



Camera
BERZIET UNIVERSITY
FACULTY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Secure Authentication through AI-Powered Contactless Palmprint Recognition

Prepared by:

Layan Shoukri 1201225

Amal Butmah 1200623

Duaa Suliman 1200909

Supervised by:

Dr. Wasel Ghanem

A graduation Project submitted to the Department of Electrical
And Computer engineering in partial fulfillment of the requirements
For the degree of B.Sc. in Electrical Engineering

BIRZEIT
February 2025

Abstract

During the COVID-19 pandemic, the necessity for hygienic, contactless authentication systems increased as people sought to avoid physical contact with publicly used devices. This project develops a contactless palmprint recognition system to enhance authentication security through distinct palmprint characteristics. User safety and comfort are improved as the proposed method eliminates the need for fingerprint-based authentication by capturing palmprint images without physical touch. A contactless palmprint recognition prototype served as the project's main focus throughout its development and testing phases. The device employs advanced processing algorithms to capture high-resolution palmprint images, ensuring accurate user identification and verification. This solution is suitable for various authentication requirements due to its compact design, user-friendly operation, and cost-effective hardware components. The developed project highlights the potential applicability of contactless palmprint technology across multiple settings as a low-cost, efficient, and hygienic user authentication solution.

Index Terms—Palmprint, Recognition, CNN, SNN, Raspberry Pi, Camera, Classification.

المستخلص

خلال جائحة كورونا، زادت الحاجة إلى أنظمة المصادقة اللاتلامسية الصحية، حيث سعى الناس إلى تجنب الاتصال الجسدي بالأجهزة المستخدمة بشكل عام. يهدف هذا المشروع إلى تطوير نظام التعرف على بصمة راحة اليد اللاتلامسية لتعزيز أمان المصادقة من خلال خصائص بصمة راحة اليد الفريدة. يتم تحسين سلامة المستخدم وراحته، حيث يلغي الأسلوب المقترح الحاجة إلى طرق المصادقة المعتمدة على بصمات الأصابع من خلال التقاط صور لبصمة راحة اليد دون أي تلامس جسدي. كان النموذج الأولي لنظام التعرف على بصمة راحة اليد اللاتلامسية محور التركيز الرئيسي للمشروع طوال مراحل التطوير والاختبار. يستخدم الجهاز خوارزميات معالجة متقدمة لالتقاط صور عالية الدقة لبصمة راحة اليد، مما يضمن دقة عمليات التعرف والتحقق من المستخدم. تعد هذه التقنية مناسبة لمتطلبات المصادقة المختلفة نظرًا لتصميمها المدمج، وسهولة تشغيلها، ومكوناتها المادية منخفضة التكلفة. يبرز المشروع المطور الإمكانيات التطبيقية لتقنية بصمة راحة اليد اللاتلامسية في بيئات متعددة، باعتبارها حلاً فعالاً ومنخفض التكلفة يوفر مصادقة صحية وآمنة للمستخدمين.

Contents

Acronyms and Abbreviations.....	v
List of Figures	vi
List of Tables	viii
Chapter 1 Introduction.....	1
1.1 General Introduction.....	1
1.2 Problem statements.....	1
1.3 Motivation.....	2
1.4 Device work description	2
1.5 Organization of the report	3
Chapter 2 Literature Review.....	4
2.1 Introduction to Contactless Biometric Authentication Using Palm Print and Palm Vein Recognition	4
2.2 The Beginning of the World of palmprint-based identification systems	5
2.3 Quick overview of the palm veins	5
2.4 Palmprint and palm veins recognition system development according to the reviewed papers	6
Chapter 3 System Implementation and Design	10
3.1 Datasets.....	10
3.2 Prototype Device and Hardware Components	13
3.2.1 Prototype Box	13
3.2.2 Raspberry Pi 5.....	13
3.2.3 Raspberry Pi camera module – pi noir 8MP Version2	14
3.2.4 Micro SD card	15
3.2.5 White Light	15
3.3 Used algorithms and techniques.....	16
3.3.1 Traditional Algorithms and Techniques	16
3.3.2 Deep Learning Algorithms	16

3.3.3	Machine learning algorithms.....	20
3.4	Methodology	21
3.4.1	Palm Image Acquisition.....	21
3.4.2	Image pre-processing.....	21
3.4.3	ROI Extraction	21
3.4.4	Feature Extraction	22
3.4.5	Feature Matching.....	22
Chapter 4	Simulation Results.....	23
4.1	ROI Extraction	23
4.2	Training model.....	28
4.3	Matching Features	34
Chapter 5	Conclusion, Challenges, and Future Work	37
5.1	Conclusion	37
5.2	Challenges.....	37
5.3	Future work.....	38
References	39

Acronyms and Abbreviations

SD	Secure Digital
CLAHE	Contrast-limited Adaptive Histogram Equalization
LED	Light Emitting Diode
PCA	Principal Component Analysis
NPE	Neighborhood-Preserving Embedding
SDSPCA-NPE	Supervised Discriminative Sparse Principal Component Analysis - Neighborhood-Preserving Embedding
CPU	Central Processing Unit
I/O	Input / Output
SDRAM	Synchronous Dynamic Random Access Memory
HDR	High Dynamic Range
ISP	Image Signal Processor
ROI	Region of Interest
CNN	Convolutional Neural Network
SNN	Siamese Neural Network
ORB	Oriented FAST and Rotated BRIEF
LoG	Laplacian of Gaussian
BMP	Bitmap
DWT	Discrete Wavelet Transform

List of Figures

Fig 2.1: Schematic diagram of the palmprint capture device [3]	5
Fig 2.2: Contactless device to capture multi-spectral palm vein image [15].....	8
Fig 3.1: Samples from CASIA dataset [16]	10
Fig 3.2: Samples from IIT Delhi dataset [17]	11
Fig 3.3: Samples from Palmprint dataset.....	11
Fig 3.4: Samples from our dataset.....	12
Fig 3.5: Prototype box and its design	13
Fig 3.6: Raspberry pi 5 [19]	14
Fig 3.7: Raspberry Pi Noir camera module [21]	15
Fig 3.8: 64GB Micro-SD card.....	15
Fig 3.9: CNN Architecture [28]	17
Fig 3.10: Convolutional layer in CNN [31]	18
Fig 3.11: Pooling layer in CNN [33]	18
Fig 3.12: Fully connected layer in CNN [35]	19
Fig 3.13: SNN Architecture [36]	20
Fig 3.14: K-means clustering [37].....	20
Fig 3.15: System flow chart	21
Fig 4.1: Original image	23
Fig 4.2: Grayscale image	24
Fig 4.3: Center of hand	24
Fig 4.4: Valley points	25
Fig 4.5: Fingertips	25

Fig 4.6: Middle fingertip.....	26
Fig 4.7: First valley point.....	26
Fig 4.8: Second valley point	27
Fig 4.9: Valley points	27
Fig 4.10: Image rotation and valley point connection	27
Fig 4.11: ROI image	28
Fig 4.12: Retrieval model accuracy	30
Fig 4.13: Model of 159 people from our dataset accuracy	31
Fig 4.14: Correct prediction based on Model of 159 people from our dataset	31
Fig 4.15: CNN model summary	32
Fig 4.16: Performance measures for ResNet model on our dataset	34
Fig 4.17: Feature Points and Matches	35
Fig 4.18: Comparison of Otsu Thresholding and K-means Segmentation on Training and Testing Images.....	36

List of Tables

Table 2.1: Contributions to palmprint and palm vein recognition [2].....	7
Table 2.2: Algorithms on different datasets [15]	9
Table 4.1: Results summary for different CNN models	29
Table 4.2: ResNet model on different numbers of data from dataset.....	30

Chapter 1

Introduction

1.1 General Introduction

The subject of technology and systems in identification has remained a topic of discussion as technology advances and the need for proper implementable security measures across the public grows. These systems have been adopted in areas such as access control, security, employee management, banking, and healthcare. Biometric aspects and unique features have garnered the most attention in biometric authentication technologies, such as fingerprints, face recognition, and iris scans, due to their effectiveness and security, as demonstrated in some research. More recently, the distinctiveness of palmprints has gained attention for its unique features, including lines, ridges, and textures, which make it an effective biometric modality and the foundation of our project [1].

To build a contactless system for palmprint recognition, we worked on researching the necessary components for the device and concluded with essential elements i.e. a Raspberry pi, a white LED, and a high-resolution camera. We conducted a comprehensive search to source these components, and this process involved several challenges, which will be discussed in the challenges section. The decision was made to focus solely on capturing the palmprint for recognition. The required components, including a white LED strip, 64GB micro-SD, Raspberry Pi, and a high-resolution camera were ordered. When the device was built, it captured detailed images of the palm for data acquisition. The program then processed these images, extracted the unique region of interest, applied feature extraction and matching, and verified the user's identity.

1.2 Problem statements

As mentioned before, the most popular user identification and verification systems are fingerprints, facial recognition, and iris scanning. However, it has been shown that these methods, based on the previously mentioned features, suffer from several issues, the most significant being the risk of biometric information theft, such as fingerprints which pose hygiene concerns due to direct contact

between the finger and the verification device. As another example of the problem, there is iris recognition technology, which is greatly affected by lenses. As for facial recognition, many people do not want to take pictures of their faces. Additionally, it is affected by factors such as face masks and glasses. The most significant issue, however, is the high cost of the equipment required for these technologies, which leads to increased selling prices. All of these contributed to the trend towards the palmprint recognition system, which solved most of the problems facing user identification and verification systems [1].

1.3 Motivation

Compared to the identification systems, the palm is not considered sensitive to privacy compared to facial recognition. The absence of touch in this feature made the system that relies on it for identity recognition a healthy system that gained the admiration and interest of its users, especially after discovering the uniqueness of the palm feature i.e. palmprint, which is rich in features such as lines and spots. Thus, a contactless identification system appeared with more safety, reliable, accurate, and highly comfortable due to its reliance on features that studies have proven to be unique and distinctive among people, in addition to the difficulty in capturing and stealing [2].

1.4 Device work description

After building our device, we are being able to capture high-resolution images of the palm to assemble our dataset. Once the palm image is captured, the processing phase begins, where the region of interest is extracted to retain vital information while ensuring high classification accuracy. This process minimizes both processing time and memory usage.

In the processing stage, several algorithms are applied to accurately extract the region of interest. A deep learning-based classification model is trained on a large dataset of palmprint images to learn distinctive patterns. During verification, the recently captured palmprint image is classified based on the learned features and compared with predefined classes that the model was trained on. If the image is classified as belonging to a registered user, verification is successful; otherwise, access is denied.

The key advantage of this system is its low overall cost, making it feasible for widespread deployment at an affordable price while maintaining greater accuracy compared to existing authentication systems.

1.5 Organization of the report

The report is organized into five chapters which are organized as follows:

- Chapter 2 presents the overall background and prior works.
- Chapter 3 introduces datasets, hardware, device prototype, algorithms used, and the methodology followed.
- Chapter 4 describes the experiments and results analysis.
- Chapter 5 concludes the work presented, presents the challenges faced in this project and introduces some possible future works.

