Designing a stand alone finger vein scanner

Bachelor Thesis

M. van Elderen
Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente
Enschede, The Netherlands

***Abstract*—Finger vein scanner are a new biometric form currently being developed. Multiple commercial devices are already on the market, but their performance is not backed up by publicly available research. The University of Twente created a finger vein scanner for academic purposes. This reports details redesigning said system into an embedded system. By integrating the user interaction onto the device itself with the help of a touchscreen and a new Graphical User Interface. Alongside this, a new method of LED adjustment is tried for better image quality. The performance of the device is tested against the first version and was found to be great in speed, but the accuracy of the device was left to be desirable.**

***Index Terms*—Biometric, Vascular pattern, Finger vein, Near Infrared, NIR**

I. Introduction

Biometrics is identifying someone based on their physical properties, behaviour or the traces they leave behind. An example of this is the fingerprint scanner found on most smartphones these days. However the fingerprint scanner among others scanners have a big caveat namely that you leave them behind everywhere. Or in the case of surface area detection, the biometric can change over time.

Here the finger vein scanner comes into play. By making use out of the veins in the finger, you are not leaving it behind everywhere like your fingerprint. Next to this it is not effected by the skin conditions [1]. Making it a more secure biometric. Over the past few years research has been done on both finger vein algorithms as capturing devices [2]–[5]. And even the first few devices became commercially available.

The University of Twente has also developed a finger vein scanner, however it currently uses external hardware for vein extraction and to display the user interface [6]. By integrating a touchscreen within the devices itself, it will become an embedded system. The added advantage is that it will become easier to carry around. Further by including a simple user interface it should become more accessible to other users.

The goal of this project is to create an all-in-one device which at has least the following:

1) enrolment, disenrollment and verification functionality
2) Touchscreen for user interface
3) Live video feed
4) Same performance as the first iteration

In section II a quick look is taken into standalone finger vein scanner are currently available on the commercial market and a few academic devices. Afterwards in section III the algorithms used for vein extraction and matching together. Afterwards in section IV the hardware and software design is made. Following this the design is compared to the first iteration of the University of Twente design in section V and discussed in section VI. The paper ends with a conclusion and recommendations in section VII and section VIII respectively.

II. Related Work

A. Commercial devices

Currently there are multiple options for finger vein scanners on the market. The ones discussed here will all be for security and are mostly stand alone.

Fig. 1: Hitachi SecuaVeinAttestor authentication terminal [7]

In Fig. 1 the Hitachi SecuaVeinAttestor is displayed. This device has two parts to it, namely the authentication and the registration terminal. It has a false acceptance rate (FAR) of 0.0001% and a false rejection rate (FRR) of 0.01%, with an authentication time of 0.8 seconds. However it has limited feedback to the user as it only has a LED and 16x2 LCD screen.

Fig. 2: EverFocus EBC980R [8]

Fig. 2 shows the EverFocus EBC980R, which has an web application for extra user option. The web application is however not necessary for basic operation. With its 0.0000067% FAR it has the lowest FAR out of this list [8]. However this comes at the cost of a slightly longer authentication time of 2 seconds. The screen, which is used for user feedback, has a range of 4x20 characters.

Fig. 3: Korecen FV-100 [9]

Fig. 3 is device which has no extras to it meaning that it is a fully stand alone device. It has a FRR of 0.05%, while having a FAR of 0.0001% [10]. While also having a 1 second authentication time. The user feedback and input for this device is fully done by the 5 inch touchscreen display.

Fig. 4: ZKTeco FV350 [11]

Fig. 4 displays the ZKTeco FV350, which is a hybrid between fingerprint and finger vein scanner. The acceptance and rejection rate could not be found by the author, but it has a authentication rate of up to 2 seconds. Both feedback to the user as input from the user is done by the attached 2.8 inch screen.

TABLE I: Overview of commercial devices

Device	Standalone	FAR/FRR [%]	Authentication time
Hitachi SecuaVeinAttestor	No	FAR=0.0001 FRR=0.01	0.8 seconds
EverFocus EBC980R	Yes	FAR= 0.0000067	2 seconds
Korecen FV-100	Yes	FAR=0.0001 FRR=0.05	1 seconds
ZKTeco FV350	Yes	Unkown	2 seconds

In Table I a comparison is made between all mentioned devices and their stand alone capabilities, FAR, FRR and authentication time.

