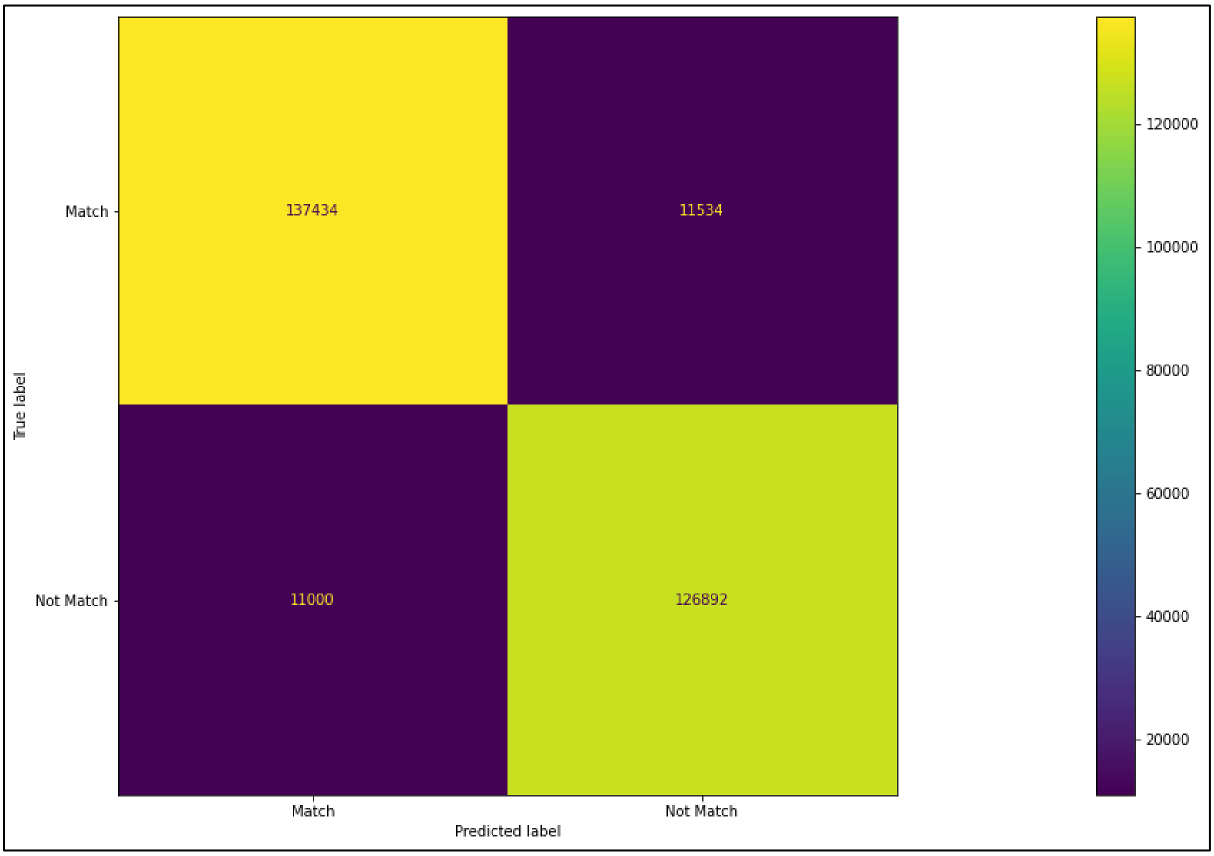


Figure 4.8.4.2 Confusion Matrix for model testing

Based on the classification metrics tabulated, the trained model was performing quite well as it was able to generalize well to unseen samples. The model was then saved into a h5 file by calling the model.save() method. The model was accepted for usage, as no overfitting was detected, and the performance of the model was very satisfactory.

4.9 Application Development and Model Integration

4.9.1 Overview

This section will walk through the implementation of the application that utilizes the trained Siamese model for inference. The application is implemented by translating the design models that was defined in Chapter 3. The application is designed using Object Oriented principles and implemented using MVC architecture pattern. Using MVC pattern ensures that changes to each individual class will not have cascading effects to the entire system.

The UI library that was used in developing the application is PyQT5. PyQT5 is a native Python UI development library that supports event-driven programming to add functionalities for the UI components on the application window.

The name of the developed application was named VeriPalm, as it combines the word verification and palm together.

4.9.2 Block Diagram

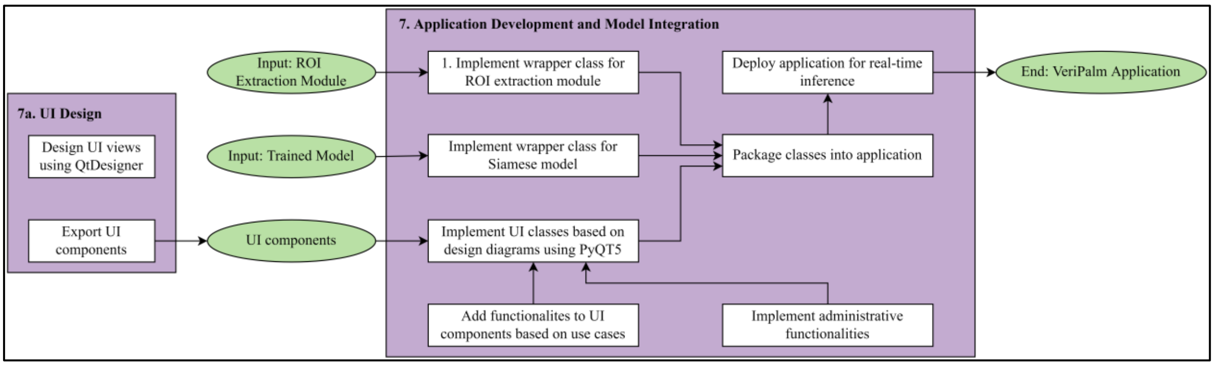


Figure 4.9.2.1 Block Diagram for Application Development

4.9.3 Requirements

The outcome of this module is to deliver an application that packages all the functionalities of the previous modules. The application is a desktop application that will utilize the trained model to perform registration and verification for palmprints. Besides, the application will utilize the palmprint ROI extraction module to extract ROI from palms that are placed in front of a camera device. Additionally, the application also supports administrative functions such as resetting and deleting profiles.

4.9.4 Implementation Details

The development of this module begins by creating wrapper classes for the ROI extraction module and the trained model. The wrapper classes are implemented in Python code using the previously defined class diagram.

The wrapper classes are created to provide public functions for the view component to use. For example, the wrapper class (SiameseNetUtil) for the Siamese model encapsulates all functionalities that the Siamese model provides. The view component may call the predict_pairs() function, and pass in both palm ROIs for verification. The underlying structure of predict_pairs() is using the keras.model.predict() method, however to increase the level of abstraction and hide unnecessary details, the predict_pairs() function is then written for this case. The SiameseNetUtil class also provides functionalities apart from model inference, such as preprocessing the image and reshaping. Additionally, the embedding model was extracted from the Siamese Model, to provide direct feature extraction on palmprint images for storage purposes.

Moving on, the ROI extraction module is then encapsulated into a class called MdpUtil that provides functions to extract the palmprint ROI from a video frame. The details of ROI extraction were discussed in the previous section.

The design class diagrams are defined in Chapter 3: Section 3.1.4

QtDesigner was used as the main tool for developing the UI components of the application. A total of 7 views were designed using the tool. The functionalities for each UI component are listed in the table below.

Table 4.9.4.1 Summary of UI views

Name	Window Type	Brief Functionality
admin.ui	Main Window	Admin menu page that links to the popup widgets
homepage.ui	Main Window	Default launch page that contains links to other windows
register.ui	Main Window	Registration page that allows for live palmprint registration and profile creation
verify.ui	Main Window	Verification page that allows for live verification with the stored palmprint embeddings in created profiles
direct_verify.ui	Popup Widget	Offline verification page that supports the upload of 2 palmprint ROIs and computes its similarity
edit_thresholds.ui	Popup Widget	Administration submenu that allows the admin to tweak various detection thresholds and determine the strictness of the verification during runtime.
manage_profile.ui	Popup Widget	Administration submenu that allows the admin to manage created and existing profiles.

The figure below shows the “register” page being created using QtDesigner.

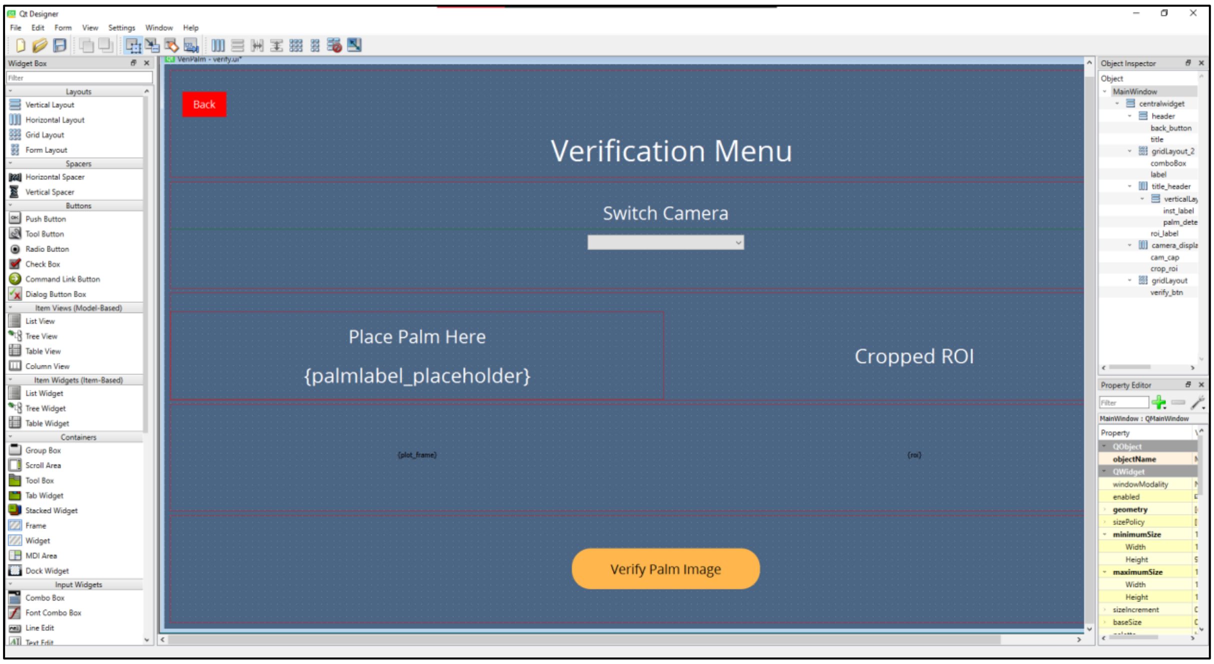


Figure 4.9.4.2 Using QtDesigner to design the verification page

After all the pages were designed using QtDesigner, the pages are then exported as UI files according to the file name defined in the table above. Using similar practices, a class was written to handle each UI. The UI classes will inherit the QMainWindow class for the menus, while the other UI classes will inherit the QWidget class. The UI files were loaded into their respective UI classes via the .loadui() method and specifying the filename. Then, the UI components on the page are accessible as members of the class.

Subsequently, each UI component on the page such as the buttons, drop down lists and etc were given functionality via event-driven programming in the code.