Chapter 2

Literature Review

Before any crucial step in graduation projects, such as this one, the topic of the proposed project is studied with a superficial and quick reading of previous similar projects. It is then read deeply and carefully to collect useful and comprehensive information about the project after it is agreed upon. This is one of the first and most important steps taken before starting anything related to the project, such as extracting features, building the system, and so on. There was great credit for previous relevant studies, research, and papers, which helped in understanding the project, knowing its characteristics, its importance, and how it is likely to be applied.

In this chapter, we will discuss the background and review related works related to the development of palmprint-based and palm-vein-based authentication systems.

2.1 Introduction to Contactless Biometric Authentication Using Palm Print and Palm Vein Recognition

As already mentioned in the chapters of this report, it is necessary to reaffirm the reliability of biometric authentication devices that take advantage of people's unique physiological or behavioural characteristics for identification and verification. Many of these characteristics have been mentioned that favour it over other traditional authentication methods because of its ability to provide high security due to the difficulty of theft and forgery. One of the most important of these biometric characteristics on which our project was based is the palmprint of the hand and its veins, which are considered distinctive characteristics.

One of them is visible, which is the palmprint and the other is invisible and can only be seen using near-infrared light, which are the palm veins. What makes the biometric authentication system based on the palmprint and its veins an interesting device and a focus for study and research is the lack of need for physical contact with the sensor of the authentication device, as these biometric features are contactless, which automatically leads to the user feeling comfortable, especially for those who have concerns regarding hygiene-related problems. This feature also allows for the long life of the authentication device, as it does not corrode due to lack of touching.

2.2 The Beginning of the World of palmprint-based identification systems

After the distinctive and unique characteristics of the palmprint were revealed among people, Zhang and others decided in 2003 to work on a palmprint-based identity verification device using low-quality images in which a two-dimensional Gabor coding scheme was used to extract features and represent the palmprint. The device was primitive and relatively large and its use requires hand contact with the device, as shown in Fig. 2.1. However, it achieved good results in terms of speed and accuracy, with a reasonable true acceptance rate of 97% [3].

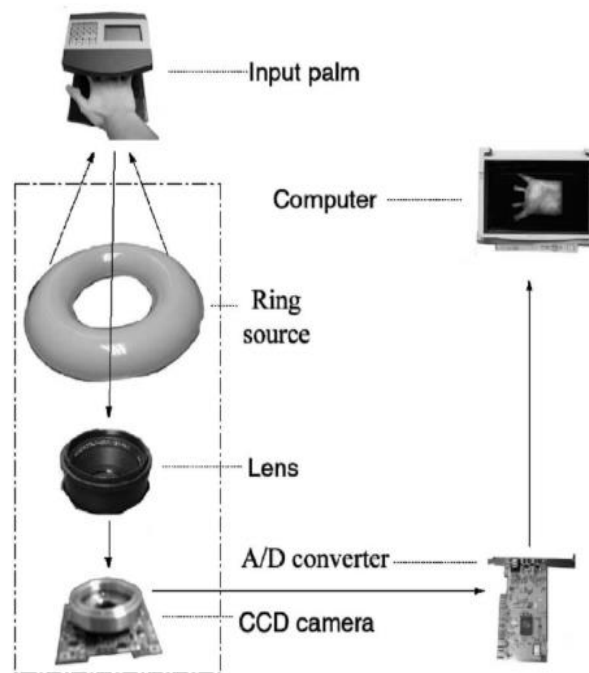


Fig 2.1: Schematic diagram of the palmprint capture device [3]

In 2007, Han and others presented the first system that captures images of the palm without direct contact with the device in unrestricted scenes. They used two cameras, one of which photographed near-infrared rays to determine the shape of the hand, and the other to capture palmprints in visible light to extract features from them using a new algorithm based on using color and shape information of the palm [4].

2.3 Quick overview of the palm veins

The randomness of the vein patterns in the palm is genetic, and even twins each have a vein pattern different from the other, and this is what makes it a focus of interest. The most interesting aspect of it

is that it is invisible and hidden inside the body. It cannot be seen with the naked eye, so it cannot be impersonated or its representation can be left anywhere.

To take a picture of the veins, the palm or any part of the body to see its veins is exposed to near-infrared light, and then certain devices that sense this light are used, such as some cameras[5].

Many biometric authentication devices have used palm veins, and Gunjan Shah in 2015 implemented this in practice using mainly near-infrared light and a camera that takes images at that frequency. Several algorithms were used to process images, the best of which was the template matching technique, producing a system with a good accuracy of 93.54% [6].

Research is still achieving tremendous development and achievement with the advancement of technology and the accuracy of the tools used, achieving results with much higher accuracy.

2.4 Palmprint and palm veins recognition system development according to the reviewed papers

Developments and research continued in the field of devices based on biometric characteristics, as researchers were not satisfied with the handprint alone in the field of identity verification, especially since these palmprints are affected by external factors to which the hand is exposed, the most important of which is age and some works that erase these palmprints. They have worked hard to increase the reliability of such systems to become palm-based and veined.

In recent years, interest in palmprint and palm vein recognition technology has begun, and there have been many researches that discussed and followed the process of developing the design of the palmprint and palm vein recognition device to reach a device with high efficiency and low costs. Several hardware components and algorithms in the construction and design process were followed.

Table 2.1 shows some of the many contributions made to the topic of palmprints and palm veins recognition using various techniques and data sets, with the accuracy percentage for each contribution. The table itself shows the number of attempts and achievements in the same subject through the innovation of numerous techniques and the production of different data sets, and this indicates the importance of this system and the attempt to develop it to the extent that makes it very accurate.

Contribution	Technique	Dataset(s) used	Recognition Accuracy
Mirmohamadsadeghi and Drygajlo (2011) [7]	Local derivative pattern, histogram intersection	CASIA	98.30%
Lee (2012) [8]	2D Gabour filter, Hamming distance	Self-constructed	99.18%
Kang and Wu (2014) [9]	Local Binary pattern, Support Vector Machine	CASIA	100%
Elansir and Shamsuddin (2014) [10]	Linear Discriminative analysis, Cosine distance	PolyU	99.74%
Lu et al (2016) [11]	Multi-scale Local Binary pattern, Local derivative pattern, similarity measure	CASIA	99.99%
Cho and Kar-Ann (2018) [12]	Gabor filter, Hamming distance	PolyU	99.13%
Hernández-García et al (2019) [13]	CLAHE	CASIA Multi-Spectral Palmprint Images	99.28%
Pititheeraphab et al (2020) [14]	Geometric affine invariants	Self-constructed	99.76%

Table 2.1: Contributions to palmprint and palm vein recognition [2]

The following paper was the most recent paper we read that studied and applied in practice the subject of identification using the palmprint and palm vines features. This paper was produced in 2023 by Chinese researchers and it was the main reference for everything we needed during our research, so it has the largest share in this section to explain, what components and algorithms were used to build the system.

Wei Wu, Yunpeng Li, Yuan Zhang, and Chuanyang Li were interested in building an Identity Recognition System Based on Multi-Spectral Palm Vein Image, they built the contactless device that is shown in Fig. 2.2, from two Near-Infrared LED CST brand model BL-270-42-IR with 850nm wavelength which are equipped with an intensity adjustment controller and a camera model MV-VD120SM with multispectral imaging which uses multiple spectral ranges to capture palm veins with more detailed information [15].

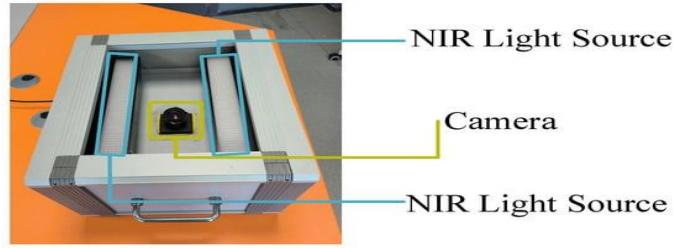


Fig 2.2: Contactless device to capture multi-spectral palm vein image [15]

For the methodology, they followed four steps. First, they chose the region of interest in the image pre-processing step which was a square whose side length is equal to the distance between the valleys between the little finger and ring finger, the index finger and middle finger, and the square shape starts from these valleys towards the inside of the hand. Then, they adopt supervised discriminative sparse principal component analysis (SDSPCA-NPE), which combines supervised discriminative information and sparse constraints into the PCA model, in feature extraction which reduces the dimensionality and recognition and the projected palm vein data exhibits an improved distance distribution, enhancing the classification performance of palm vein data. After that, the next step was to test the feature matching and recognition which is based on Euclidean distance between feature vectors derived from the projected subspace with matching thresholds determined by the intersection of intra-class and inter-class matching curves [15].

This paper used several datasets, including the CASIA database, the Hong Kong Polytechnic University database, the Tongji University database, as well as several self-contained datasets. To increase their accurate research, they used several algorithms, where the results are shown in Table 2.2, which shows some of the algorithms used for each type of data set, with the results resulting from equal error rates and identification time [15].

Algorithms	Database	EER (%)	Time($10^{-4}s$)
PCA	Self-built	0.28	19.59
	CASIA	2.38	19.77
	PolyU	1.5	19.45
	Tongji	6	19.58
NPE	Self-built	0.50	13.81
	CASIA	7.50	13.90
	PolyU	1	14.19

	Tongji	9.6	14.11
SDSPCA	Self-built	1.50	13.56
	CASIA	15.39	13.89
	PolyU	5.50	13.57
	Tongji	10.75	13.63
SDSPCA-NPE	Self-built	0.10	19.77
	CASIA	0.50	38.50
	PolyU	0.16	19.75
	Tongji	0.19	19.69

Table 2.2: Algorithms on different datasets [15]

After evaluating multiple algorithms and their respective results, as depicted in the table above, the combination of SDSPCA and NPE proved effective, showing a significant reduction in the Equal Error Rate.

Chapter 3

System Implementation and Design

This chapter introduces the system construction including hardware and software, where the components of our recognition device, the algorithms used in programming the system, especially those related to the machine and deep learning, the datasets, and the methodology will be discussed.

3.1 Datasets

Some of the related works discussed in Chapter 2 touched on the open-source datasets they used. Among these works, there were 3 common datasets used i.e. CASIA Version1.0, PolyU-IITD V1.0, and Palmprint which were researched and used to build the model initially until assembling our dataset.

CASIA Multi-Spectral Palmprint Image Database Version 1.0 is a collection of 7200 palm vein images that were captured from 100 different hands by a multispectral camera, each image is an 8-bit gray-level JPEG file. The images of each hand were taken in two sessions, one month apart. In each session, images were taken at six different electromagnetic spectrums, i.e. 460nm, 630nm, 700nm, 850nm, 940nm, and white light while every six images were taken at a different spectrum [16]. Fig 3.1 shows samples from this dataset.