B. Academic Devices

Within the academic world, researchers usually build their own finger vein capturing device. It should be noted that it just captures them and the processing happens on another device. Then build software to extract the veins or look at how the scanner itself compares to another.

For example Huang et al. build the device shown in Fig. 5. Which is a fully enclosed device except for the cutout where the user is supposed to put in their finger. With the little feedback the device has the user can not be sure if the finger is placed correctly. To counteract this any deviations in translation or rotation are handled within the software.

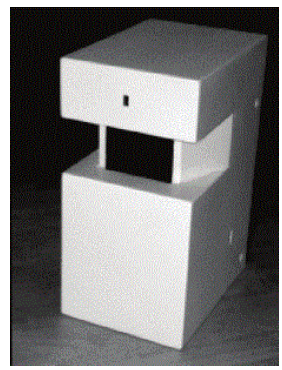


Fig. 5: Huang Scanner [12]

The Dongguk University build the finger vein capturing device shown in Fig. 6. The user has a slight obstruction of their view of the finger. But the device has edges on one side and at the end to help users place their fingers the same way on repeated uses. In the software it still has some algorithms to help counteract the translation errors.

Fig. 6: Dongguk University [13]

The University of Twente has multiple finger vein capturing devices, however the one shown in Fig. 7 is the latest version. By including the LED strip on the top the user has partial view of their finger. However just like the Dongguk University device it has a way to help the user place their finger in a more consistent way by placing the finger in rounded edges. The software helps the translation and rotation errors in one direction.

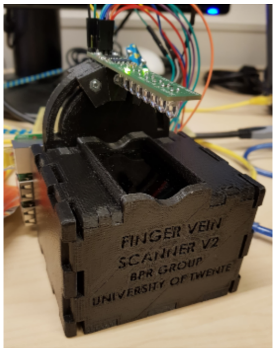


Fig. 7: University of Twente Scanner without LED cover [6]

III. Method

In this section the system overview as displayed in Fig. 8 is discussed. Each function block, except the user interface, will be explained.

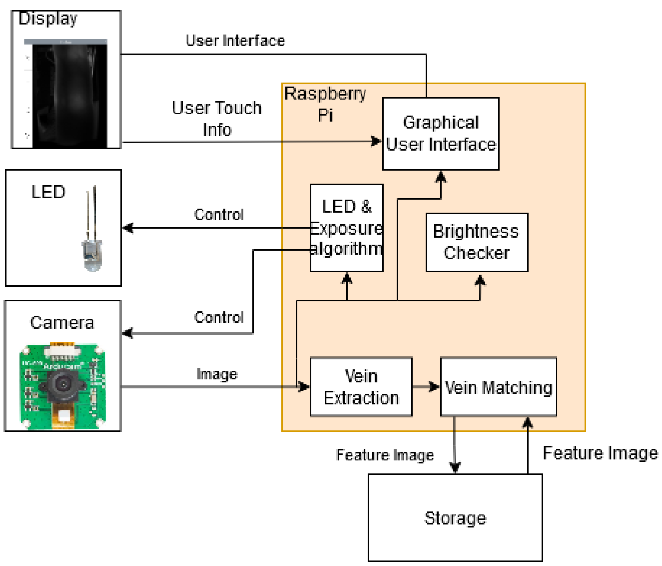


Fig. 8: Architecture overview

A. Vein extraction

To extract the veins from a image it needs to go through a few steps all of which are shortly explained.

1) Lee Region: This paper makes use out of the region of interest described by Lee et al. [14]. Due to changes of finger shapes across people, the finger area needs to be identified. The finger area is brighter than the background due to the IR light shines through it. By using a mask along the length of the finger, the position at which the masking value becomes maximal is said to be the transition between background and the finger area.

2) Huang Normalise: The normalisation method used in this paper is described by Huang at el. [12]. However it only compensates for the rotation in the xy-plane. By attempting to fit a straight line in between the middle of the detected finger edges. And making it orthogonal to the y-axis of the image.

3) Miura Max Curvature: Naoto Miura proposed an algorithm for extracting finger-vein patterns called max curvature [3]. By looking at the brightness curvature of the image in different directions the veins centre can be found. And afterwards connecting these centres together or not depending on the neighbouring pixels.

B. Miura Match

Miura Matching as described in [2] is an biometric matching method. It is in fact just a correlation. By trimming the reference image slightly and moving it around to find the maximum resembles. Then where both images have veins is counted and divided by the total amount of vein pixels across the reference and input image.