Besides, a class called CamThread was created to handle the webcam interfacing functionalities of the module in section 4.4. A QThread was used to interface with the webcam on a separate thread. Then, this thread resource was then provided to the Registration and Verifcation pages as it was required by those pages for using the webcam to retrieve the live video frames. The View classes will also utilize the functions from the MdpUtil class and SiameseNetUtil class to perform ROI extraction, model inference and other related operations.

Administrative functions were also provided in the VeriPalm app. The admin functions serve as a platform the admin to manage the registered profiles in the system. The admin menu also features a direct verification that allows for offline verification for palmprints. In other words, the admin may directly upload 2 palm images and use the model to predict whether both palm matches. It is worth noting that the admin views were implemented as QWidgets so that they can be opened as popout windows.

As effort to further optimize the user experience of the program, a thread was also used to load the Siamese model when the application launches since model loading is long and will block the application until it finishes loading. This ensures a smooth UI experience as the application UI can be displayed to the user first.

Finally, a launcher file was created to launch the program without having to go into the code to run certain functions. The flow of the program will be discussed in depth during the system testing chapter as test cases will be used to verify the completeness and correctness of the VeriPalm app.

4.10 Concluding Remark

All in all, the modules that make up the system was discussed in their respective subchapters. All the modules including palm print ROI extraction, and the trained Siamese Model, were then combined into a single application named VeriPalm that was implemented using Python and a GUI library called PyQT5. Furthermore, The VeriPalm application can perform verification using live palmprint images or via offline verification and provides administrative and profile registration using palmprints.

Chapter 5

System Evaluation and Discussion

This chapter is dedicated to discussing the testing procedures and frameworks utilized to evaluate the developed system. Performance definition and test cases for the system will be defined in this chapter. Additionally, the trained model will also be evaluated using classification metrics.

5.1 System Verification Plan

5.1.1 Overview

In this section, the test cases along with their expected results will be laid out to evaluate every aspect of the developed system. The test cases were noted during the system development and were evaluated together during the implementation stage.

5.1.2 Contactless Palm Capture via Camera

Table 5.1.2.1 Test Cases for Contactless Palm Capture

Id	Test Case	Expected Results
1	Camera interfacing code is executed	Captured video frame is displayed as video stream on the window.
2	User places palm in front of the camera	Captured video frame is able to display the palm on the window.
3	User removes palm in front of the camera	Captured video frame is updated and no palm will be displayed.

5.1.3 ROI Segmentation

Table 5.1.3.1 Test Cases for ROI Segmentation

Id	Test Case	Expected Results
1	Video frame with palm is passed to the segmentation module	Bounding box is drawn on the palm, and the ROI is extracted. The label indicating the detected palm is also displayed.
2	Video frame with palm, but the palm is not properly aligned and is titled at an angle	ROI will not be segmented, and a black image with “error” word is displayed.
3	Video frame with more than 1 palm	ROI will not be segmented, and a black image with “error” word is displayed.
4	Video frame with no palm is passed to the segmentation module.	ROI will not be segmented, and a black image with “error” word is displayed.

5.1.4 Siamese Model

Table 5.1.4.1 Test Cases for evaluating Siamese Model

Id	Test Case	Expected Results
1	2 exactly same palm images are inputted to the model. The input images are trained.	Model outputs “Match” and displays a matching confidence of over 90%
2	2 similar palm images are inputted to the model. The input images are trained.	Model outputs “Match” and displays a matching confidence of over 90%
3	2 exactly same palm images are inputted to the model. The input images are unseen.	Model outputs “Match” and displays a matching confidence of over 80%
4	2 similar palm images are inputted to the model. The input images are unseen.	Model outputs “Match” and displays a matching confidence of over 80%
5	2 dissimilar palm images are inputted to the model. The input images are trained.	Model outputs “Not-match” and displays a matching confidence of less than 5%

6	2 dissimilar palm images are inputted to the model. The input images are unseen.	Model outputs "Not-match" and displays a matching confidence of less than 10%
7	A palm image is inputted to the embedding model	The model outputs a feature vector of length 128

5.1.5 VeriPalm App

Table 5.1.5.1 VeriPalm App Launcher and Main Menu

Id	Test Case	Expected Results
1	VeriPalm app is launched using the launch.py	The application loads up and displays the main menu window.
2	VeriPalm is launched for first time, and there are no existing profiles in the system.	The verification button on the main menu is locked and "No Profiles" is display.
3	"Register Palmprint" button is clicked.	The window switches to the registration page.
4	"Admin console" button is clicked.	An input dialog pops up and an admin password is required for access.
5	Wrong admin password is entered	An "Incorrect Password" dialog popups and access is prevented.
6	Correct admin password is entered	The window switches to the admin menu page.
7	"Exit App" button is clicked	The main menu window closes, and the application terminates.

Table 5.1.5.2 Registration and Verification General Cases

Id	Test Case	Expected Results
1	“Back” button is clicked	The registration page is exited and the window switches back to the main menu.
2	The “Switch Camera” dropdown list is selected.	All available cameras that are detected, are filled to the list.
3	Another camera device is selected from the “Switch Camera” dropdown list	The camera displayTabl feed is changed to the new camera’s video stream.
4	Observe the camera display feed	The camera display feed displays what is being captured.
5	A palm is placed in front of the camera as per the “Match” cases in table 5.1.3.1	• The camera feed will display the palm with the landmarks plotted, and the ROI bounding box. • The extracted ROI is displayed in the cropped ROI feed. • The detected palm label is updated according to the palm detected. Possible detections are “Left” or “Right” and highlighted green.
6	No palm or invalid object is placed in front of the camera	• The camera feed will display the background captured. • Error image will be displayed in the cropped ROI feed. • The detected palm label is updated to “No Palm Detected” and highlighted red.

Table 5.1.5.3 Registration Functions

Id	Test Case	Expected Results
1	User clicks on the "Register Palm Image" button with an invalid palm present in front of the camera.	The "Register Palm Image" button is locked, and registration is prevented.
2	User clicks on the "Register Palm Image" button with valid palm present in front of the camera.	The "Register Palm Image" button is unlocked, and the registration process begins.
3	First time user registers a palm	User acknowledges the creation of new profile and is required to enter a username. The ROI embeddings is added to the newly created profile, and a profile password is created and displayed to the user for future use. "Palm Registered" is displayed. User declines the profile creation. The registration process is aborted and return to the registration page.
4	Returning user registers a palm and the entered password matches with the password stored in the profile.	The ROI embeddings is appended to the existing profile. "Palm has been registered successfully!" is displayed.
5	Returning user registers a palm but entered the wrong password.	Display "Password does not match" and abort the process.
6	User registers palm with strict registration mode on	The current palm image is similar to the previously registered palm images. The current palm image is allowed to register. The current palm image is not similar to the previously registered palm images. Process is aborted

Table 5.1.5.4 Verification Functions

Id	Test Case	Expected Results
1	User clicks on the “Verify Palm Image” button with an invalid palm present in front of the camera.	The “Verify Palm Image” button is locked, and registration is prevented.
2	User clicks on the “Verify Palm Image” button with valid palm present in front of the camera.	The “Verify Palm Image” button is unlocked, and the verification process begins.
3	Verification process	The detected ROI is compared with all stored embeddings in all profiles. The profile name with the highest matches is displayed along with the confidence value and clock in time. No valid profiles pass the matching threshold, “Unrecognized palm. Please try again” is displayed.

Table 5.1.5.5 Admin Functions

Id	Test Case	Expected Results
1	Admin clicks on the “Manage Registered Profiles” button	The profile management widget pops up.
2	Admin clicks on the “Direct Verification” button	The direct verification widget pops up.
3	Admin clicks on the “App Settings” button	The app settings widget pops up.
4	“Back” button is clicked	The admin menu page is exited and the window switches back to the main menu.

Table 5.1.5.6 Manage Registered Profiles Functions

Id	Test Case	Expected Results
1	Profile Management popup is displayed	• All registered profiles are displayed in the table • "No existing profiles found" is displayed when no existing profiles exist
2	Admin clicks on the "Reset" button associated with the profile	System requests for confirmation on reset the embeddings stored on the profile.
3	Admin clicks on the "Delete" button associated with the profile	System requests for confirmation on the deletion of the profile. The table view is updated based on the selection.
4	"Back" button is clicked	The current popup is exited.

Table 5.1.5.7 Direct Verification Functions

Id	Test Case	Expected Results
1	Direct Verification popup loads up	The verification status and confidence labels are set to pending result.
2	Admin clicks on the "Verify Palm Images" button with no images uploaded or only 1 image uploaded	The "Verify Palm Images" button is locked, and verification cannot be done with no image or 1 image only.
3	Admin properly uploads 2 palm images via the "Upload Image 1" and "Upload Image 2" buttons, and then clicks the "Verify Palm Images"	The verification process begins, and the result of the match is displayed in the verification status and verification confidence labels Verification status will be updated to "Not Matched" and highlighted red for dissimilar palm images

		Verification status will be updated to “Matched” and highlighted green for similar palm images.
4	“Back” button is clicked	The current popup is exited.

Table 5.1.5.8 App Settings (Edit Threshold) Functions

Id	Test Case	Expected Results
1	Admin tunes the profile detection threshold	The profile detection threshold is updated and will affect the verification process.
2	Admin tunes the average confidence threshold	The average confidence threshold is updated and will affect the verification process.
3	Admin tunes the average confidence threshold	The matching threshold is updated and will affect the verification process.
4	Admin checks/unchecks the strict registration mode	The registration mode is updated and will affect the registration process.
5	“Close” button is clicked	The current popup is exited.

5.2 Testing Setup and Results

5.2.1 Overview

In this section, the test cases will be applied to determine if the developed modules meet the standard performance and requirements set.

5.2.2 Contactless Palm Capture via Camera

Table 5.2.2.1 Test Case: Testing Camera Input for Capturing Palm

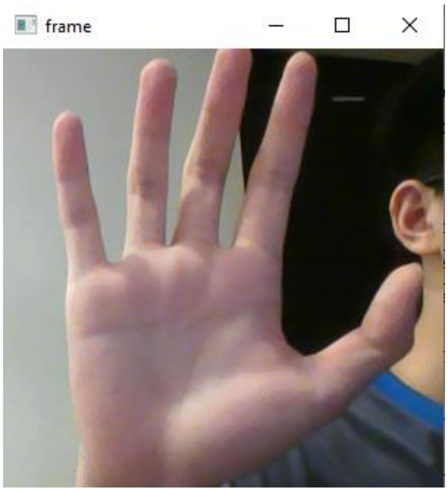
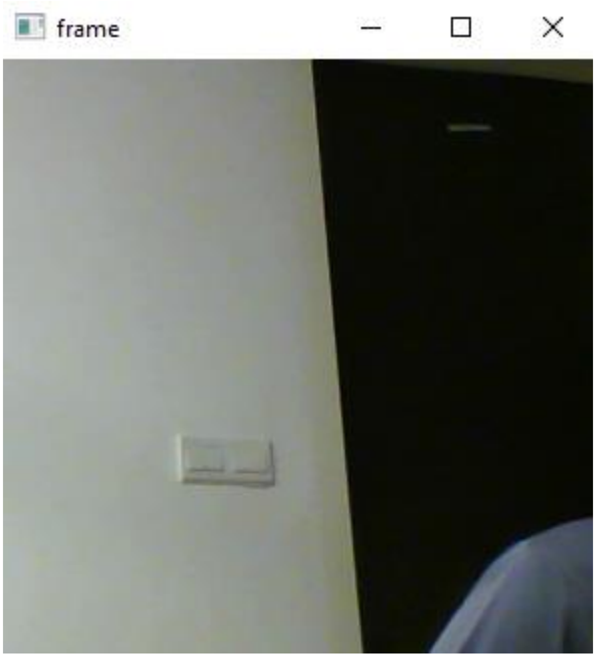
Test Case Name	User places palm in front of the camera
Test Case Description	User places their palm in front of the camera and will be streamed live
Expected Output	Captured video frame is able to display the palm on the window
Input	
Results	
Status (Pass/Fail)	Pass