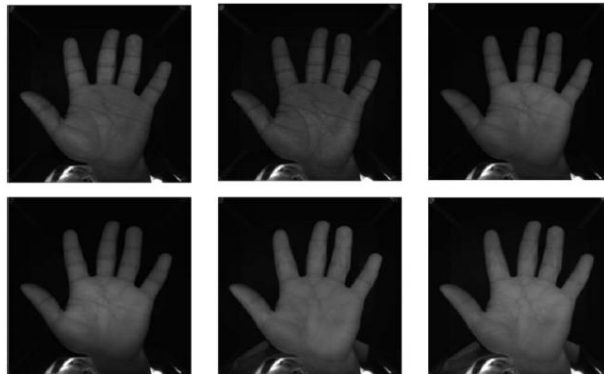


Fig 3.1: Samples from CASIA dataset [16]

IIT Delhi Touchless Palmprint Database (Version 1.0) consists of palmprint images that were taken from 230 different people from IIT Delhi University students and employees in the age group 12-57 using contactless imaging device during July 2006 - Jun 2007 [17].

Seven images were taken for each hand from each person in different hand positions. All were captured in a closed small room for the hand, using a camera with a circular fluorescent illumination around its lens [17].

The images were in bitmap (*.bmp) format with a resolution of 800×600 pixels, in addition to 150×150 pixels cropped and normalized palmprint images [17]. Fig 3.2 shows samples from this dataset.

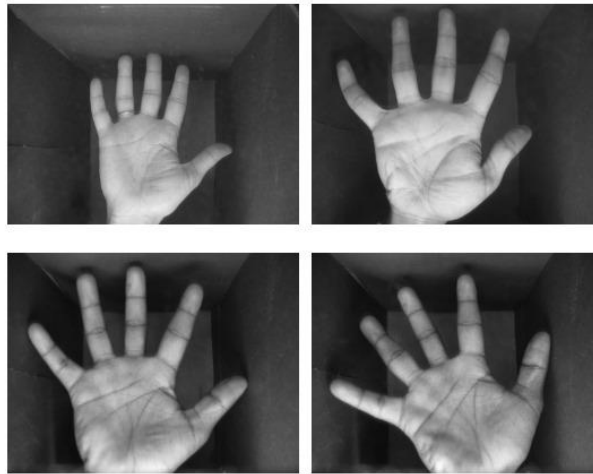


Fig 3.2: Samples from IIT Delhi dataset [17]

The Palmprint dataset is an open-source dataset containing grayscale images of ROIs from 100 individuals, with each user having a total of six images. Initially, the dataset was divided into three training images and three validation images per person. However, to enhance model performance and maintain an 80:20 split, we modified the distribution by assigning four images to the training set and two to the validation set. Fig 3.3 shows samples from this dataset.

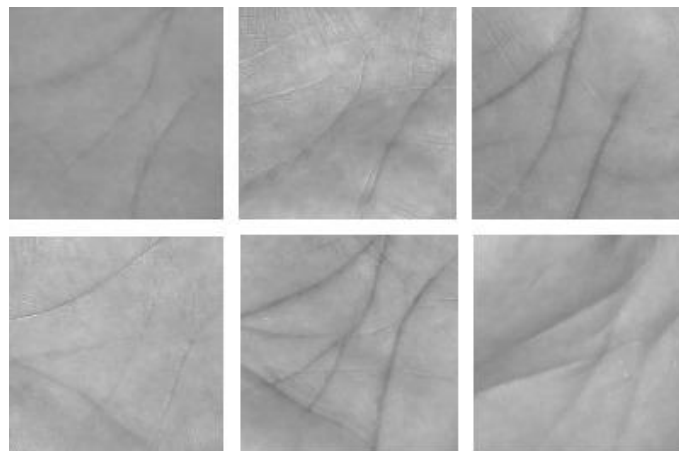


Fig 3.3: Samples from Palmprint dataset

After utilizing these open-source datasets for initial testing and model-building, we assembled and used our own dataset, the Birzeit University Palmprint Image Dataset, to further refine and enhance the system.

Our dataset consists of 954 palm images from 159 students, specifically designed to support the development of a palmprint authentication system. The dataset was collected using our custom-built imaging device, which features a Raspberry Pi Noir camera module. The images were captured under white light conditions and stored as RGB-level BMP files, ensuring high-quality and detailed data for biometric research.

The dataset was created to enable a palmprint-based authentication system for controlling access to the *Ka3kash* room at the university. Each user placed their hand inside a prototype box, where six images were captured from different angles. A Raspberry Pi processed these images to extract ROIs for authentication.

The dataset was collected from students enrolled in the Interfacing Techniques course. The authentication process ensures that access to the "Ka3kash" room is granted only if the detected palmprint matches a previously registered user. This dataset not only supports the biometric system but also provides a valuable resource for research on palmprint recognition, biometric authentication, and user identification systems. As demonstrated in the provided samples in Fig 3.4, the dataset includes a variety of high-quality palm images showcasing the detailed patterns required for effective authentication. Additionally, the dataset was provided in two formats: **full hand images** and **ROI**, allowing flexibility for researchers and developers to experiment with different approaches in palmprint recognition.

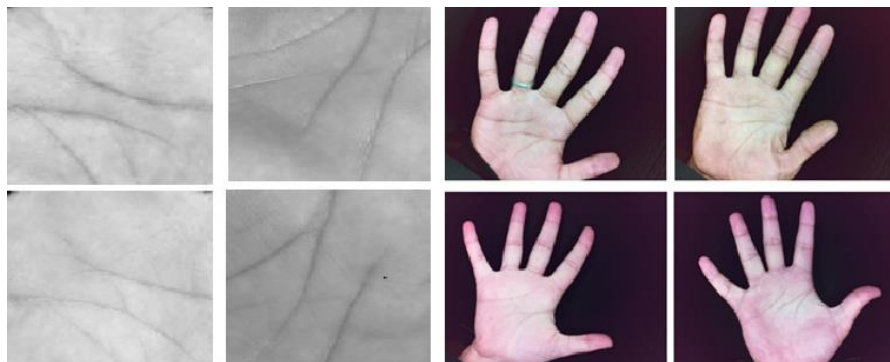


Fig 3.4: Samples from our dataset

3.2 Prototype Device and Hardware Components

Our palmprint authentication system consists of a specially built box that functions to obtain high-quality palmprint images needed for biometric verification. The following details every hardware element in the prototype.

3.2.1 Prototype Box

The system consists mainly of a specially constructed box that users can place their hands inside. The box achieves dual functions by shielding away outside light and creating a stable test environment for image acquisition. Hand positioning and stability are optimized within the interior design of the system, which leads to better system accuracy during the capture process. The box and its design are shown in Fig 3.5.

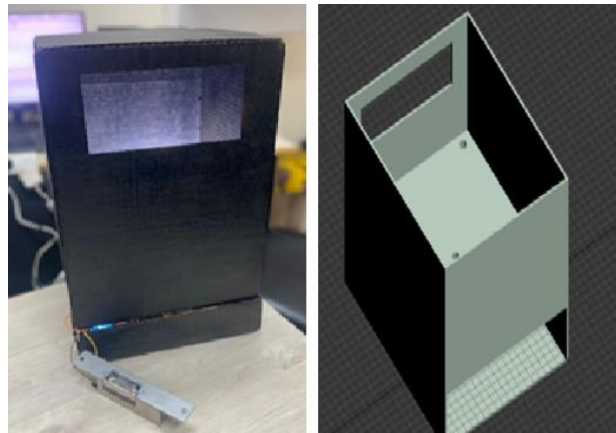


Fig 3.5: Prototype box and its design

3.2.2 Raspberry Pi 5

Raspberry Pi is considered to be a minicomputer that uses different kinds of processors which makes it efficient in performing several tasks. It has Raspbian operating system based on Linux, and it supports installing only open-source operating systems and apps. Raspberry Pi also supports several programming languages such as Python, C, BASIC, Ruby, Perl, and C++ [18].

Raspberry Pi 5 is the latest version of a single-board computer, it is integrated with a 1.6GHz Processor. Also, it includes Wi-Fi, Bluetooth, Micro SD card storage, LPDDR4X-4267 SDRAM (4GB or 8GB), and many of computer features. It is 2 to 3 times faster in CPU performance and twice the memory and I/O bandwidth over the previous models [19].

The Raspberry Pi 5 which is shown in Fig 3.6, was chosen as a processing unit. Because it has an improved Image Signal Processor (ISP) which processes the raw data from the camera sensor into a high-quality image. In addition to its HDR imaging feature which combines images to prevent the highlights from blowing out and dark areas from being too dark [20].



Fig 3.6: Raspberry pi 5 [19]

The Raspberry Pi 5 operates as the central processing unit for the device. It controls the camera module functions and manages LED light operations while operating all image capturing and processing software on the system.

The Raspberry Pi 5 was selected because it combines small dimensions with powerful computation capabilities to perform its I/O operations successfully.

3.2.3 Raspberry Pi camera module – pi noir 8MP Version2

The Raspberry Pi NoIR which is shown in Fig 3.7, is a camera module with no infrared filters which makes it sensitive to infrared light at wavelengths from 700nm and up. It can take clear pictures in the dark by applying an infrared light on the object. It includes a Sony IMX219 sensor which can capture IR light in addition to visible light, to give images with high resolution, up to 3280×2464 pixels (8megapixels) [21].

The Raspberry Pi NoIR is compatible with all Raspberry Pi models and can be connected using Camera Serial Interface (CSI), and used in different applications such as security systems and scientific projects [22].



Fig 3.7: Raspberry Pi Noir camera module [21]

The system places the Raspberry Pi Noir Camera Module beneath the hand slot for direct image capture of palmprints. The camera module operates best in low-light situations without an infrared filter because it produces clear palmprint images under white light illumination. The camera hole in the box enables exact alignment which lowers image distortions during usage.

3.2.4 Micro SD card

Micro-SD is a removable flash memory card used as storage in several devices such as smartphones, digital cameras, USB flash devices [25] and Raspberry Pi's [19]. The design consists of one or more flash memory dice and a microcontroller which acts as an interface between the card reader and the memory to supply the storage space abstraction [26].

As a benefit, the micro-SD provides a low-cost, fast writing speed and a large capacity of 400GB additional storage for electronic devices [26].

In our device, we use a 64GB micro-SD card as the one shown in Fig 3.8, to save all the images that are captured by the camera. This card is put in a micro-SD card slot inside the Raspberry Pi to process the images.



Fig 3.8: 64GB Micro-SD card

3.2.5 White Light

The device box contains white LED strip that has specific placements to deliver equal palm illumination. By using illumination from white LED strip, the system improves palmprint image clarity

through shadow reduction and by providing proper illumination for all unique palmprint patterns. The white light functionality is to maintain uniformity of all recorded images.