C. LEDs & Exposure

Miura Max Curvature is not affected by vein brightness [3], since it makes use of the brightness curve. Meaning that the individual areas do not have to be as specific for optimal performance as done by Ton [4]. To increase performance speed all LEDs are set to the same brightness. However fingers across humans do not have the same diameter. So to set the LED's PWM, it looks at the difference in between the preferred gray value and the average current gray value. This difference is than added to the current value according to Equation 1, where G stands for average gray value.

$$PWM_{new} = PWM_{old} + 15(G_{wanted} - G_{actual}) \quad (1)$$

If the desired value can not be reached by only changing the LED, the camera exposure time changes according to Equation 2.

$$Exposure_{new} = Exposure_{old} + (G_{wanted} - G_{actual}) \quad (2)$$

D. Brightness Checker

Also by making use out of the average gray value on the center of the image, it can be determined if a finger is present by using Equation 3. This works when the combination of LED PWM and exposure time is set in in such a way that it is bright if there is no finger present, but does the light does not pass through the finger when it is.

$$Finger_{present} = logical(G_{actual} < threshold) \quad (3)$$

IV. Design & Implementation

The new finger vein scanner should full fill some requirements both from a hardware and software side before being able to say it is finished.

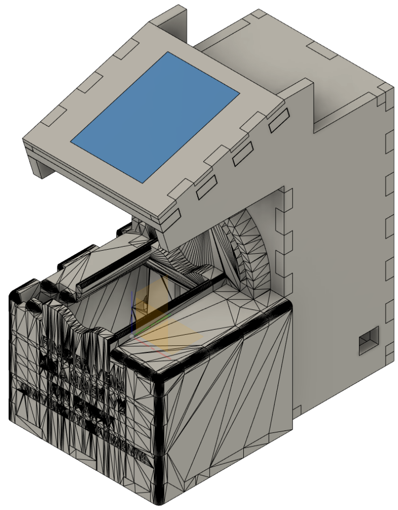


Fig. 9: 3D model of stand alone finger vein scanner

A. Hardware

The second iteration of the finger vein scanner at the University of Twente (as shown in Fig. 7), was quite compact and easy carry around. However to utilise it the user needed a external screen and mouse. To improve the easy of use of the device, a touchscreen will be added.

For the screen the PiHat 2.4” display is chosen, because of its smaller size but relatively high pixel count (320x240). By adding it to the SPI protocol of the Raspberry Pi it does not interfere with other attached modules.

The screen, Raspberry Pi 4B and power board need to be kept in place over time. Due to this a housing needs to be designed. The resulting 3D model attached to the previous scanner is shown in Fig. 9.

The screen is placed above the LED strip to prevent unwanted user input during finger placement and due to having a camera feed on the screen the user can see where they are placing their finger. Further is the screen attached under an angle to allow for better view ability when standing in front of it.

B. Software

In Fig. 10 the flow diagram is shown (full version is shown in Fig. A.1), from which can be seen that the user has three options. These are add and remove themselves to the system and verify the user in the system on a open set biometric.

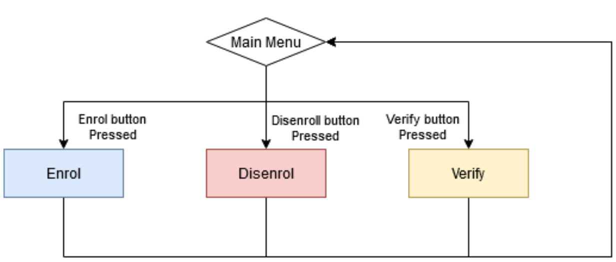
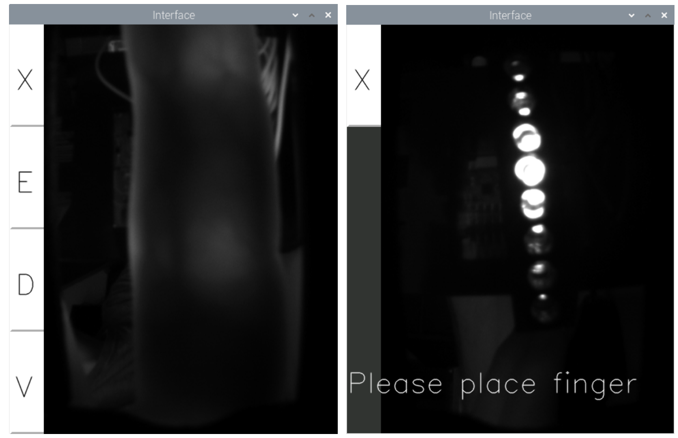


Fig. 10: Short version flow diagram



(a) Main Menu (b) enrolment menu

Fig. 11: Graphical user interface

Next to this another thing is noted. Namely that the user can not see if their finger is in view of the camera. To work around this a live feed of the camera should be added to the graphical user interface (GUI). For a better human-technology interaction literature has shown that there are a few guidelines which should be followed [15]–[17]. These are, but not limited, to the following.