Table 5.2.2.2 Test Case: Testing Camera Input without Palm

Test Case Name	User removes palm in front of the camera
Test Case Description	User removes their palm in front of the camera.
Expected Output	Captured video frame is updated and no palm will be displayed.
Input	
Results	
Status (Pass/Fail)	Pass

5.2.3 ROI Segmentation

Table 5.2.3.1 Test Case: Testing ROI Segmentation for palms

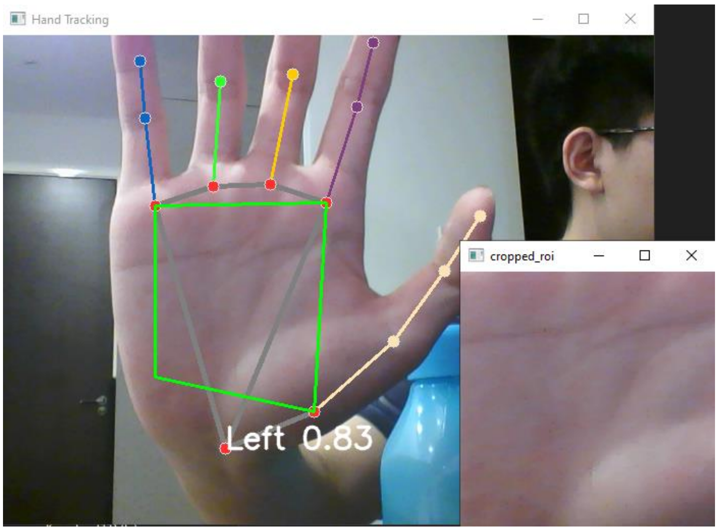
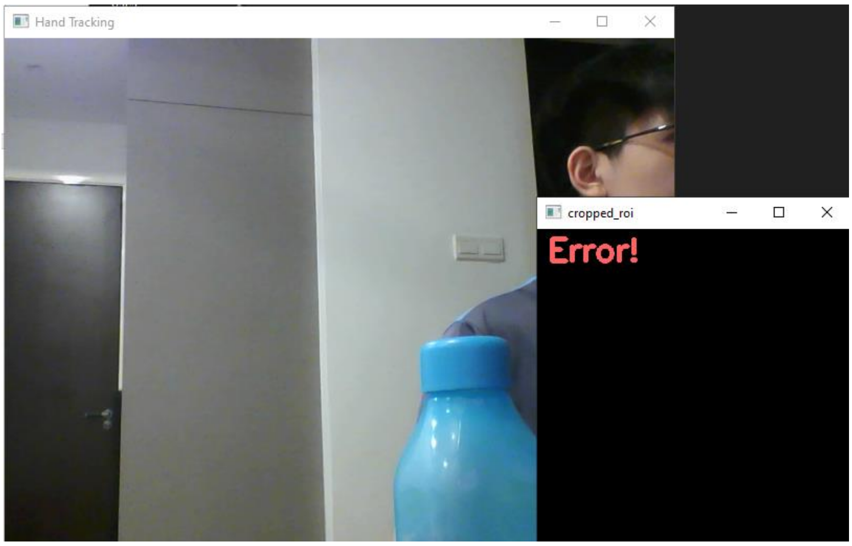
Test Case Name	Video frame with palm is passed to the segmentation module
Test Case Description	A video frame containing the palm image will be inputted for ROI segmentation
Expected Output	Bounding box is drawn on the palm, and the ROI is extracted. The label indicating the detected palm is also displayed.
Input	
Results	
Status (Pass/Fail)	Pass

Table 5.2.3.2 Test Case: Testing ROI Segmentation for invalid objects

Test Case Name	Video frame with invalid objects passed to the segmentation module
Test Case Description	A video frame containing the objects other than palm images will be inputted for ROI segmentation
Expected Output	ROI will not be segmented, and a black image with "error" word is displayed.
Input	
Results	
Status (Pass/Fail)	Pass

5.2.4 Siamese Model

Table 5.2.4.1 Test Case: Verifying 2 similar palm images

Test Case Name	Two similar palm images are inputted to the model
Test Case Description	Two prepared and similar ROIs are used to verify the model's predictive performance
Expected Output	Model outputs "Match" and displays a matching confidence of over 90%
Input	Image 1:  Image 2: 
Results	Match Matching confidence: 0.9799238443374634
Status (Pass/Fail)	Pass

Table 5.2.4.2 Test Case: Verifying 2 dissimilar palm images

Test Case Name	Two dissimilar same palm images are inputted to the model
Test Case Description	Two prepared and dissimilar ROIs are used to verify the model's predictive performance
Expected Output	Model outputs "Not-match" and displays a matching confidence of less than 10%
Input	Image 1:  Image 2: 
Results	No Match Matching confidence: 7.218816790555138e-06
Status (Pass/Fail)	Pass

Table 5.2.4.3 Test Case: Extracting features from palm image

Test Case Name	A palm image is inputted to the embedding model
Test Case Description	The Siamese Model's embedding model is used to extract the features from a palm image
Expected Output	The model outputs a feature vector of length 128
Input	Input Image: 
Results	`Embedding shape: (1, 128)`
Status (Pass/Fail)	Pass

5.2.5 VeriPalm App

Table 5.2.5.1 Test Case: Launch App

Test Case Name	Launch App
Test Case Description	The VeriPalm application is launched using the launcher
Expected Output	The main menu is displayed
Input	Launch Command: `>python launch.py_`
Results	
Status (Pass/Fail)	Pass

Table 5.2.5.2 Test Case: Access registration page

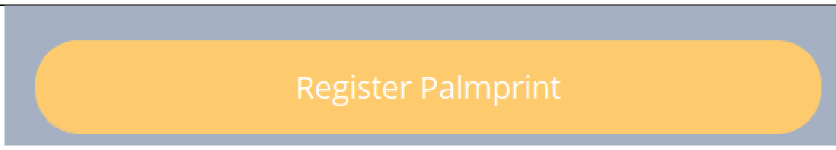
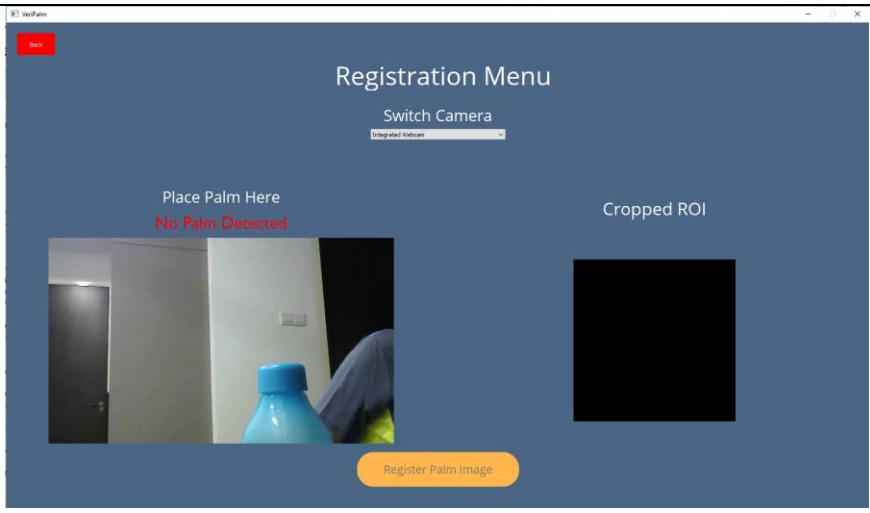
<table>
<tr><th>Test Case Name</th><td>Access registration page</td></tr>
<tr><th>Test Case Description</th><td>The user accesses the registration page from main menu</td></tr>
<tr><th>Expected Output</th><td>The registration page is displayed</td></tr>
<tr><th>Input</th><td>
</td></tr>
<tr><th>Results</th><td>
</td></tr>
<tr><th>Status (Pass/Fail)</th><td>Pass</td></tr>
</table>

Table 5.2.5.3 Test Case: Access verification page

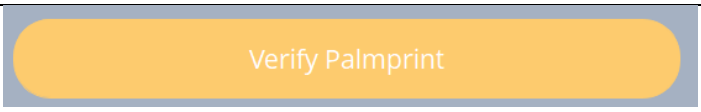
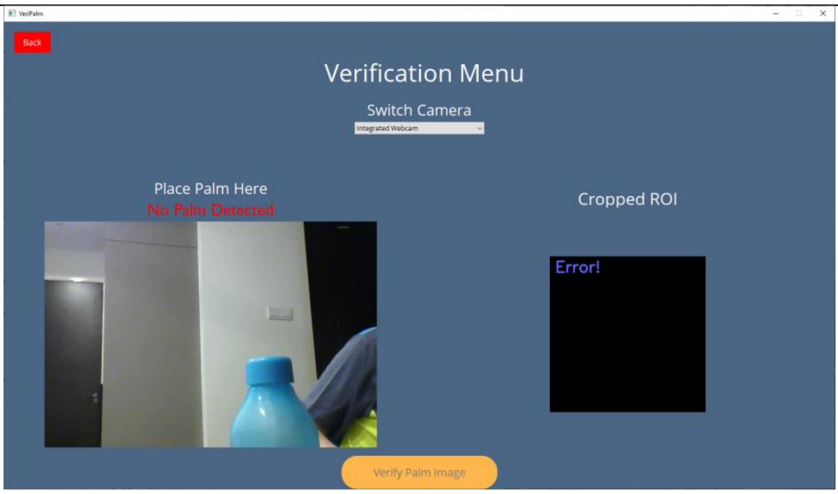
Test Case Name	Access verification page
Test Case Description	The user accesses the verification page from main menu.
Expected Output	The verification page is displayed
Input	
Results	
Status (Pass/Fail)	Pass