3.3 Used algorithms and techniques

Our work is based on the following algorithms and techniques:

3.3.1 Traditional Algorithms and Techniques

Traditional processing methods within the image workflow enable users to perform successful image enhancement and analysis needs. The conversion to grayscale removes processing complexity because it eliminates colour details and models need uniform input dimensions after resizing. The Non-Local Means method delivers noise reduction by maintaining both edges and details for the images. The object boundaries become detectable through Canny edge detection while Otsu's thresholding procedure transforms grayscale data into easily analysed binary form. The morphological operations help clear shapes from artifacts thus enabling contour detection to extract object boundaries. The evaluation of shapes depends on both convex hull techniques and convexity defect detection while perspective transformation enables precise extraction of object characteristics. These techniques collectively improve image clarity, facilitate feature detection, and enhance object recognition.

3.3.2 Deep Learning Algorithms

The system implements deep learning technology which leads to more precise and effective observational results. The learning and spatial pattern distinction of input data strongly depends on Convolutional Neural Networks (CNNs). Pre-trained models help the system to improve performance through transfer learning when dealing with small amounts of labelled input data and shorten training time. For optimal task classification models use cross-entropy loss as their fundamental measurement tool because it evaluates the distance between probability predictions and true class labels. The learning process of models is guided by this loss function through parameter adjustments which stem from prediction mistakes.

Section 3.3.2.1 provides the details about CNNs' methodology along with their architectural elements.

3.3.2.1 CNN

Convolutional Neural Networks (CNNs) are a specialized class of deep neural networks primarily used for processing structured grid data, especially images. CNNs excel in computer vision tasks such as object detection, image classification, and image segmentation. Their strength lies in the ability to process hierarchical features directly from raw pixel values, allowing them to recognize complex

patterns and improve model performance.

The core operation in CNNs is the convolution operation, which applies filters to the input data matrix to generate feature maps. Unlike regular matrix multiplication, convolution identifies local patterns within the data, preserving key features and enhancing data regularity [28].

A standard CNN follows the architectural design shown in Fig 3.9.

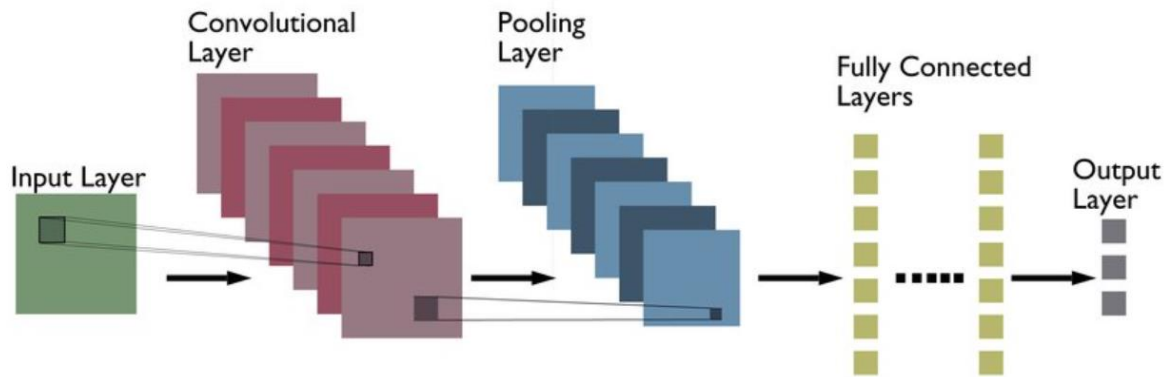


Fig 3.9: CNN Architecture [28]

The convolution operation in CNN has equation 3.3:

$$(I * K)(x, y) = \sum_i \sum_j I(x + i, y + j) \cdot K(i, j) \quad (3.3)$$

Where:

- I is the input image
- K is the kernel (filter)
- x,y are the spatial coordinates in the output feature map
- i,j are the offsets within the kernel window

The CNN architecture typically consists of multiple layers. The input layer accepts the image data, followed by the **convolutional layers**. These layers apply filters to local regions of the image, reducing its dimensions while preserving key features [30]. Fig 3.10 illustrates the working principle of a

convolution layer.

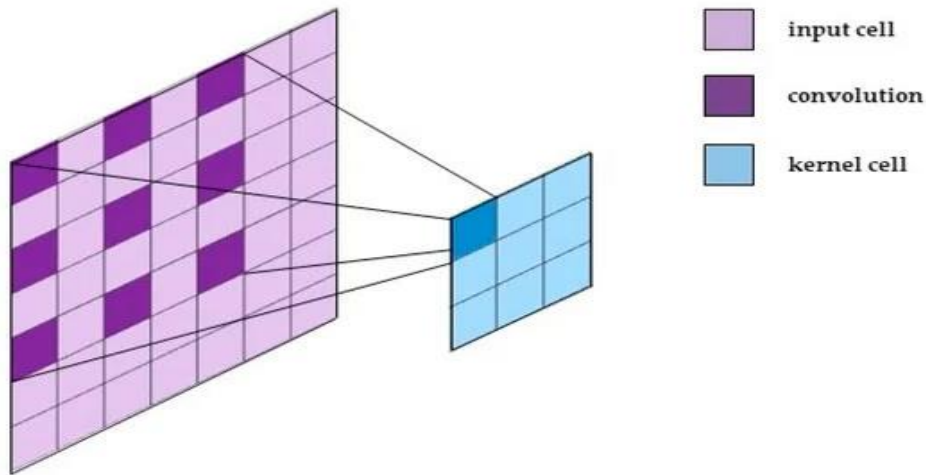


Fig 3.10: Convolutional layer in CNN [31]

The third component in CNN architecture is the pooling layer, which reduces the spatial dimensions of feature maps while preserving essential information. One common technique is max pooling, which selects the maximum value in a region, improving computational efficiency and feature robustness [32]. The pooling operation is represented in Equation 3.4:

$$Y(i,j) = \max\{x(s,t) \mid s \in [i, i + n - 1], t \in [j, j + m - 1]\} \quad (3.4)$$

Where:

- xxx is the input feature map
- (i,j)(i, j)(i,j) are coordinates in the output feature map
- n,mn, mn,m are the size of the pooling window
- s,ts, ts,t are the coordinates within the pooling window

Fig 3.11 illustrates the pooling operation.

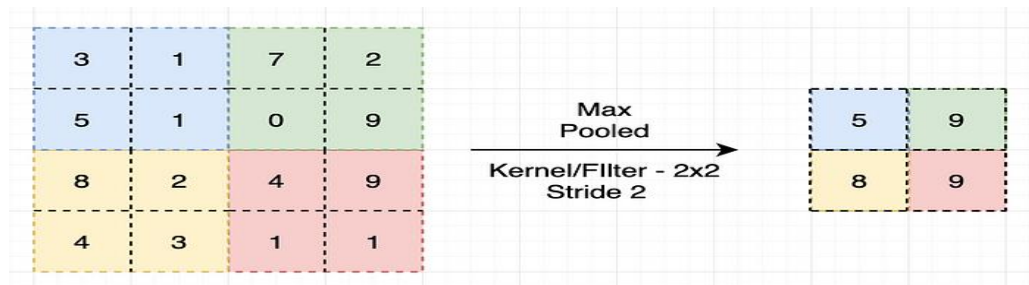


Fig 3.11: Pooling layer in CNN [33]

Finally, in the fully connected layer, each neuron is connected to all activations from the previous layer, enabling the network to make class predictions. To mitigate overfitting, dropout is applied during training, randomly disabling neurons, typically 50% of the network's neurons [34]. The operation of the fully connected layer is shown in Fig 3.12.

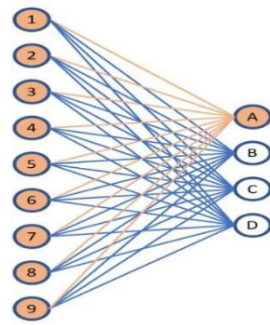


Fig 3.12: Fully connected layer in CNN [35]

3.3.2.2 SNN

SNN possesses two identical subnetworks alongside equal structure and shared parameters and weights. The similarity evaluation between two inputs depends on the comparison of their feature vectors through these networks. Traditional neural network models have different functionality because SNNs lack the capability to produce image classifications. The system analyses a probability value to establish image similarities [36].

An SNN contains two identical subnetworks that apply convolutional and fully connected layers to generate feature vectors from both images (A and B). The compared encodings $E(A)$ and $E(B)$ lead to Euclidean distance (L2 distance) computation which determines their similarity. Higher similarity values occur when the distance between these entities is brief and dissimilarity grows with longer distance measurements.

Through the L2 Distance Layer, the system determines the Euclidean distance between feature vectors to directly yield similarity scores. The goal-oriented nature of SNNs makes them practical tools for limited labelled case scenarios since their primary purpose is to measure similarities rather than perform image classification tasks. Fig 3.13 illustrates the SNN architecture.

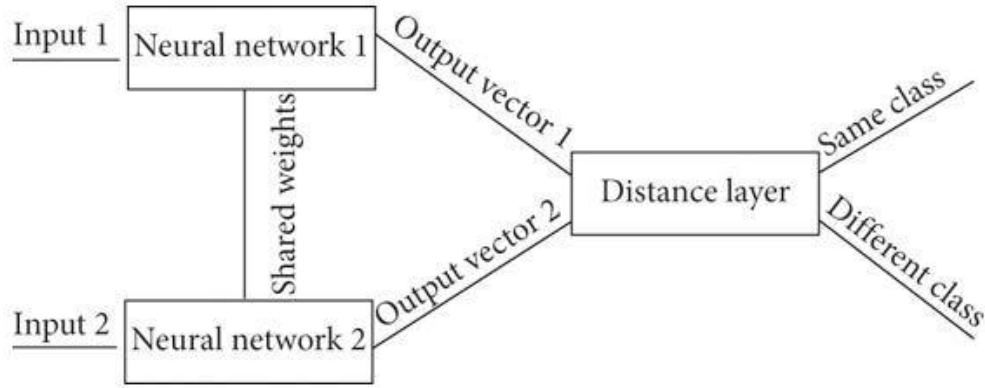


Fig 3.13: SNN Architecture [36]

3.3.3 Machine learning algorithms

In our current work on contactless palmprint and palm vein detection, our system incorporates an authentication mechanism utilizing advanced machine learning algorithms. Specifically, we have employed K-Means Clustering as an effective method for data clustering.

3.3.3.1 K-Means Clustering

K-Means operates as a common unsupervised learning algorithm that divides data collections into K-separate clusters through attribute alignment. The predefined value of K dictates the total number of clusters that will form in the partition. This modern method assists palmprint and palm vein detection by identifying various dataset groups in unlabelled samples without depending on any pre-existing exposure or training practices. The process repeats itself through data point clustering until optimal data groupings reach maximum accuracy. Common features emerge within groups of data points in clusters even though each cluster remains separate from other clusters. Functionally K-Means minimizes the total distances that connect data points to their assigned cluster centres [37].

Fig 3.14 illustrates the working of the K-Means clustering algorithm.