1) **Simple and natural dialogue** all information should be concise and follow a logical order.
2) **Speak user language** The text given to the user should be in words, phrases and concepts familiar to the user rather than using technical term.
3) **Minimize user memory load** The user should not have to remember information and instructions should be accessible at all times.
4) **Be consistent** Users should not wonder if different words have the same meaning. Furthermore the same user action should result in the same system action.
5) **Provide feedback** At all times the system should let the user know what it is doing within at least one second, otherwise the user’s flow of thought will be interrupted [18].
6) **Clearly mark exits** The user needs to be able to go to a previous state if wanted.
7) **Error prevention** A careful design prevents a problems from occurring.

The graphical user interface as showcased in Fig. 11a shows the user options and their finger, which is reflecting the users finger. The buttons in the screen might seem small, they are extended beyond the picture to enhance human comfort. In the menu X is the quit button, E stand for enrolment, D means disenrollment and V stands for verification. When selecting one of the options on the main menu, dialogue is put on the screen to guide the user through the process or provide feedback. Which can be seen in Fig. 11b. To prevent user

error a few things are done by the scanner itself. This includes taking a picture if necessary. This is all written within C++ to allow for a fast response time.
During the enrolment process the user is prompted to place their finger twice. This is the enrolment checker method used. By matching the extracted features from both tries indicates if the system will match them later on as well. This process uses the same threshold as when verifying. If there is enough similarity the user is placed in the database with the first picture, otherwise the user is required to go through the process one more time.

V. Testing & Results

Designing a new system is all well, but it should also be functional. To test for this performance is tested against the first iteration in both a speed as finger recognition.

Starting out with testing the speed, Fig. 12 shows the average speed over 64 images in between algorithms from both versions. From which becomes clear that stand alone version is faster overall. On average it will take the stand alone version 1.9 seconds to go through all algorithms, while the first iteration does it in 5.5 seconds. The biggest time difference is in the Miura Max Curvature algorithm.

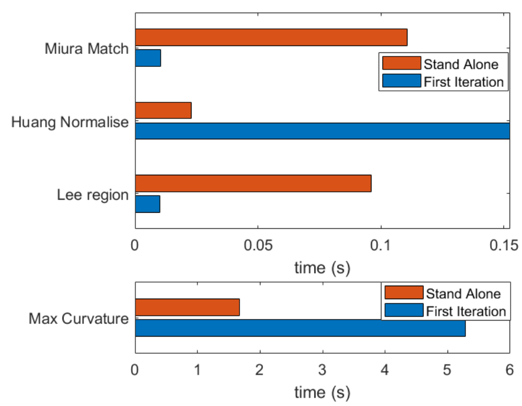
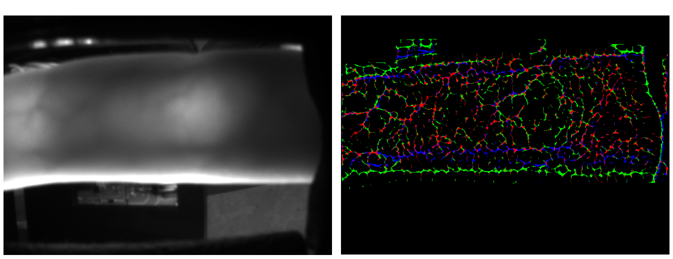


Fig. 12: Speed per function

Speed means nothing without results. Therefor Fig. 13 shows a comparison of extracted veins from Fig. 13a from both versions.



(a) Original Image (b) Extracted veins comparison

Fig. 13: Veins extracted

Within Fig. 13b red means where both identified veins, green where the first version only detected veins and blue for the stand alone version.