Table 5.2.5.4 Test Case: Access admin console page

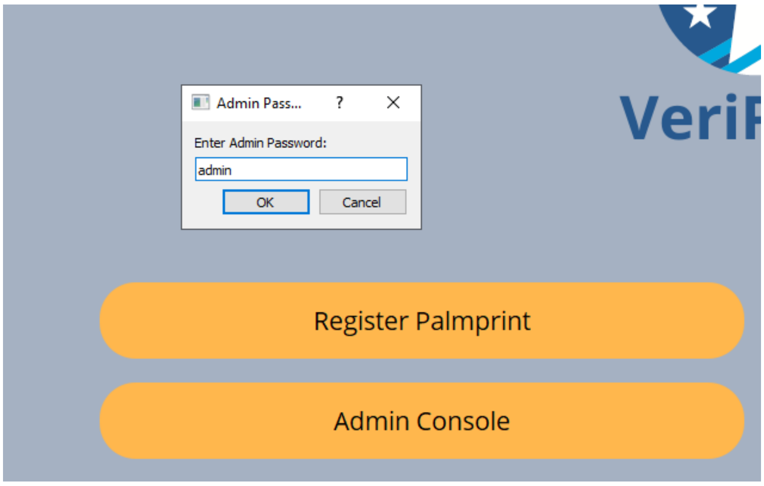
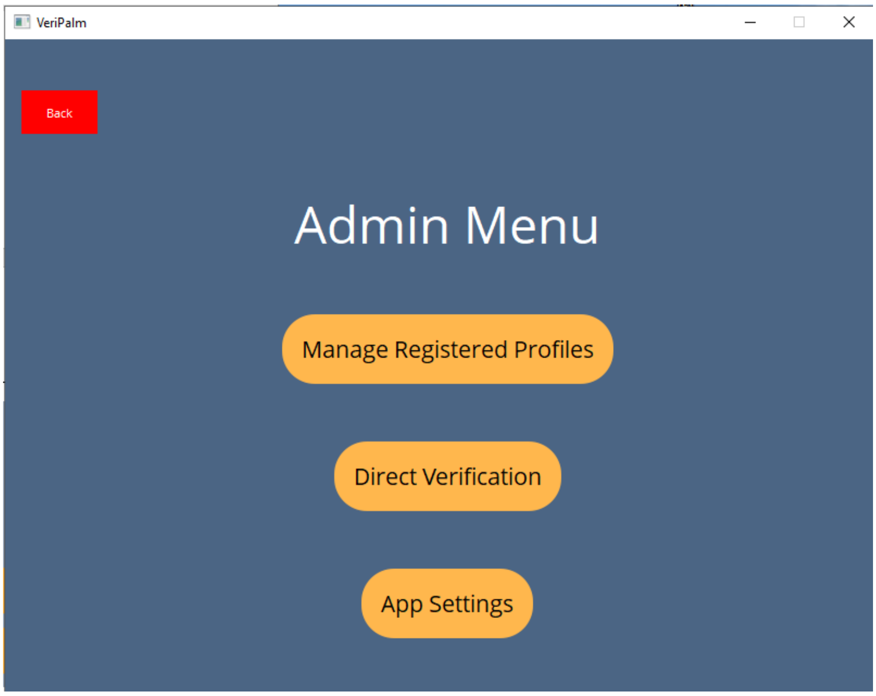
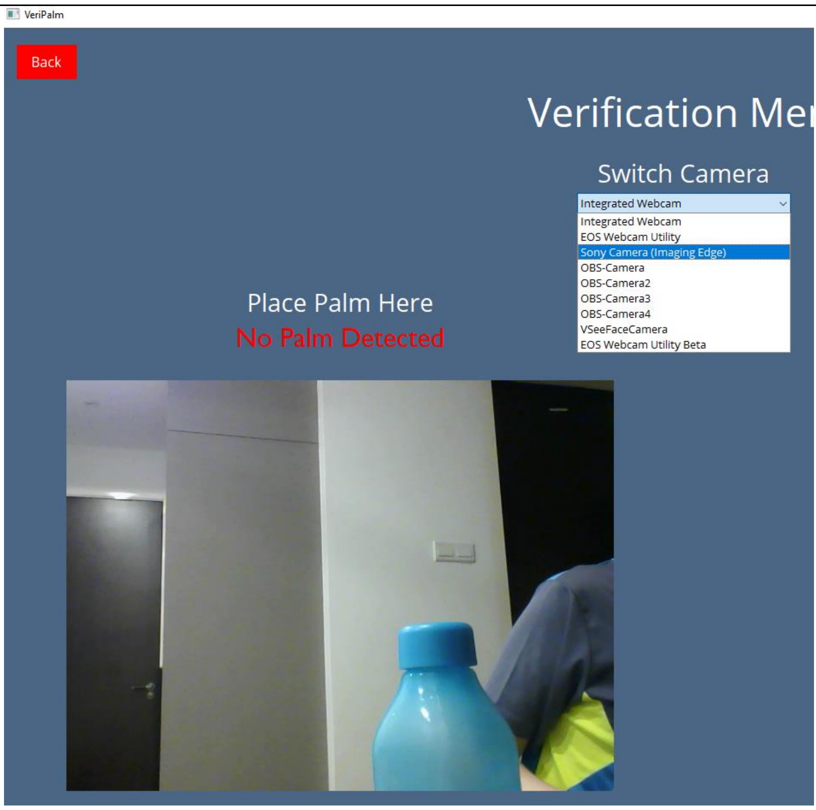
Test Case Name	Access admin console page
Test Case Description	The user accesses the admin console from main menu and provides a valid password.
Expected Output	The admin console is displayed
Input	
Results	
Status (Pass/Fail)	Pass

Table 5.2.5.5 Test Case: Switching camera in registration and verification page

Test Case Name	Switch camera from dropdown list
Test Case Description	The user selects and changes to an available camera from the dropdown list.
Expected Output	The camera feed switches from the default camera to another camera
Input	

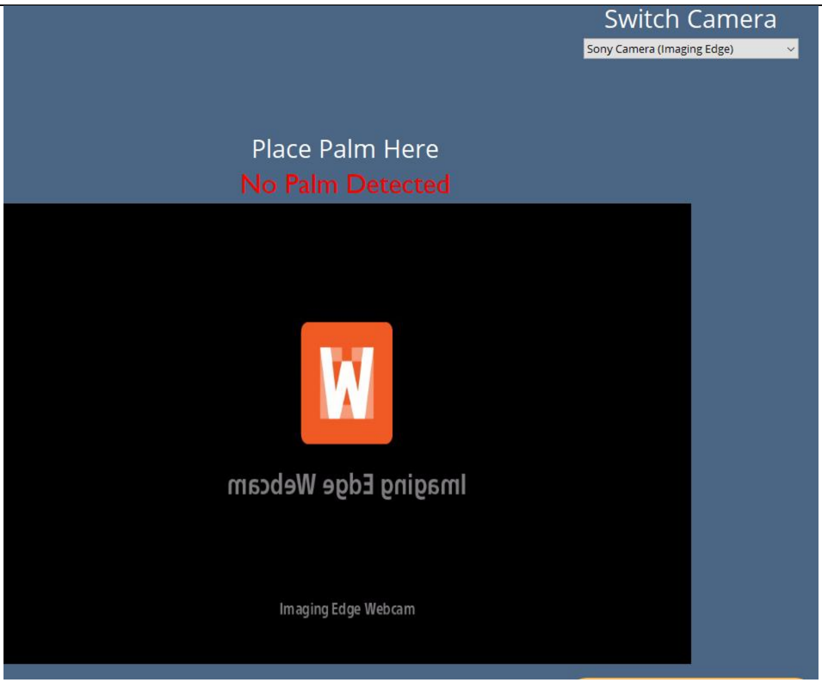
Results	
Status (Pass/Fail)	Pass

Table 5.2.5.6 Test Case: Testing valid ROI Segmentation in VeriPalm app

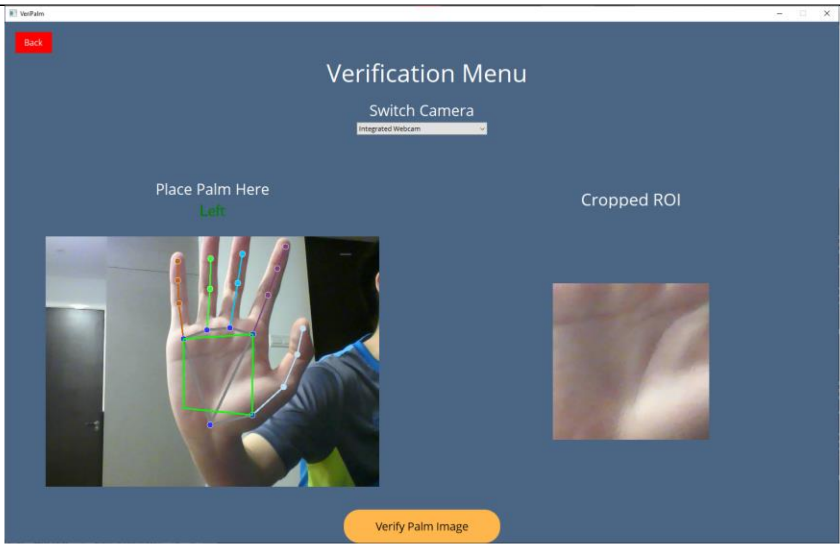
Test Case Name	Testing valid ROI Segmentation in VeriPalm app
Test Case Description	A palm is placed in front of the camera
Expected Output	• The camera feed will display the palm with the landmarks plotted, and the ROI bounding box. • The extracted ROI is displayed in the cropped ROI feed. • The detected palm label is updated according to the palm detected. Possible detections are “Left” or “Right” and highlighted green.
Input	
Results	
Status (Pass/Fail)	Pass

Table 5.2.5.7 Test Case: Testing invalid ROI Segmentation in VeriPalm app

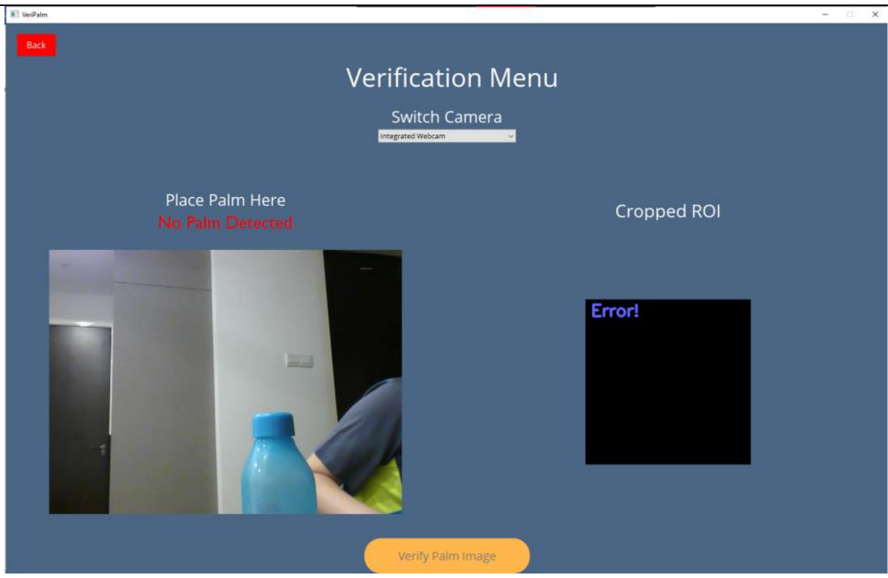
Test Case Name	Testing invalid ROI Segmentation in VeriPalm app
Test Case Description	No palm is placed in front of the camera
Expected Output	• The camera feed will display the background captured. • Error image will be displayed in the cropped ROI feed. • The detected palm label is updated to “No Palm Detected” and highlighted red.
Input	
Results	
Status (Pass/Fail)	Pass