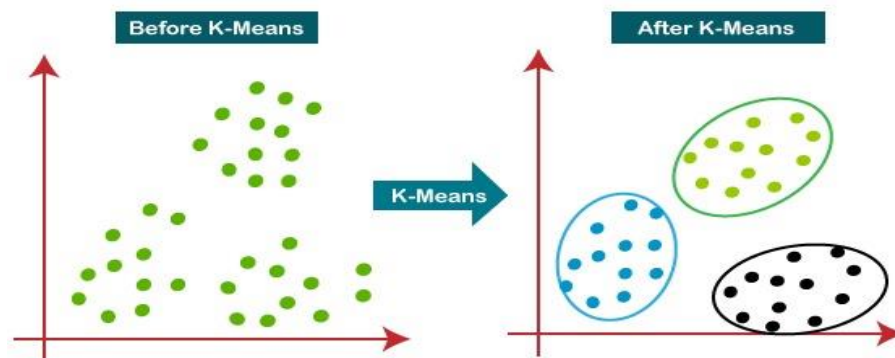


Fig 3.14: K-means clustering [37]

3.4 Methodology

For the palmprint recognition task in this project, the process is divided into five main steps, which are outlined in the flowchart shown in Fig 3.15. A detailed explanation of each step follows the flowchart.

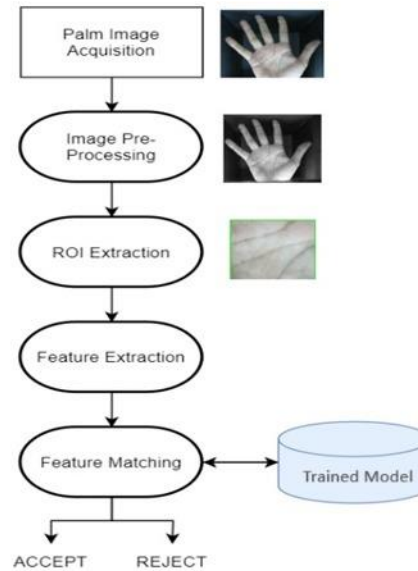


Fig 3.15: System flow chart

3.4.1 Palm Image Acquisition

In this step, obtaining a clear image of the palm is essential for accurate data analysis. The image should have high resolution to facilitate effective auto-encoding. The system ensures the image is loaded in a consistent format using specific image loading functionality.

3.4.2 Image pre-processing

Image pre-processing is essential for preparing the input images for analysis, ensuring their quality, and standardizing them for further operations. The images undergo resizing and normalization to bring the pixel values within a range of 0 to 1, making them suitable for deep learning models. To enhance the model's robustness and prevent overfitting, data augmentation techniques such as random rotations and resized crops are applied, increasing the variability of the training data. Additionally, the images are converted to grayscale to emphasize texture and pattern-based features, which are crucial for tasks like ROI extraction and feature extraction.

3.4.3 ROI Extraction

The system implements ROI extraction through deep learning methods which automatically detect critical areas in the palm. The method cuts specific part from the image which contain significant texture and patterns in order to concentrate on important features. The training process allows

convolutional layers of the model to identify significant regions. Through training, the model learns to focus on key areas that improve the accuracy of palmprint classification.

3.4.4 Feature Extraction

Following the identification of the ROIs, the subsequent step involves extracting features from these areas. Algorithms are employed to analyse the ROIs and capture distinctive characteristics, such as patterns, texture, shape, histograms, and angles, which facilitate differentiation between palm images. The feature extraction process is conducted through pre-trained layers that capture high-level patterns, followed by fully connected layers that generate embeddings to distinguish between various palm images. The application of a cross-entropy loss function during training ensures precise classification based on the extracted features.

3.4.5 Feature Matching

After the model has been trained, the extracted features are compared using a classification approach. During the training phase, the model is optimized on labelled palmprint datasets to enhance its ability to accurately match input images with their corresponding labels. During inference, the model predicts the class of the input palm image, and if the predicted class corresponds to an authorized user based on the trained model, the palm image is accepted. If there is no match, the image is rejected.

Chapter 4

Simulation Results

This chapter presents the results of the system simulation. It begins with the extraction of the ROI across the entire dataset, a fundamental step in data preprocessing. Next, the model training phase is discussed, followed by an overview of the feature-matching procedure.

4.1 ROI Extraction

We worked on a code that sets up a fully functional image processing and machine and deep learning flow optimized for a palmprint recognition system based on CNN. The operation of the system can be represented by a set of steps which include pre-processing of inputs that are the images, feature extraction, as well as classification of the images based on the palmprint. Here is an in-depth explanation of the pipeline:

1. Data Preprocessing

- **Image Loading and Display:** The pipeline starts with loading the image that the verification process will be applied on it. This image as shown in Fig. 4.1 is displayed with the help of the matplotlib package for the visualization of the results.



Original Image

Fig 4.1: Original image

- **Image Padding and Conversion:** This is done to standardize the size while working with the images as some of them may be of different sizes. After that, the image is transformed from RGB format to grayscale format as shown in Fig. 4.2 to reduce the complexity of the further processing steps.



Gray Scale Image

Fig 4.2: Grayscale image

- **Thresholding:** Otsu's thresholding technique is used to segment the image of the hand from the background and for this, the images are converted to binary format, the centre of the hand is defined which is important for the subsequent steps as shown in Fig. 4.3.

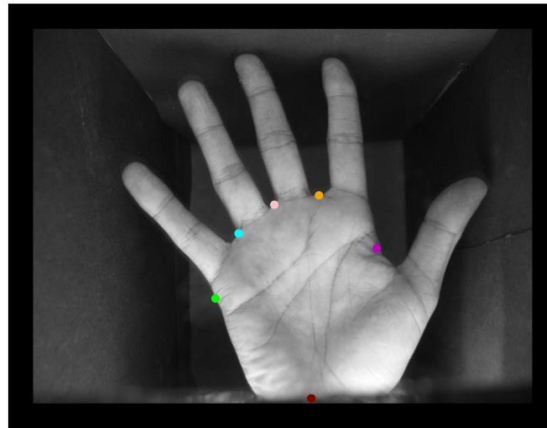


Centre of Hand

Fig 4.3: Center of hand

2. Feature Extraction

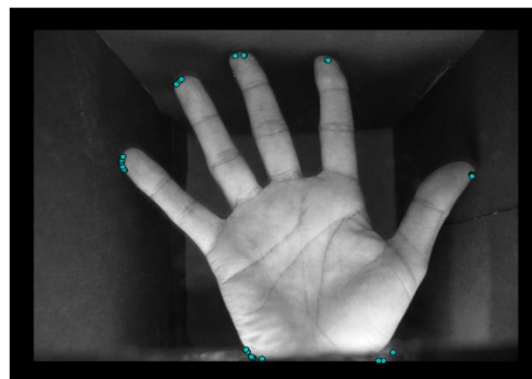
- Hand Contour Detection: based on erosion and boundary detection techniques the contour of the hand is then detected.
- Valley Points Detection: the valley points are the points between the fingers as shown in Fig. 4.4, these points are essential for setting the selected ROI and they are critical to explaining the hand's configuration.



All Valley Points

Fig 4.4: Valley points

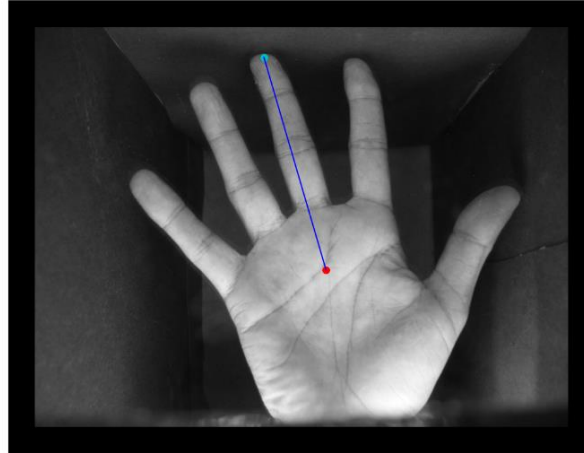
- Convex Hull and Angle Calculation: The convex hull of the hand is the outermost boundary of the hand. It is calculated to determine the fingertips, as shown in Fig. 4.5, Which in turn helps in aligning hand angles, computing rotation and extracting ROI.



Finger Tips

Fig 4.5: Fingertips

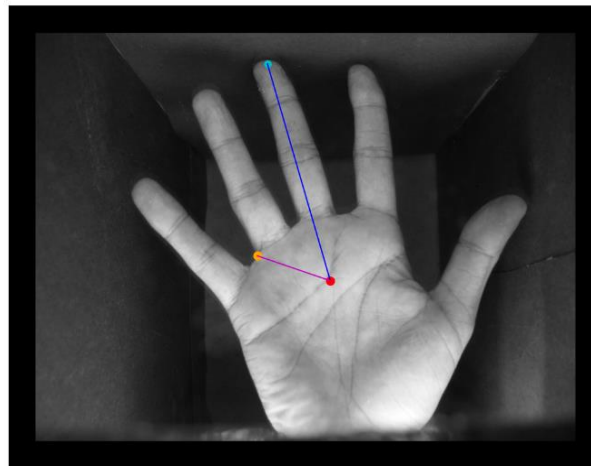
The angle between some or all of the valley points and the midpoint of the hand is computed to properly orient the hand. The middle fingertip is singled and taken as a reference while calculating other angles from it as shown in Fig. 4.6.



Middle Finger Tip

Fig 4.6: Middle fingertip

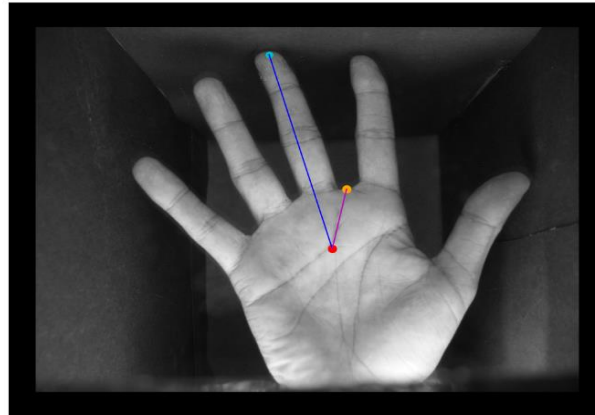
The first valley point is aligned with the centre of the hand and the middle of the fingertip, as shown in Fig. 4.7, in order to determine the designated number of rotations which is required for efficiency.



Valley Point 1

Fig 4.7: First valley point

The second valley point is applied for rotation and proper positioning of the hand as well as shown in Fig. 4.8.



Valley Point 2

Fig 4.8: Second valley point

Fig. 4.9 and Fig. 4.10 demonstrate the points that have been determined as the valley points after filtering and angle measurements which are used ROI determination.