Matching the extracted vein shown in Fig. 13b against all other 63 feature image results in Fig. 14. The highest 10 scores are from images using the same finger. From which can be seen that all images except the same image have a higher matching score using the stand alone matching algorithm.

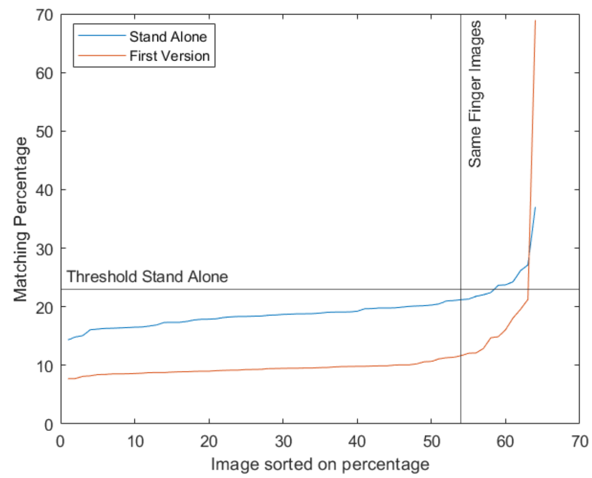


Fig. 14: Matching From Single Image

All image captured for this section used the LED & Exposure algorithm. Its affect on the centre image gray value are displayed in Fig. 15. From which can be seen that the image does indeed reach a satisfactory gray value, if it can go through enough iterations. The biggest difference can be found in the bottom region of the image, where the first version picks up the edge of the finger, while the stand alone version picks the overexposed part.

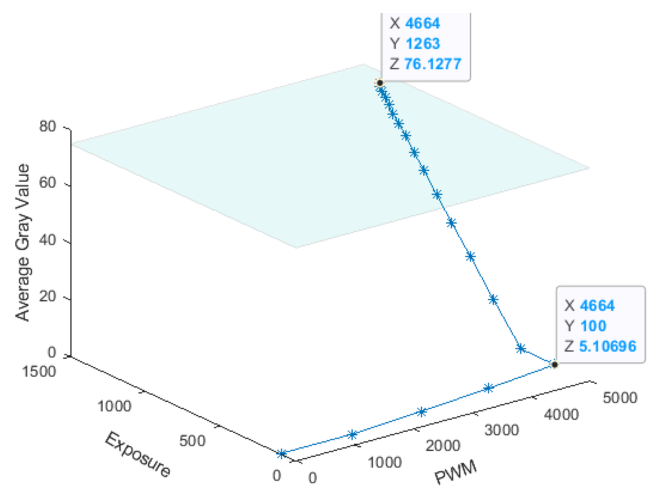


Fig. 15: LED and Exposure time

Fig. 15 makes it clear that only a combination of LED PWM and exposure time can reach a satisfactory image.

VI. Discussion

When images pass the Lee Region function on the stand alone version. Their image does not actually normalize. Instead they keep their original properties. This is will reduce the overall match rate of the device. Since rotations are not handled in this case.

Further the wires going through the screen are highly reflective of the IR light. If these are in view of the camera they will become a point which is picked up by the lee region algorithm. Next to the wires the 3D printed parts of the capturing device also have the tendency of being picked up as a finger edge.

The screen has an ideal form factor for this application, however it has a limited view angle. Which in some cases can make it hard to see what is exactly displayed on the screen. This made it such that the preview of the finger became bigger, but the button have a smaller appearance.

Another disadvantage the screen is that it receives data from the Pi over the SPI protocol using framebuffers. Doing it this way lowers the frame per seconds what the display can handle. Such that in some cases important frames are missed. Lastly the Raspberry Pi is connected using jumper wires to a power and data board. Making it that the Pi is in low power mode, which in turns decreases the speed performance of the device.

VII. Conclusion

The goal was to was to make a embedded finger vein scanner, which had at least the following specifications enrollment, disenrollment and verification functionality, an integrated touchscreen with live feedback and that it had the same performance as the first version created by the University of Twente. The first three points were fully accomplished. To test the device it created a small dataset to compare the performance to the first version. The performance from a speed view point as well, while the matching rate of the device seems to be less reliable. Small changes in the hardware of the device as well as the vein extraction are needed before the system can be considered finished.

VIII. Recommendations

One of the topics which need to be researched which can improve finger vein scanners overall is looking at the exposure time of the camera. A lot of research is done on LED brightness and its affect on the vein image, but little could be found on exposure time. This might result in an less noisy image.