Table 5.2.5.8 Test Case: Registering palm for a new user

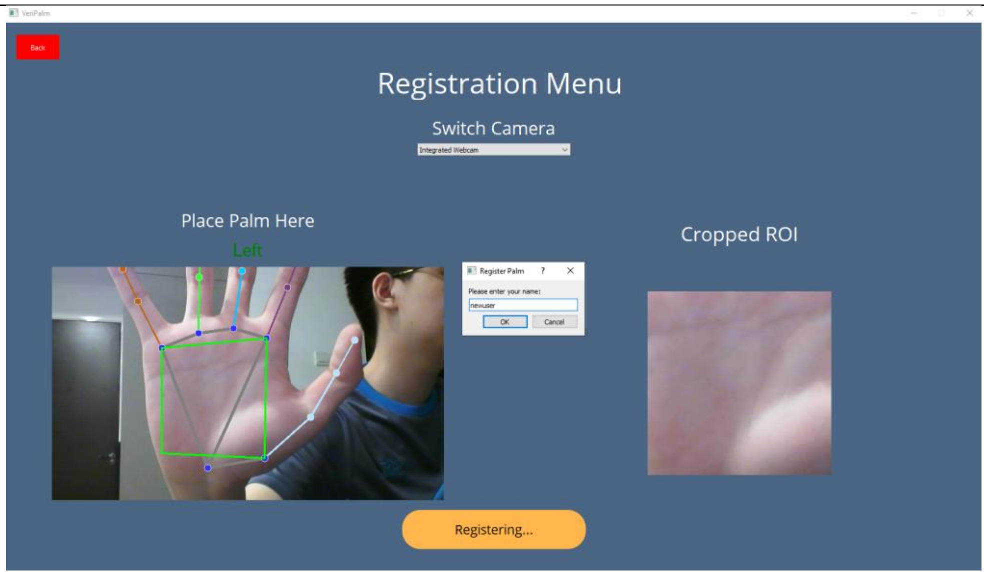
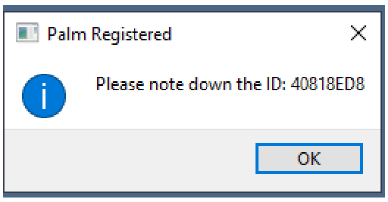
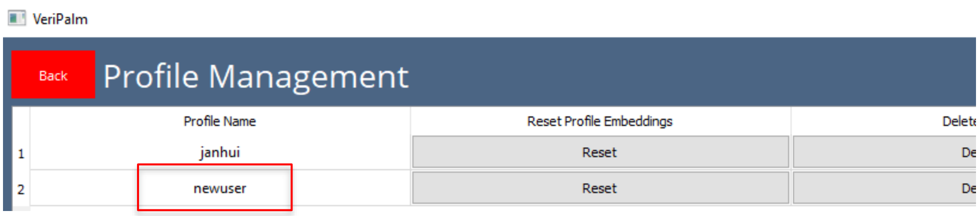
Test Case Name	Registering palm for a new user
Test Case Description	A new user to the system registers their palm into the system
Expected Output	User acknowledges the creation of new profile and is required to enter a username. The ROI embeddings is added to the newly created profile, and a profile password is created and displayed to the user for future use. "Palm Registered" is displayed. New profile visible in profile management.
Input	
Results	  
Status (Pass/Fail)	Pass

Table 5.2.5.9 Test Case: Registering palm for returning user

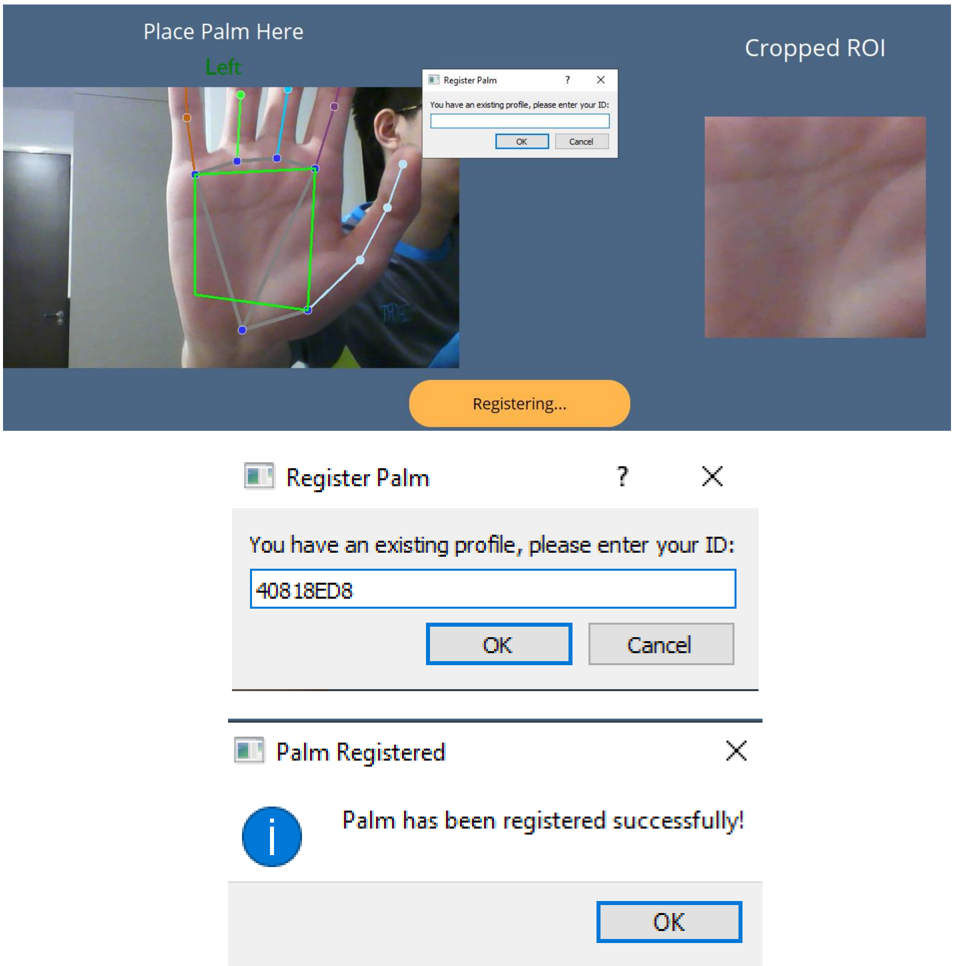
Test Case Name	Registering palm for returning user
Test Case Description	Returning user registers a palm and the entered password matches with the password stored in the profile.
Expected Output	The ROI embeddings is appended to the existing profile. “Palm has been registered successfully!” is displayed.
Input	
Results	
Status (Pass/Fail)	Pass

Table 5.2.5.10 Test Case: Live verification for palms

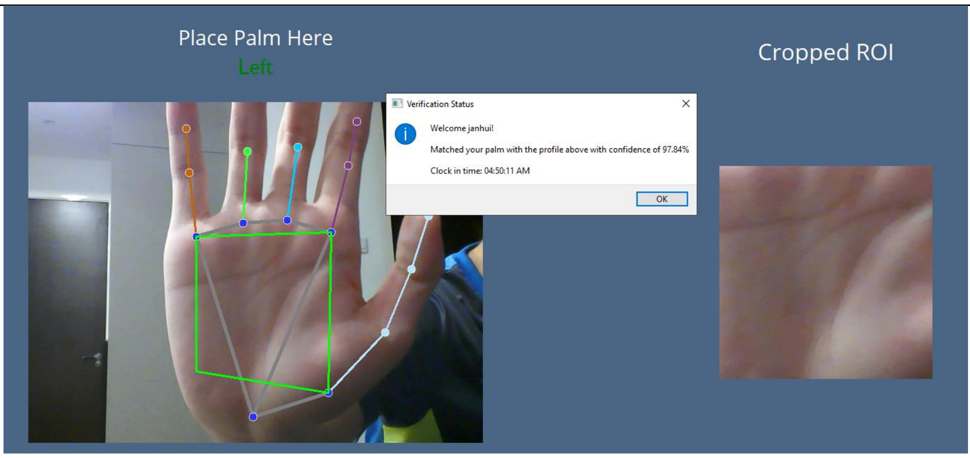
Test Case Name	Live verification for palms
Test Case Description	The detected ROI is compared against all stored ROI embeddings of all profiles.
Expected Output	The profile name with the highest matches is displayed along with the confidence value and clock in time. No valid profiles pass the matching threshold, “Unrecognized palm. Please try again” is displayed.
Input	
Results	
Status (Pass/Fail)	Pass

Table 5.2.5.11 Test Case: Reset Profiles in Profile Management

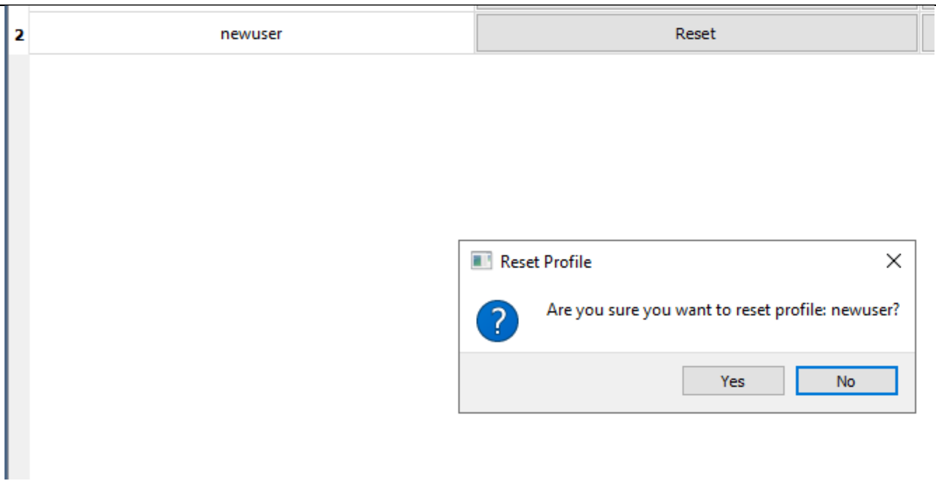
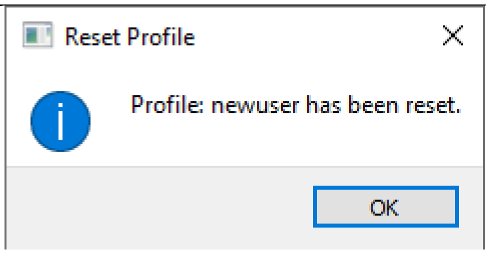
Test Case Name	Reset Profiles in Profile Management
Test Case Description	Admin resets the embeddings stored in a profile
Expected Output	System requests for confirmation on reset the embeddings stored on the profile and prompts a success message upon completing.
Input	
Results	
Status (Pass/Fail)	Pass