Fig 4.9: Valley points

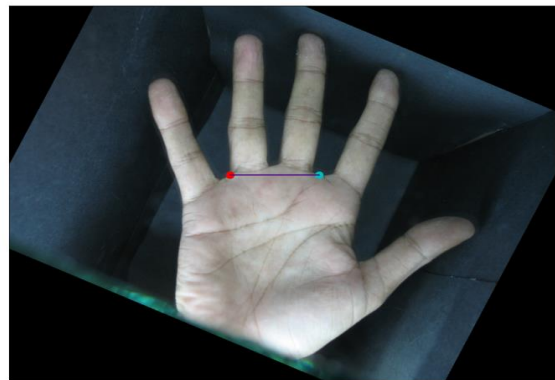


Image Rotation

Fig 4.10: Image rotation and valley point connection

- Region of Interest (ROI) Extraction: According to the valley points and convex hull traced here the ROI is obtained. This region is where most of the useful information in the hand is located – its primary elements are critical for gesture interpretation.

The ROI is extracted according to the aligned valley points with the maximum hand area included for palmprint recognition. Fig. 4.11 shows the extracted ROI of the palm.



Location of ROI

Fig 4.11: ROI image

4.2 Training model

Before training the model, the images undergo preprocessing using the LoG filter, which enhances image quality and reduces noise. Following this, feature extraction is performed using a Gabor filter bank, which captures detailed texture information from palmprint images, ensuring robust feature representation.

After preprocessing and feature extraction, several algorithms were tested to determine the most effective approach for the system. Two types of models were evaluated: a **retrieval-based model**, which identifies similar images from a stored dataset, and a **classification model**, which categorizes images into predefined classes. The goal of this evaluation was to identify the model that achieves the highest accuracy for adoption in the system.

1. Classification Models

Several deep learning architectures were evaluated for palmprint recognition purposes for the classification models. The evaluation included four different CNN models: AlexNet, ResNet, VGG16 and VGG19. These CNN networks were developed to identify images and execute computer vision tasks. The number of layers and convolution operations distinguishes these network architectures by depth from each other. A predefined parameter of 500 epochs enabled sufficient learning experiences during training before we assessed all models on the Palmprint ROI dataset. Table 4.1 presents

accuracy results and training times for all models.

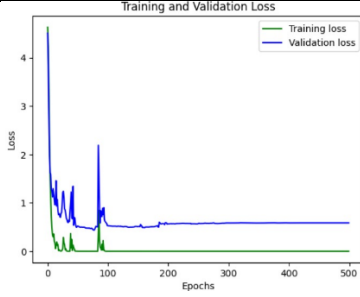
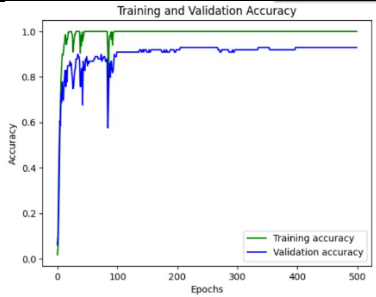
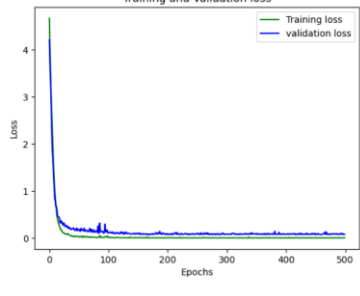
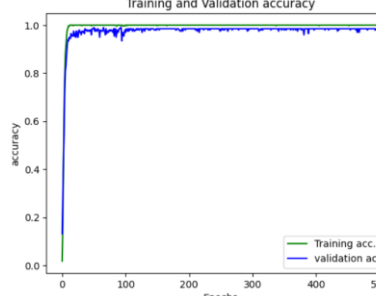
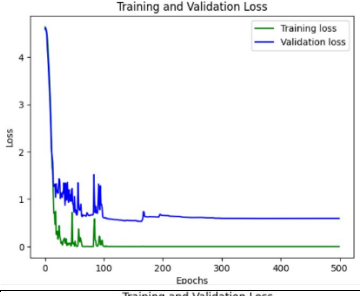
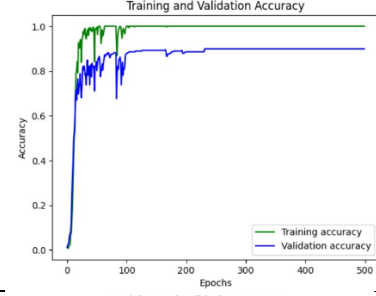
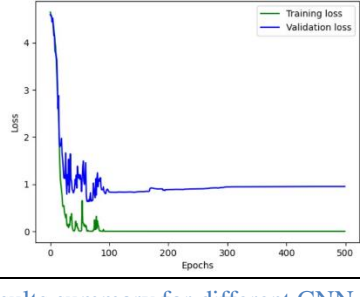
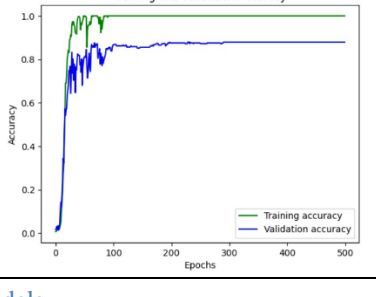
CNN Type	Accuracy	Taken Time	Training and validation Loss	Training and validation Accuracy
AlexNet	92.9293%	41m 25s		
ResNet	98.9899%	40m 22s		
VGG16	89.8990%	100m 16s		
VGG19	87.8788%	127m 47s		

Table 4.1: Results summary for different CNN models

2. Retrieval Model

A Siamese neural network was implemented for biometric authentication using full hand version from our Dataset for the retrieval-based model through similarity-based learning. The model utilizes VGG16 for feature extraction, generating embeddings that are compared using Euclidean distance (L2 norm) or cosine similarity. It was trained and tested on our full hand palmprint Dataset, with 300 epochs and early stopping to optimize performance. The retrieval model's performance was assessed,

achieving a validation accuracy of 94.1% and a training accuracy of 86%, as illustrated in Fig 4.12.

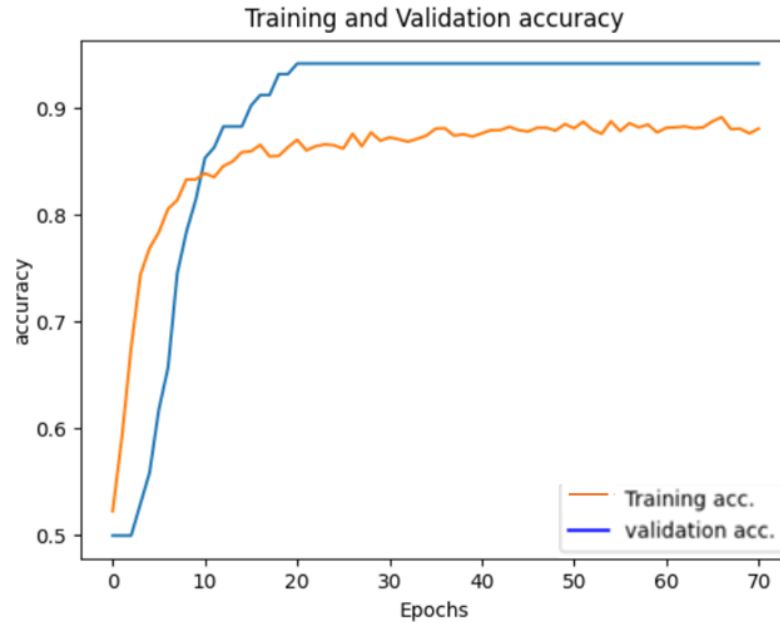


Fig 4.12: Retrieval model accuracy

The Resnet from classification-based models was the model with the highest accuracy so it was used to build our system. We gradually started building more than one model, the first model was trained by 20 university students, the second by 40, the third by 60, and so on until we reached 100 which is the same number of people in the open-source dataset. All the results for all the above attempts are shown in the Table 4.2.

Number of students	Accuracy	Taken Time
20	84.861%	13m 49s
40	87.935%	22m 3s
60	90.372%	35m 5s
80	93.452%	44m 46s
100	99.005%	63m 19s

Table 4.2: ResNet model on different numbers of data from dataset

As seen in Table 4.2, larger datasets enhance the accuracy of the ResNet CNN model; by providing more diverse examples for learning, which improves generalization and reduces overfitting. They also ensure better class representation, leading to more effective feature learning. Additionally, larger datasets yield more accurate gradient estimates, facilitating stable weight updates during training. However, this increased data size requires more computational resources and longer training times. Overall, while larger datasets improve model performance, they also come with the trade-off of increased training duration.

After collecting the initial dataset of 100 students, we continued gathering data until we reached 159 students, which became the final dataset used for training the model. The model was then trained on this dataset, completing training in 61 minutes and 36 seconds and achieving a best validation accuracy of 99.6689%, as shown in Fig. 4.13.

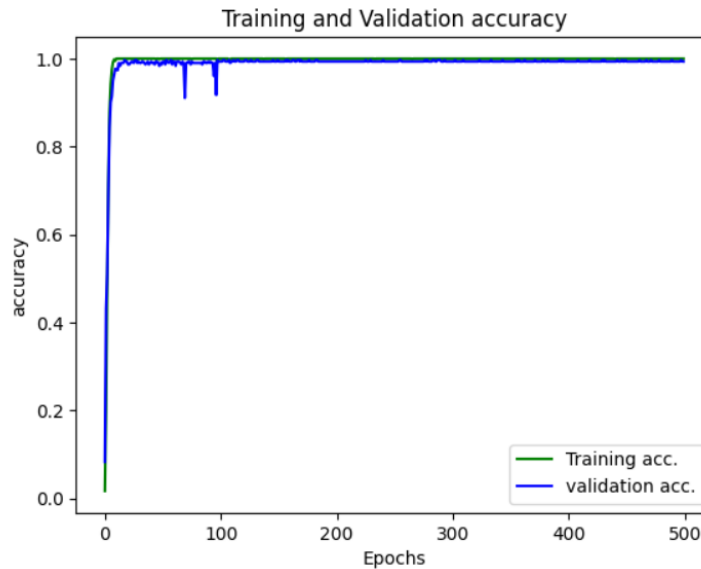


Fig 4.13: Model of 159 people from our dataset accuracy

During testing, as shown in Fig 4.14, the model demonstrated strong performance in identifying and authorizing images. Each test result included the actual class, predicted class, authorization status, and confidence level. The predictions were highly accurate, with confidence levels ranging from 0.95 to 1.00, highlighting the model's reliability and effectiveness in handling the palmprint recognition task.

Image: 1210100-4.bmp	Actual Class: 1210100	Predicted Class: 1210100	Authorized: Yes	Confidence: 1.00
Image: 1210345-4.bmp	Actual Class: 1210345	Predicted Class: 1210345	Authorized: Yes	Confidence: 0.95
Image: 1210904-4.bmp	Actual Class: 1210904	Predicted Class: 1210904	Authorized: Yes	Confidence: 0.96
Image: 1210827-4.bmp	Actual Class: 1210827	Predicted Class: 1210827	Authorized: Yes	Confidence: 0.99
Image: 1211061-4.bmp	Actual Class: 1211061	Predicted Class: 1211061	Authorized: Yes	Confidence: 1.00
Image: 1211234-4.bmp	Actual Class: 1211234	Predicted Class: 1211234	Authorized: Yes	Confidence: 1.00

Fig 4.14: Correct prediction based on Model of 159 people from our dataset

To explore the benefits of larger datasets, the model was later trained on a combined dataset of 159 samples from our dataset and 99 samples from the open-source Palmprint dataset. This expanded dataset improved the model's generalization by increasing diversity, exposing it to a wider variety of features, and reducing overfitting. Although this required more computational resources, the larger dataset significantly enhanced the model's ability to perform well on unseen data. As a result of its superior performance, this model was adopted for the system.