Table 5.2.5.12 Test Case: Delete Profiles in Profile Management

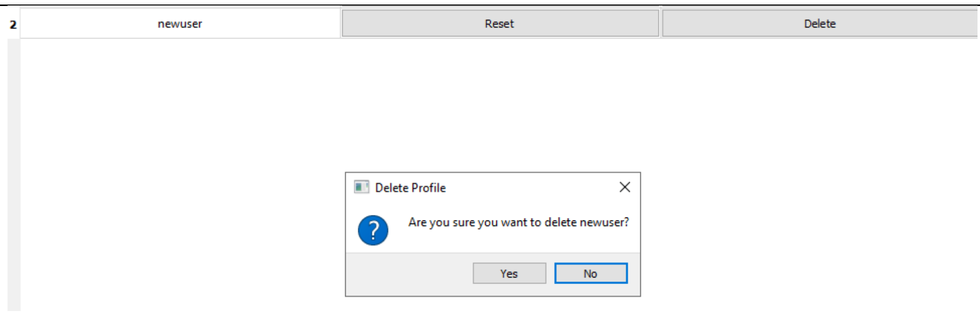
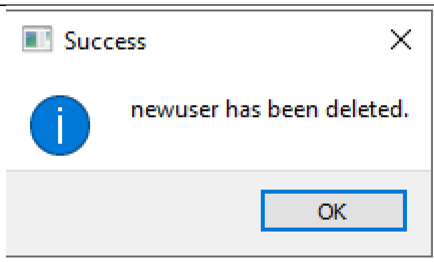
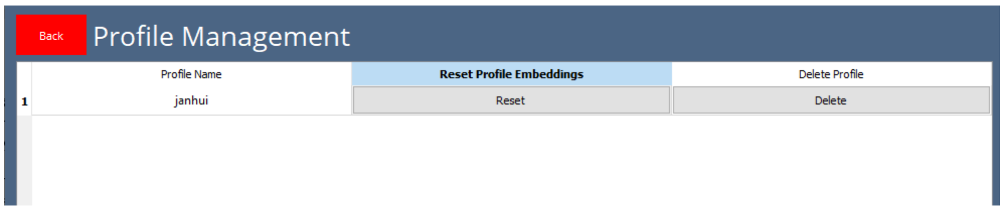
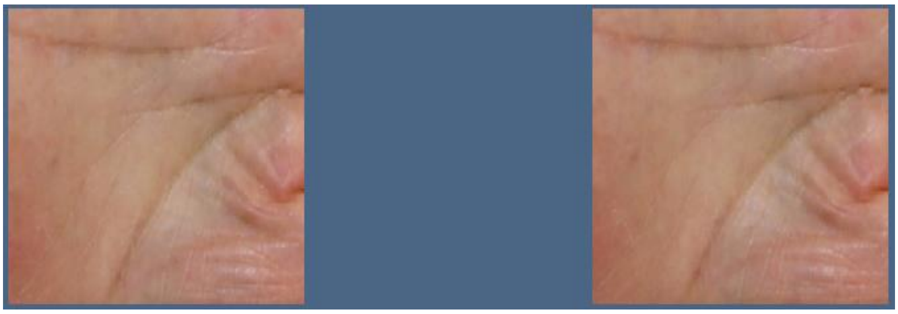
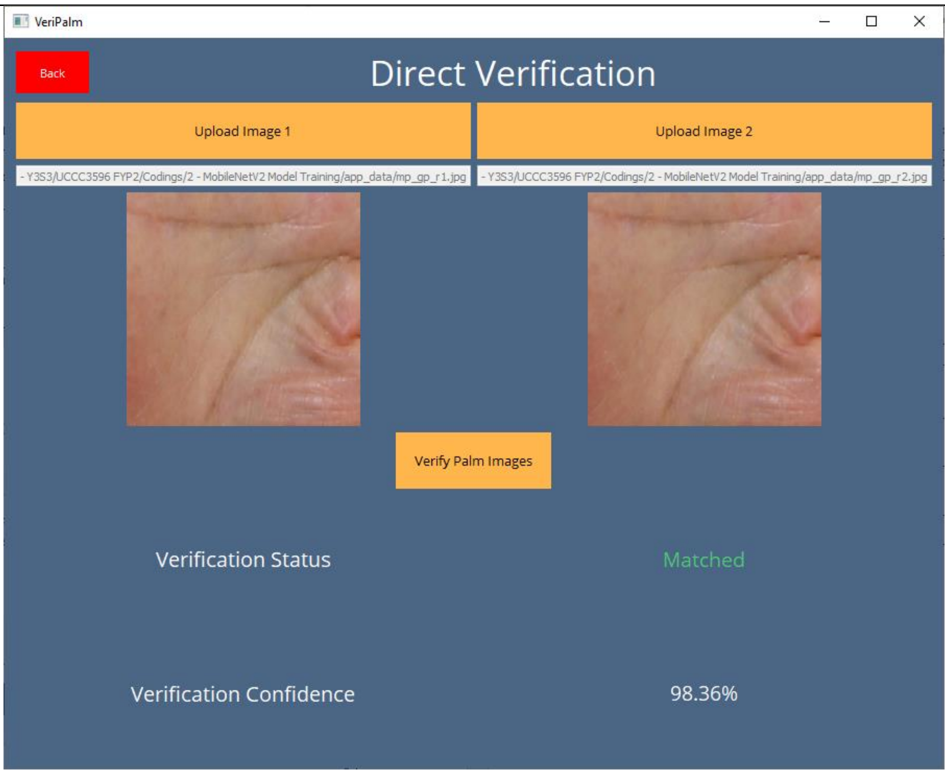
Test Case Name	Delete Profiles in Profile Management
Test Case Description	Admin deletes a registered profile from the system
Expected Output	System requests for confirmation on the deletion of the profile. The table view is updated based on the selection.
Input	
Results	
Status (Pass/Fail)	Pass

Table 5.2.5.13 Test Case: Offline-direct verification

Test Case Name	Offline-direct verification of palm images
Test Case Description	Admin directly upload two palm images for verification
Expected Output	• The result of the match is displayed in the verification status and verification confidence labels • Verification status will be updated to "Not Matched" and highlighted red for dissimilar palm images • Verification status will be updated to "Matched" and highlighted green for similar palm images.
Input	Similar Image Pairs:  Dissimilar Image Pairs: 

<table>
<tr><td>Results</td><td>

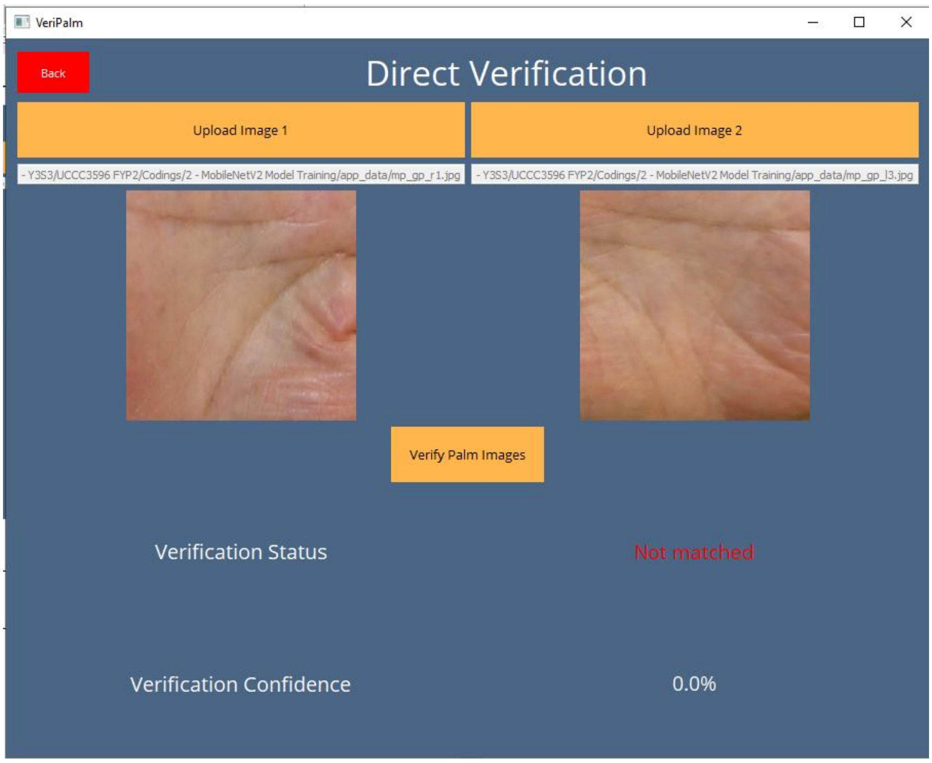
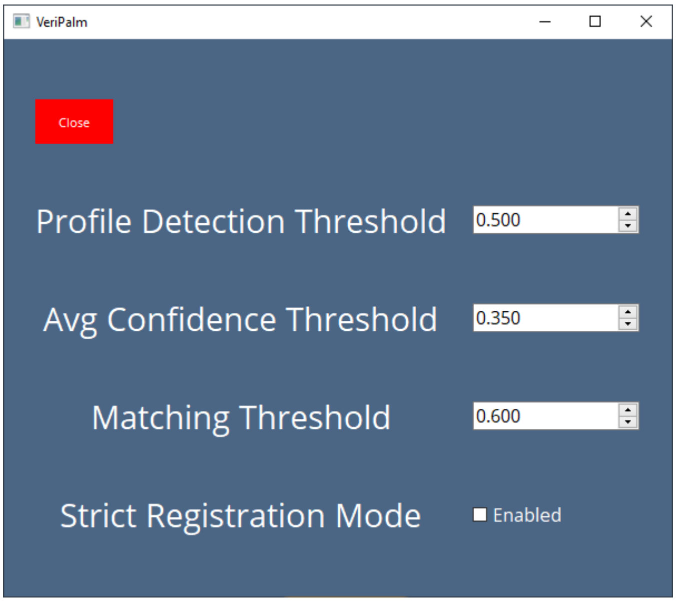
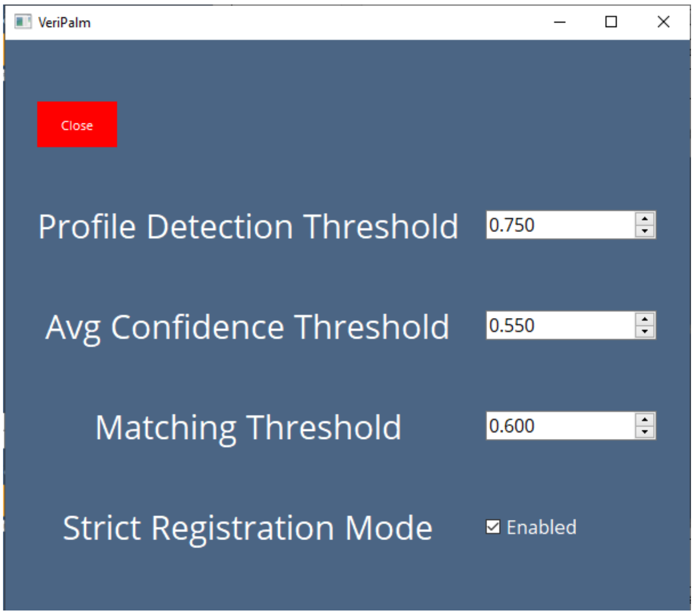

</td></tr>
<tr><td>Status (Pass/Fail)</td><td>Pass</td></tr>
</table>

Table 5.2.5.14 Test Case: Updating model detection thresholds

Test Case Name	Updating model detection thresholds
Test Case Description	Admin updates the threshold values
Expected Output	The threshold values are updated and will affect the training process
Input	Initial Threshold Settings 
Results	Tuned to a stricter setting 
Status (Pass/Fail)	Pass

5.3 Project Challenges

The primary challenge faced in this project revolves around the performance of the model. It is noted that the Siamese Model was retrained at least 10 times, and more data was added to each retraining session in order to increase its performance. Besides, multiple CNN architectures ranging from complex models like InceptionResNetV2, VGG16 to lightweight models like MobileNetV2 were experimented to find the best feature extractor for palmprints. However, in the end, the final Siamese model that was trained using the samples defined in Chapter 4 could perform well on predicting unseen samples.

The key takeaway from the issues is that state-of-the-art pretrained models do not necessarily translate to top notch performance, when being used for transfer learning on another dataset. The most important aspect is to acquire enough training samples so that the model is able to learn to extract better features and make an accurate prediction.

Besides that, the development of the ROI extraction module was also a challenge to this project. The previously developed ROI extraction module using the Double Line Single Point (DLSP) algorithm inspired by [5], proven to be not robust as the graphical assistants would produce palmprint ROIs of odd dimensions due to the varying camera settings used. Additionally, the DLSP algorithm did not have any form of liveness detection, and random background images could be segmented and cause an error.