The model was trained using several parameters after a convolutional neural network model with its

layers was built. The model was assembled using the Adam optimizer, which is preferred for deep learning models and was considered to be the first parameter. It updated the model parameters based on the gradient of the loss function, which is the one that measures the difference between the expected probabilities and the real labels during the training process, and this was the second parameter. The model in this code was trained 500 times, which indicates the number of Epochs that was previously determined.

Model: "embedding"

Layer (type)	Output Shape	Param #
input_image (InputLayer)	(None, 128, 128, 3)	0
conv2d (Conv2D)	(None, 128, 128, 64)	6,976
max_pooling2d (MaxPooling2D)	(None, 64, 64, 64)	0
conv2d_1 (Conv2D)	(None, 64, 64, 128)	131,200
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 128)	0
conv2d_2 (Conv2D)	(None, 32, 32, 256)	295,168
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 256)	0
flatten (Flatten)	(None, 65536)	0
dense (Dense)	(None, 512)	33,554,944

Total params: 33,988,288 (129.66 MB)
Trainable params: 33,988,288 (129.66 MB)
Non-trainable params: 0 (0.00 B)

Fig 4.15: CNN model summary

Fig. 4.15 shows a CNN model summary which represents details about the CNN layers including the output shape and number of parameters of each layer type. This shows the data flow from the input layer to the dense layer through this model.

The number of parameters for convolutional layers is calculated using equation 4.1:

$$\text{Parameters} = (kw \times kh \times Cin + 1) \times Cout \quad (4.1)$$

Where:

- kw = kernel width
- kh = kernel height

- C_{in} = number of input channels
- C_{out} = number of filters (output channels)
- The “+ 1” accounts for the bias term for each filter [38].

And the number of parameters for the Dense layer is calculated using equation 4.2:

$$\mathbf{Parameters} = (\mathbf{input\ units} \times \mathbf{output\ units}) + \mathbf{output\ units} \quad (4.2)$$

Where:

- Input units= the size of the flattened input
- output units= number of neurons in the dense layer

As for the input, max pooling, and flattened layers, they do not have parameters to learn.

The performed layers are:

- The input layer takes a three-channel (RGB) image of 128x128 size.
- Three convolutional blocks extract the features from the input images. Each block consists of a convolutional layer followed by a pooling layer.

Each convolutional layer applies a different number of filters of different sizes with a ReLU activation function according to the layer number. The size of the input and the output images of these layers are still the same, and the number of parameters depends on the number of filters, the size of them, and the number of channels.

Each max pooling layer performs a 2x2 pool which reduces the size of the image by a factor of 2.

- Flatten layer flattens the 3D tensor into a 1D tensor of size 65536
- Dense layer is a fully connected layer with a sigmoid activation function and 512 units and 33554944 parameters [39].

The total number of trainable parameters is 33988299 which includes biases that will be learned through training which indicates a complex and large model.

The trained model is then evaluated based on testing data using several metrics such as Accuracy, Recall, Precision, and F1-Score. The values of this matrix are shown in Fig. 4.16. In general, these values indicate that the model works well on the test data.

	accuracy	macro avg	weighted avg
precision	0.996855	0.993750	1.000000
recall	0.996855	0.990625	0.996855
f1-score	0.996855	0.991667	0.997904
support	0.996855	318.000000	318.000000
Accuracy: 0.9969			

Fig 4.16: Performance measures for ResNet model on our dataset

4.3 Matching Features

The classification model's feature matching process begins with extracting contours from the captured images. To achieve this, techniques such as Canny edge detection and morphological operations are applied. These methods help identify and refine contours, ensuring the model focuses on the most relevant features of the image.

Once the contours are detected, the algorithm computes the convex hull and analyses convexity defects to identify distinct points within the largest contour, known as valley points. These valley points serve as key features that define the unique characteristics of the object in the image. By identifying these points, the model can establish correspondences between features in both the training and test images.

To enhance classification accuracy, the model evaluates the relationships between these key points. It compares the extracted features from the input images against the learned patterns in the training dataset. Based on this comparison, the model determines whether a user is accepted or rejected. If the detected features align with the predefined criteria from training, the user is classified as accepted; otherwise, they are rejected.

As for the retrieval model, Fig. 4.17 provides a visual representation of matching feature points between two images. Two images, one from the train image and one from the test image, are linked to the background image. The characteristic points of the train image are represented by red dots, while the equivalent of the test image is represented by blue dots. On the other hand, the green lines connect the same distinct points. Each line connects two distinct points, one from each image, where these points correspond to their symmetry in their similarity.

The ORB algorithm was used to extract distinct points from both images and to match these points, a brute-force matcher with Hamming distance was used.

Fig 4.17 in general shows a great match and similarity between the two images, and this was clear from the many green lines. The number of green lines can give a quick visual assessment of matching quality.

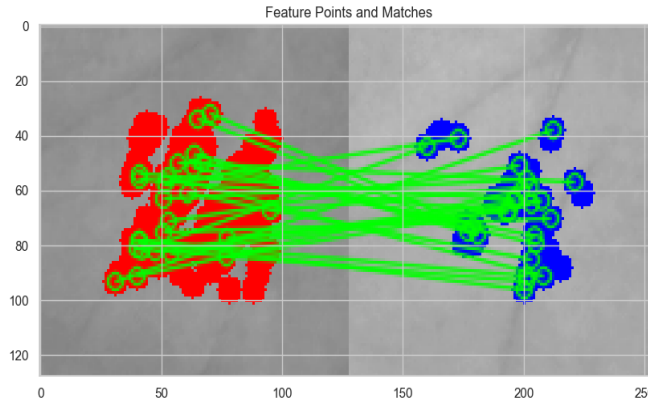


Fig 4.17: Feature Points and Matches

Fig. 4.18 shows a comparison between two image processing techniques, namely Otsu thresholding and K-means segmentation, which are widely used in the image processing process. These two techniques were used to demonstrate their effectiveness and the differences between them in dividing images into distinct regions.

Otsu Threshold Techniques: This technique, in its most basic form, generates a single intensity threshold that divides pixels into two classes: background and foreground. Minimizing the variance in intra-class intensity, or equivalently, maximizing the variance between classes, determines this threshold [40].

The equation of this technique:

$$\sigma_B^2(t) = \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2 \quad 4.3$$

where t : Threshold value, ω_0 , ω_1 : Weights of the two classes (foreground and background) and μ_0, μ_1 : Means of the two classes, and these are the inputs. And $\sigma_B^2(t)$ is the output which is the variance for the threshold t between classes [40].

K means Segmentation Technique: One of the most popular techniques for accurately classifying an image's pixels in a decision-oriented application is image segmentation. To achieve high contrast and high similarity between regions, it separates an image into several discrete regions. This technique uses a K-means clustering algorithm to segment the interest area from the background [41].

These techniques were applied to two images, one from the training set and the other from the testing set. The benefit of using Otsu thresholding on the training image was demonstrated in converting the grayscale image into a binary image while separating the background from the foreground. As for K-means segmentation, it is useful in dividing the image into regions effectively based on the intensity of the pixels in the image when applied to the same training image.

As for applying the two techniques to the testing image, it was for the purpose of seeing their effect on a different image, and this provided a visual understanding of how well the techniques generalized to new images.

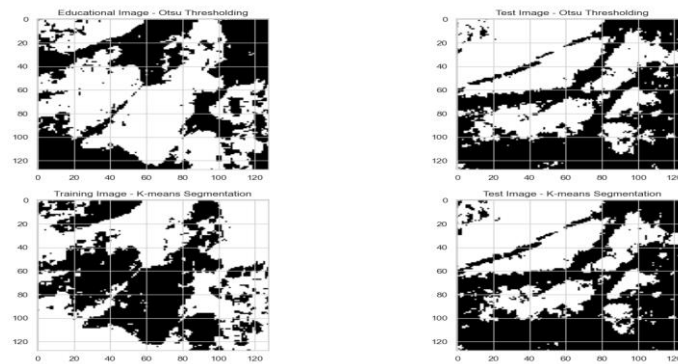


Fig 4.18: Comparison of Otsu Thresholding and K-means Segmentation on Training and Testing Images

Chapter 5

Conclusion, Challenges, and Future Work

5.1 Conclusion

This project worked to build an identity authentication system using contactless palmprint recognition for secure and faster verification operations. Systematic research and experimental procedures led to a complete grasp of palmprint-based biometric systems and multiple hardware and software configurations for increasing accuracy and reliability. The developed prototype tested successfully and delivered better outcomes than our predictions. The system performed recognition tasks effectively using palmprint images taken without physical contact.

The project reached its main goal by establishing a working prototype at “ka3kash” Laboratory which installed the system on the doors to enable practical demonstrations and real-world uses. The device demonstrates strong potential for broader authentication systems through its tested performance in usability and security measures while maintaining comfort for users.

5.2 Challenges

Our project development led to various challenges that enhanced our understanding of the best way to proceed. The project's original goal included palm veins and palmprint identification as detection components. The advanced infrared LEDs and cameras setup for vein imaging failed to meet expectations because the obtained images were not clear enough to work properly. As we cannot obtain a high-price multispectral camera intended for vein detection we decided to work on the palmprint only.

When we put all our efforts into palmprint recognition, we encountered challenges with the dimensions of the initial prototypes. The first prototype design failed to accommodate different hand sizes, leading to issues in accurately capturing images and correctly extracting regions of interest (ROI). Using the first prototype, we collected data from around 50 individuals and found that for 20 of them,

the ROI was incorrectly identified due to dimension mismatches. To address this, we developed and tested three prototype designs to determine the optimal size that accommodates various hand sizes and ensures accurate image capture for proper ROI extraction.

We experienced multiple system problems with Raspberry Pi 5 during the development period because of operating system issues that impacted device stability together with system performance.

The required troubleshooting and performance optimizations solved operational problems for reliable data collection. We adapted and perfected the system which resulted in building an effective contactless palmprint recognition prototype despite various obstacles.

5.3 Future work

Our future operations centre on making the contactless palmprint recognition system more efficient and scalable with an emphasis on budget-friendly palm vein detection implementation.