To curb the issues, the ROI segmentation algorithm was overhauled and replaced by using MediaPipe Hands to do it. This overhaul process took around 2 weeks of development and had caused overhead time to begin development of other modules. However, the process was enriching and rewarding, as a novel ROI extraction algorithm was developed. The new ROI segmentation algorithm could work very well in varying conditions and are also smart enough to only recognize palms and no other objects.

5.4 Objectives Evaluation

All the objectives that were defined in Chapter 1 was successfully achieved. As a recap, the process of registering or verifying a palmprint was fully contactless as a camera device was set up to capture high resolution palm images.

Besides, a robust ROI extraction algorithm using MediaPipe Hands, was also developed and could extract square shaped ROIs from the palmprints, as long as it was detected in the video frame. Moreover, the ROI extraction algorithm was also used to extract ROIs from still palmprint images to be used as training samples. This move had helped during model training tremendously as the model benefited from having more unique training samples.

Suitable palmprint databases were also collected and reengineered to form a labelled dataset with palmprint image pairs.

Additionally, a Siamese Neural Network that performs verification via computing the distance vector between 2 feature embeddings of 2 palms was implemented. The Siamese Neural Network could generalize well to unseen samples as it did not learn a predetermined number of palms, but instead learned to extract the features that will determine whether both input palms match. The EfficientNet CNN was also fine-tuned, as a custom top layer was built to learn features that are agnostic to palmprints. The Siamese Neural Network had undergone rigorous evaluation by analysing its classification metrics to determine whether the model is fit to be accepted for application usage. In the end, the model was also saved into a single artifact.

Finally, the last objective, which is to develop an application that utilizes all developed modules was achieved. An application named VeriPalm that contains a wide array of functionalities as stated in the subobjectives was delivered.

5.5 Concluding Remark

All in all, the system was complete and had delivered all the requirements and objectives set for it. Furthermore, each aspect of every module had passed the test cases that were set for it. Having undergo rigorous testing and validation, this evidenced that the system was robust and is fit for deployment for small scale authentication usage scenarios where hygiene is of utmost concern.

Chapter 6
Conclusion and Recommendation

6.1 Conclusion

As a concluding statement, the development of this FYP was an enriching journey. Knowledge from multiple fields of study such as Image Processing, Data Science, Deep Learning, Computer Vision, and Application Development were explored and gained. Ultimately, each aspect of study was implemented as different modules in the system. The modules were then integrated into a single application that could perform verification and registration of palmprints in real-time.

This project contributed an effort towards the study of biometrics authentication using palmprints and had proposed a novel ROI extraction algorithm using MediaPipe Hands.

6.2 Future Work

The future work proposed is to experiment with different loss functions that are better suited to train Siamese Networks for verification purposes. For example, contrastive loss can be considered as it is a loss function that is able to leverage label information better than Binary Cross Entropy. Contrastive loss can extract particular features to allow image of the same class to be closer in the feature space, and further apart for images of different classes.

Besides, a triplet network can also be considered as future work. The triplet network is a natural extension to the Siamese Network, where image triplets are trained using the triplet loss rather than using image pairs.

BIBLIOGRAPHY

[1] A. Poinsot, M. Paindavoine and F. Yang, "Small Sample Biometric Recognition Based on Palmprint and Face Fusion," Cannes, 2009.

[2] M. Aykut and M. Ekinci, "Developing a Contactless Palmprint Authentication System by Introducing a Novel ROI Extraction Method," *Image and Vision Computing,* vol. 40, pp. 65-74, 2015.

[3] A. Morales, M. A. Ferrer, C. M. Travieso and J. B. Alonso, "Multisampling approach applied to contactless hand biometrics," Boston, 2012.

[4] W. Gong, X. Zhang, B. Deng and X. Xu, "Palmprint Recognition Based on Convolutional Neural Network-Alexnet," Leipzig, 2019.

[5] L. Leng, F. Gao, Q. Chen and C. Kim, "Palmprint recognition system on mobile devices with double-line-single-point assistance," *Personal and Ubiquitous Computing; London,* vol. 22, no. 1, pp. 93-104, February 2018.

[6] L. Fei, G. Lu, W. Jia, S. Teng and D. Zhang, "Feature Extraction Methods for Palmprint REcognition: A Survey and Evaluation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 49, no. 2, pp. 346-363, February 2019.

[7] M. Mouad, H. M. Vivke, L. Y. Pravin and T. G. Ashok, "A Review: Palmprint Recognition Process and Techniques," *Internal Journal of Applied Engineering Research,* vol. 13, no. 10, pp. 7499 - 7507, 2018.

[8] S. E. Salcudean and D. Karimi, "Reducing the Hausdorff Distance in Medical Image Segmentation with Convolutional Neural Networks," 2020.

[9] Y. Liu and A. Kumar, "A Deep Learning based Framework to Detect and Recognize Humans using Contactless Palmprints in the Wild," 2018.

[10] F. Marattukalam, W. Abdulla and A. Swain, "N-shot Palm Vein Verification Using Siamese Networks," in *2021 International Conference of the Biometrics Special Interest Group (BIOSIG),* Darmstadt, 2021.

[11] S. Lin, T. Xu and X. Yin, "Region of interest extraction for palmprint and palm vein recognition," in *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*, Datong, China, 2016.

[12] H. Shao, D. Zhong, X. Du, S. Du and R. N. J. Veldhuis, "Few-Shot Learning for Palmprint Recognition via Meta-Siamese Network," *IEEE Transactions on Instrumentation and Measurement,* vol. 70, 2021.

[13] G. Koch, R. Zemel and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," *In ICML deep learning workshop,* vol. 2, 2015.

APPENDICES

APPENDIX A

A.1 Weekly Logs

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 1**
Student Name & ID: Ng Jan Hui 18ACB02347	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Contactless Palmprint Verification Using Siamese Networks	

1. WORK DONE Redesigned palmprint ROI extraction module using MediaPipe Hands
2. WORK TO BE DONE Implement design ideas using code
3. PROBLEMS ENCOUNTERED None
4. SELF EVALUATION OF THE PROGRESS Progress is on track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 2**
Student Name & ID: Ng Jan Hui 18ACB02347	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Contactless Palmprint Verification Using Siamese Networks	

1. WORK DONE

- In progress of implementing ROI Extraction module using MediaPipe Hands
- Able to track and plot the landmarks on the palm in live video input

2. WORK TO BE DONE

Complete implementation by next week

3. PROBLEMS ENCOUNTERED

None

4. SELF EVALUATION OF THE PROGRESS

Progress is on track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 3**
Student Name & ID: Ng Jan Hui 18ACB02347	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Contactless Palmprint Verification Using Siamese Networks	

1. WORK DONE

- Completed implementation of new ROI extraction module
- Tested the module and is able to track the palm and segment the ROI

2. WORK TO BE DONE

Prepare dataset and begin model design

3. PROBLEMS ENCOUNTERED

None

4. SELF EVALUATION OF THE PROGRESS

Progress is on track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 4**
Student Name & ID: Ng Jan Hui 18ACB02347	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Contactless Palmprint Verification Using Siamese Networks	

1. WORK DONE
• Compiled the gathered public palmprint databases • Written a script to form image pairs from the raw databases • Labelled the image pairs to form a labelled dataset
2. WORK TO BE DONE
Begin model design and training
3. PROBLEMS ENCOUNTERED
None
4. SELF EVALUATION OF THE PROGRESS
Progress is on track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 5**
Student Name & ID: Ng Jan Hui 18ACB02347	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Contactless Palmprint Verification Using Siamese Networks	

1. WORK DONE
Designed the Siamese Model; the CNN embedding model chosen is MobileNetV2 Trained this Siamese Model with a subset of data prepared due to memory constraints Evaluated the model, performance was not satisfactory
2. WORK TO BE DONE
Select another CNN embedding model that has better performance
3. PROBLEMS ENCOUNTERED
Data was not fully utilized due to memory restrictions on Google Colab
4. SELF EVALUATION OF THE PROGRESS
Progress is on track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 6**
Student Name & ID: Ng Jan Hui 18ACB02347	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Contactless Palmprint Verification Using Siamese Networks	

1. WORK DONE

- Redesigned the Siamese Model; the CNN embedding model chosen is EfficientNet
- Trained this Siamese Model with a subset of data prepared due to memory constraints
- Evaluated the model, performance was satisfactory; However, performance can be increased if the full dataset can be used for training

2. WORK TO BE DONE

Discuss with supervisor on borrowing a more powerful machine for model training purposes.

3. PROBLEMS ENCOUNTERED

None

4. SELF EVALUATION OF THE PROGRESS

Progress is on track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 7**
Student Name & ID: Ng Jan Hui 18ACB02347	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Contactless Palmprint Verification Using Siamese Networks	

1. WORK DONE • Begin application design by drafting UML diagrams such as Use Case Diagram and Class Diagram • Updated report with progress so far
2. WORK TO BE DONE Translate the design diagrams into code
3. PROBLEMS ENCOUNTERED None
4. SELF EVALUATION OF THE PROGRESS Progress is on track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 8**
Student Name & ID: Ng Jan Hui 18ACB02347	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Contactless Palmprint Verification Using Siamese Networks	

1. WORK DONE
Designed UI components with QtDesigner
2. WORK TO BE DONE
Study the documentation on how to create application using PyQT5
3. PROBLEMS ENCOUNTERED
Unfamiliar with adding functionality to the UI components
4. SELF EVALUATION OF THE PROGRESS
Progress is on track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 9**
Student Name & ID: Ng Jan Hui 18ACB02347	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Contactless Palmprint Verification Using Siamese Networks	

1. WORK DONE

- Studied on how to add functionalities to the QT UI components via event-driven programming
- Configured functionalities for the main page

2. WORK TO BE DONE

Add critical functionalities such as camera input, registration, and verification

3. PROBLEMS ENCOUNTERED

None

4. SELF EVALUATION OF THE PROGRESS

Progress is on track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 10**
Student Name & ID: Ng Jan Hui 18ACB02347	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Contactless Palmprint Verification Using Siamese Networks	

1. WORK DONE
• Acquired powerful machine from Dr. Ng. • Retrained the Siamese Network with full dataset • Model performance was superb after evaluation. The model was saved
2. WORK TO BE DONE
Integrate the model into the application
3. PROBLEMS ENCOUNTERED
None
4. SELF EVALUATION OF THE PROGRESS
Progress is on track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 11**
Student Name & ID: Ng Jan Hui 18ACB02347	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Contactless Palmprint Verification Using Siamese Networks	

1. WORK DONE

- Completed implementation of VeriPalm application by integrating model for verification and registration functions.
- Finalized the VeriPalm application by adding in admin functions.
- Demo VeriPalm application to Dr. Ng

2. WORK TO BE DONE

Perform final touches and evaluation on VeriPalm application.