We will design a simple model using DWT for palmprint feature acquisition that pairs with a retrieval-based recognition system. Such an approach enables the system to manage additional users by performing new registrations without initiating repeated model training from scratch thus making it better for practical implementation. Although we did not use palm vein detection in the current prototype due to the high cost of specialized cameras, we plan to explore alternative methods for capturing vein patterns clearly. This could involve experimenting with customized lighting setups, optimized LED configurations, or image processing techniques that enhance visibility without requiring expensive equipment. The future development phase will concentrate on enhancing methods to store user data efficiently, ensuring quick retrieval and seamless operation. Currently, our device is designed to support palmprint recognition from the right hand, but in future iterations, we aim to extend the system's functionality to support left-hand palmprints as well. This will broaden the scope of the system and increase its usability in real-world scenarios, ensuring flexibility and robustness in various practical applications.

Our goal is to create an advanced contactless authentication system combining recent advancements in order to offer efficient security along with enhanced accuracy through palmprint and palm vein detection.

References

- [1] Y. Yang, Y. Zhou, R. Huang, Q. Liu, H. He, and X. Li, "Contactless Palmprint and Palm Vein Identity Recognition via a Bimodal Network with Parameter-adaptive Log-Gabor Convolution," 2023. [Online]. Available: <https://ssrn.com/abstract=4625638>
- [2] Rohit Khokher, Rajendra Kumar, and R C Singh, "Asystematic review of palm and dorsal hand vein recognition techniques," Dec. 2021, doi: 10.3306/AJHS.2022.37.01.100.
- [3] D. Zhang, W.-K. Kong, J. You, and M. Wong, "Online Palmprint Identification," Sep. 2003. doi: 10.1109/TPAMI.2003.1227981.
- [4] Y. Han, Z. Sun, F. Wang, and T. Tan, "Palmprint recognition under unconstrained scenes," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2007, pp. 1–11. doi: 10.1007/978-3-540-76390-1_1.
- [5] D. Hartung and C. Busch, "Why Vein Recognition Needs Privacy Protection," Dec. 2012.
- [6] G. Shah, S. Shirke, S. Sawant, and Y. H. Dandawate, "Palm vein pattern-based biometric recognition system," *International Journal of Computer Applications in Technology*, vol. 51, no. 2, pp. 105–111, 2015, doi: 10.1504/IJCAT.2015.068921.
- [7] L. Mirmohamadsadeghi and A. Drygajlo, "Palm vein recognition with Local Binary Patterns and Local Derivative Patterns," *2011 International Joint Conference on Biometrics, IJCB 2011*, Oct. 2011, doi: 10.1109/IJCB.2011.6117804.
- [8] J.-C. Lee, "A novel biometric system based on palm vein image," *Pattern Recognit Lett*, vol. 33, pp. 1520–1528, Sep. 2012, doi: 10.1016/j.patrec.2012.04.007.
- [9] W. Kang, Y. Liu, Q. Wu, and X. Yue, "Contact-Free Palm-Vein Recognition Based on Local Invariant Features," *PLoS One*, vol. 9, no. 5, pp. e97548–, May 2014, [Online]. Available: <https://doi.org/10.1371/journal.pone.0097548>
- [10] S. Elnasir and S. M. Shamsuddin, "Proposed scheme for palm vein recognition based on Linear Discrimination Analysis and nearest neighbour classifier," in *2014 International Symposium on Biometrics and Security Technologies (ISBAST)*, 2014, pp. 67–72. doi: 10.1109/ISBAST.2014.7013096.
- [11] W. Lu, M. Li, and L. Zhang, "Palm Vein Recognition Using Directional Features Derived from Local Binary Patterns," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 5, pp. 87–98, May 2016, doi: 10.14257/ijvip.2016.9.5.09.
- [12] S. Cho and K. A. Toh, *Palm-Vein Recognition Using RGB Images*. 2018. doi: 10.1145/3278229.3278239.

- [13] R. Hernández-García, R. J. Barrientos, C. Rojas, and M. Mora, "Individuals identification based on palm vein matching under a parallel environment," *Applied Sciences (Switzerland)*, vol. 9, no. 14, Jul. 2019, doi: 10.3390/app9142805.
- [14] Y. Pititheeraphab, N. Thongpance, H. Aoyama, and C. Pintavirooj, "Vein pattern verification and identification based on local geometric invariants constructed from minutia points and augmented with barcoded local feature," *Applied Sciences (Switzerland)*, vol. 10, no. 9, May 2020, doi: 10.3390/app10093192.
- [15] W. Wu, Y. Li, Y. Zhang, and C. Li, "Identity Recognition System Based on Multi-Spectral Palm Vein Image," *Electronics (Switzerland)*, vol. 12, no. 16, Aug. 2023, doi: 10.3390/electronics12163503.
- [16] Dr. Zhenan Sun, "Note on CASIA Multi-Spectral Palmprint Database." Accessed: May 10, 2024. [Online]. Available: <http://biometrics.idealtest.org/>
- [17] "IIT Delhi Touchless Palmprint Database," Oct. 2007, Accessed: May 11, 2024. [Online]. Available: https://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Palm.htm
- [18] H. Ghael, H. Dipak Ghael, L. Solanki, G. Sahu, and A. Professor, "A Review Paper on Raspberry Pi and its Applications," *International Journal of Advances in Engineering and Management (IJAEM)*, vol. 2, p. 225, 2008, doi: 10.35629/5252-0212225227.
- [19] M. Fezari, A. Al Dahoud, M. Fezari, and A. Al-Dahoud, "Raspberry Pi 5 : The new Raspberry Pi family with more computation power and AI integration," Nov. 2023, doi: 10.13140/RG.2.2.13547.52009.
- [20] "Raspberry Pi 5 features improved image processing - Geeky Gadgets," Oct. 2023, Accessed: Jun. 10, 2024. [Online]. Available: <https://www.geeky-gadgets.com/raspberry-pi-5-features-improved-image-processing/>
- [21] "Raspberry Pi NoIR camera marker tracking _ DreamOnward," Oct. 2019, Accessed: Jun. 10, 2024. [Online]. Available: <https://dreamonward.com/2019/10/16/picamera-exploration/>
- [22] "Raspberry Pi NoIR Camera V2 8MP - Bastelgarage Electronics Online Store", Accessed: Jun. 10, 2024. [Online]. Available: <https://www.bastelgarage.ch/raspberry-pi-noir-camera-v2-8mp>
- [23] C. L. Lin and K. C. Fan, "Biometric verification using thermal images of palm-dorsa vein patterns," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 2, pp. 199–213, Feb. 2004, doi: 10.1109/TCSVT.2003.821975.
- [24] "Infrared Thermography - ZfP - BayernCollab," Jun. 2015, Accessed: Jun. 30, 2024. [Online]. Available: <https://collab.dvb.bayern/display/TUMzfp/Infrared+Thermography#:~:text=The%20Stefan%20Boltzmann%20law%20gives,emission%20capacity%20of%20an%20object>
- [25] "Micro SD Cards Market Size, Trends and Forecast 2031 with Top Countries Data _ LinkedIn", Accessed: Jun. 27, 2024. [Online]. Available: <https://www.linkedin.com/pulse/micro-sd-cards-market-size-trends-forecast-2031-top-countries-2xxbe/>

- [26] “New Diagnostic Forensic Protocol for Damaged Secure Digital Memory Cards _ IEEE Journals & Magazine _ IEEE Xplore”, Accessed: Jun. 27, 2024. [Online]. Available: <https://ieeexplore.ieee.org/document/9733364>
- [27] “Infrared Radiation _ Definition, Uses & Examples - Lesson _ Study.com,” Nov. 2023, Accessed: Jun. 28, 2024. [Online]. Available: <https://study.com/academy/lesson/infrared-radiation-definition-uses-effects.html>
- [28] “A Hands-On Guide to Document Image Classification _ by Babina Banjara _ Artificial Intelligence in Plain English,” Nov. 2023, Accessed: Jun. 28, 2024. [Online]. Available: <https://ai.plainenglish.io/a-hands-on-guide-to-document-image-classification-daed233ebee>
- [29] “What is the mathematical formula of the convolution operation on a 2D image_ - EITCA Academy,” May 2024, Accessed: Jun. 30, 2024. [Online]. Available: <https://eitca.org/artificial-intelligence/eitc-ai-adl-advanced-deep-learning/advanced-computer-vision/convolutional-neural-networks-for-image-recognition/what-is-the-mathematical-formula-of-the-convolution-operation-on-a-2d-image/>
- [30] “Convolutional Layers vs Fully Connected Layers _ by Diego Unzueta _ Towards Data Science,” Dec. 2018, Accessed: Jun. 29, 2024. [Online]. Available: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [31] G. Yang, Q. Zhang, and G. Zhang, “EANet: Edge-aware network for the extraction of buildings from aerial images,” *Remote Sens (Basel)*, vol. 12, no. 13, Jul. 2020, doi: 10.3390/rs12132161.
- [32] “CNN _ Introduction to Pooling Layer - GeeksforGeeks,” Apr. 2023, Accessed: Jun. 30, 2024. [Online]. Available: <https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>
- [33] “Pooling-Layer (Albawi et al., 2017) _ Download Scientific Diagram”, Accessed: Jun. 29, 2024. [Online]. Available: https://www.researchgate.net/figure/Figure-3-Pooling-Layer-Albawi-et-al-2017_fig1_349054980
- [34] Y. Jia *et al.*, “Caffe: Convolutional Architecture for Fast Feature Embedding,” Jun. 2014, [Online]. Available: <http://arxiv.org/abs/1408.5093>
- [35] “Chapter 2_ Convolutional Neural Networks — Unveiling Hidden Patterns in Images _ by Issac kondreddy _ Medium,” May 2023, Accessed: Jun. 30, 2024. [Online]. Available: <https://medium.com/@issackondreddy/chapter-2-convolutional-neural-networks-unveiling-hidden-patterns-in-images-b4574d34f556>
- [36] Rinki Nag, “A Comprehensive Guide to Siamese Neural Networks _ by Rinki Nag _ Medium,” Nov. 2019, Accessed: Jun. 30, 2024. [Online]. Available: <https://medium.com/@rinkinag24/a-comprehensive-guide-to-siamese-neural-networks-3358658c0513>
- [37] “K-Means Clustering Algorithm - Javatpoint”, Accessed: Jun. 28, 2024. [Online]. Available: <https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning>

- [38] “How to calculate the number of parameters in CNN_ - GeeksforGeeks,” May 2024, Accessed: Jun. 21, 2024. [Online]. Available: <https://www.geeksforgeeks.org/how-to-calculate-the-number-of-parameters-in-cnn/>
- [39] Yagesh Verma, “What is Dense Layer in Neural Network_,” Jun. 2024, Accessed: Jun. 30, 2024. [Online]. Available: <https://analyticsindiamag.com/topics/what-is-dense-layer-in-neural-network/>
- [40] “Otsu’s method - Wikipedia,” Apr. 2024, Accessed: Jun. 25, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Otsu%27s_method
- [41] N. Dhanachandra, K. Manglem, and Y. J. Chanu, “Image Segmentation Using K-means Clustering Algorithm and Subtractive Clustering Algorithm,” in *Procedia Computer Science*, Elsevier, 2015, pp. 764–771. doi: 10.1016/j.procs.2015.06.090.