Add new contents to report

3. PROBLEMS ENCOUNTERED

None

4. SELF EVALUATION OF THE PROGRESS

Progress is on track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 12**
Student Name & ID: Ng Jan Hui 18ACB02347	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Contactless Palmprint Verification Using Siamese Networks	

1. WORK DONE Rigorously tested the VeriPalm application, all functions were able to perform as per their requirements. Add the latest contents to the report and refining previous parts.
2. WORK TO BE DONE Complete the report and prepare for submission next week.
3. PROBLEMS ENCOUNTERED None
4. SELF EVALUATION OF THE PROGRESS Progress is on track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 13**
Student Name & ID: Ng Jan Hui 18ACB02347	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Contactless Palmprint Verification Using Siamese Networks	

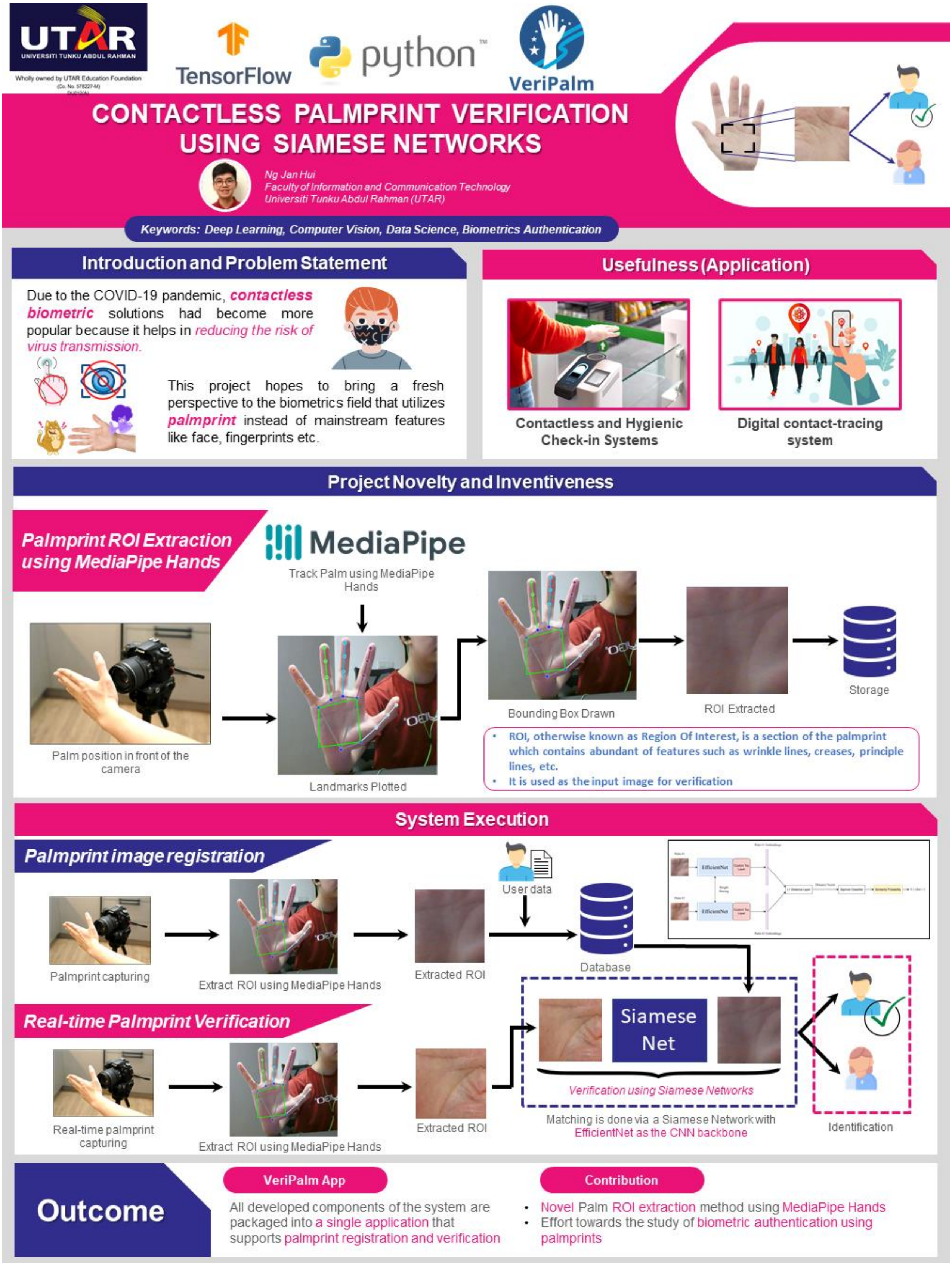
1. WORK DONE
Finalizing report and prepare for submission.
2. WORK TO BE DONE
None
3. PROBLEMS ENCOUNTERED
None
4. SELF EVALUATION OF THE PROGRESS
Progress is on track. Project could be delivered on time.

Supervisor's signature

Student's signature

APPENDIX B

B.1 Poster


UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
Wholly owned by UTAR Education Foundation
(Co. No. 578227-M)
TensorFlow
python
VeriPalm
CONTACTLESS PALMPRINT VERIFICATION USING SIAMESE NETWORKS
Ng Jan Hui
Faculty of Information and Communication Technology
Universiti Tunku Abdul Rahman (UTAR)
Keywords: Deep Learning, Computer Vision, Data Science, Biometrics Authentication
Introduction and Problem Statement
Due to the COVID-19 pandemic, contactless biometric solutions had become more popular because it helps in reducing the risk of virus transmission.
This project hopes to bring a fresh perspective to the biometrics field that utilizes palmprint instead of mainstream features like face, fingerprints etc.
Usefulness (Application)
Contactless and Hygienic Check-in Systems
Digital contact-tracing system
Project Novelty and Inventiveness
Palmprint ROI Extraction using MediaPipe Hands
MediaPipe
Track Palm using MediaPipe Hands
Palm position in front of the camera
Landmarks Plotted
Bounding Box Drawn
ROI Extracted
Storage
ROI, otherwise known as Region Of Interest, is a section of the palmprint which contains abundant of features such as wrinkle lines, creases, principle lines, etc.
It is used as the input image for verification
System Execution
Palmprint image registration
Palmprint capturing
Extract ROI using MediaPipe Hands
Extracted ROI
User data
Database
Real-time Palmprint Verification
Real-time palmprint capturing
Extract ROI using MediaPipe Hands
Extracted ROI
Siamese Net
Verification using Siamese Networks
Matching is done via a Siamese Network with EfficientNet as the CNN backbone
Identification
Outcome
VeriPalm App
All developed components of the system are packaged into a single application that supports palmprint registration and verification
Contribution
Novel Palm ROI extraction method using MediaPipe Hands
Effort towards the study of biometric authentication using palmprints

PLAGIARISM CHECK RESULT

Contactless Palmprint Verification using Siamese Networks

ORIGINALITY REPORT

6%	4%	3%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utm.my Internet Source	1%
2	eprints.utar.edu.my Internet Source	1%
3	export.arxiv.org Internet Source	<1%
4	digitalcommons.lasalle.edu Internet Source	<1%
5	deepai.org Internet Source	<1%
6	Felix Marattukalam, Waleed H. Abdulla, Akshya Swain. "N-shot Palm Vein Verification Using Siamese Networks", 2021 International Conference of the Biometrics Special Interest Group (BIOSIG), 2021 Publication	<1%
7	fict.utar.edu.my Internet Source	<1%

8	Aykut, Murat, and Murat Ekinci. "Developing a contactless palmprint authentication system by introducing a novel ROI extraction method", Image and Vision Computing, 2015. Publication	<1 %
9	Aythami Morales, Ajay Kumar, Miguel A. Ferrer. "Incorporating color information for reliable palmprint authentication", 2011 18th IEEE International Conference on Image Processing, 2011 Publication	<1 %
10	Yang Liu, Ajay Kumar. "Contactless Palmprint Identification using Deeply Learned Residual Features", IEEE Transactions on Biometrics, Behavior, and Identity Science, 2020 Publication	<1 %
11	Masayuki Ito, Toshihiro Hattori, Yutaka Yoshida, Kiyoshi Hayase et al. "An 8640 MIPS SoC with Independent Power-Off Control of 8 CPUs and 8 RAMs by An Automatic Parallelizing Compiler", 2008 IEEE International Solid-State Circuits Conference - Digest of Technical Papers, 2008 Publication	<1 %
12	Lin Zhang, Lida Li, Anqi Yang, Ying Shen, Meng Yang. "Towards contactless palmprint recognition: A novel device, a new benchmark, and a collaborative	<1 %

	representation based identification approach", Pattern Recognition, 2017 Publication	
13	lo-motion.de Internet Source	<1 %
14	a3f884e4-5ceb-45fb-b43b-0e79ec67e928.filesusr.com Internet Source	<1 %
15	"Computer Vision – ECCV 2018", Springer Science and Business Media LLC, 2018 Publication	<1 %
16	"Image Processing and Communications", Springer Science and Business Media LLC, 2020 Publication	<1 %
17	Morales, A., M.A. Ferrer, and A. Kumar. "Towards contactless palmprint authentication", IET Computer Vision, 2011. Publication	<1 %
18	Advances in Information Security, 2014. Publication	<1 %
19	Dong-Yeon Lee, Minji Lee, Seong-Whan Lee. "Classification of Imagined Speech Using Siamese Neural Network", 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2020 Publication	<1 %

20	digitalscholarship.unlv.edu Internet Source	<1 %
21	"Inventive Computation and Information Technologies", Springer Science and Business Media LLC, 2021 Publication	<1 %
22	Audrey Poinsot, Fan Yang, Michel Paindavoine. "Small Sample Biometric Recognition Based on Palmprint and Face Fusion", 2009 Fourth International Multi-Conference on Computing in the Global Information Technology, 2009 Publication	<1 %
23	Raimond L. Winslow. "Imaging-Based Integrative Models of the Heart: Closing the Loop between Experiment and Simulation", Novartis Foundation Symposia, 11/15/2002 Publication	<1 %
24	Mahdieh Izadpanahkakhk, Seyyed Razavi, Mehran Taghipour-Gorjikolaie, Seyyed Zahiri, Aurelio Uncini. "Deep Region of Interest and Feature Extraction Models for Palmprint Verification Using Convolutional Neural Networks Transfer Learning", Applied Sciences, 2018 Publication	<1 %
25	coek.info Internet Source	

		<1%
26	mirror.its.dal.ca Internet Source	<1%
27	tesi.cab.unipd.it Internet Source	<1%
28	www.datasciencecentral.com Internet Source	<1%
29	Elsa Santos Febles, Marlis Ontivero Ortega, Mitchell Valdes-Sosa, Hichem Sahli. "Machine Learning techniques for the diagnosis of Schizophrenia based on Event Related Potentials", Cold Spring Harbor Laboratory, 2022 Publication	<1%
30	dspace.mit.edu Internet Source	<1%
31	etd.adm.unipi.it Internet Source	<1%
32	intranet.cs.man.ac.uk Internet Source	<1%
33	www.spiedigitallibrary.org Internet Source	<1%

PLAGIARISM CHECK RESULT

Exclude quotes	On	Exclude matches	< 8 words
Exclude bibliography	On		

PLAGIARISM CHECK RESULT

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Ng Jan Hui
ID Number(s)	18ACB02347
Programme / Course	BACHELOR OF COMPUTER SCIENCE (HONOURS)
Title of Final Year Project	Contactless Palmprint Verification using Siamese Networks

Similarity	**Supervisor's Comments** **(Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 6 % **Similarity by source** Internet Sources: 4% Publications: 3% Student Papers: 0%	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

PLAGIARISM CHECK RESULT

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

______________________________	______________________________
Signature of Supervisor	Signature of Co-Supervisor
Name: Dr. Ng Hui Fuang	Name: __________________________
Date: 21-04-2022	Date: ___________________________

FYP2 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY

(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	18ACB02347
Student Name	Ng Jan Hui
Supervisor Name	Dr. Ng Hui Fuang

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
N/A	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)

FYP 2 CHECKLIST

N/A	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____

(Signature of Student)

Date: 21-04-2022