For direct improvement on this project, the camera position needs to be looked at. The current camera (OV9281) needs to be placed at least 2cm beneath the finger to capture the full area. Further the focus point of the camera needs to be investigated for an optimal image acquisition. For further size reduction the minimum finger area for a unique and identifiable biometric can be researched. Currently the design has a big finger area, whose size might not be necessary.

Lastly further developing the GUI to allow for additional live feeds for example detected veins can help visualise the result of rotation along the axis parallel to the length of the finger.

References

[1] B. Prommegger, C. Kauba, and A. Uhl, "Multi-perspective finger-vein biometrics," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems, BTAS 2018*, 2018.

[2] N. Miura, A. Nagasaka, and T. Miyatake, "Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification," *Machine Vision and Applications*, vol. 15, pp. 194–203, 2004. [Online]. Available: https://www.semanticscholar.org/paper/Feature-extraction-of-finger-vein-patterns-based-on-Miura-Nagasaka/1099d25cc99834487b24de1e1541976bb96279da

[3] ——, "Extraction of finger-vein patterns using maximum curvature points in image profiles," in *Proceedings of the 9th IAPR Conference on Machine Vision Applications, MVA 2005*, 2005, pp. 347–350. [Online]. Available: https://www.semanticscholar.org/paper/Extraction-of-Finger-Vein-Patterns-Using-Maximum-in-Miura-Nagasaka/85c3f43392598a2029ceda1bb2b5ea1d5236555f

[4] B. Ton, "Vascular patern of the finger: biometric of the future? Sensor design, data collection and performance verification," 7 2012. [Online]. Available: http://essay.utwente.nl/61963/

[5] A. R. Syafeeza, L. H. Kwan, K. Syazana-Itqan, N. A. Hamid, W. H. Saad, and Z. Manap, "A low cost finger-vein capturing device," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 5, pp. 3330–3335, 2016.

[6] S. P. Rozendal, "Redesign of a finger vein scanner," 2 2018. [Online]. Available: http://essay.utwente.nl/75948/

[7] asmag, "8 finger and palm vein readers that offer superior access control security," 2020. [Online]. Available: https://www.asmag.com/showpost/31991.aspx

[8] EverFocus, "EBC980R." [Online]. Available: https://www7.everfocus.com/product/ebc980rm/

[9] vingerafdrukslot, "FV-100 vinger scanner." [Online]. Available: https://www.vingerafdrukslot.com/product/fv-100-vinger-scanner/

[10] Biometric-Locks, "FV100 Finger vein scanner." [Online]. Available: https://biometric-locks.eu/artikel/FV100?lang=en&returnto=%2Fartikel%2FREXINDOOR%3Flang%3Den

[11] ZKTeco, "FV350." [Online]. Available: https://zkteco.eu/products/access-control/standalone-device/hybrid-biometric/fv350

[12] B. Huang, Y. Dai, R. Li, D. Tang, and W. Li, "Finger-vein authentication based on wide line detector and pattern normalization," in *Proceedings - International Conference on Pattern Recognition*, 2010.

[13] T. D. Pham, Y. H. Park, D. T. Nguyen, S. Y. Kwon, and K. R. Park, "Nonintrusive finger-vein recognition system using NIR image sensor and accuracy analyses according to various factors," *Sensors (Switzerland)*, 2015.

[14] E. C. Lee, H. C. Lee, and K. R. Park, "Finger vein recognition using minutia-based alignment and local binary pattern-based feature extraction," *International Journal of Imaging Systems and Technology*, 2009.

[15] Apple Computer inc, "Human interface guidelines: the Apple desktop interface," *Addison-Wesley Publishing Company Reading,*, 1992.

[16] C. D. Wickens, S. E. Gordon, and Y. Liu, *An Introduction to Human Factors Engineering*, 1st ed., R. J. Dudley, Ed. Addison Wesley Longman, 1997. [Online]. Available: https://web.archive.org/web/20180619090847/http://opac.vimaru.edu.vn:80/edata/EBook/Anintruductiontohumanfactorsengineering.pdf

[17] R. Molich and J. Nielsen, "Improving a Human-Computer Dialogue," *Communications of the ACM*, 1990.

[18] S. K. Card, G. G. Robertson, and J. D. Mackinlay, "The information visualizer, an information workspace," in *Conference on Human Factors in Computing Systems - Proceedings*, 1991.

APPENDIX

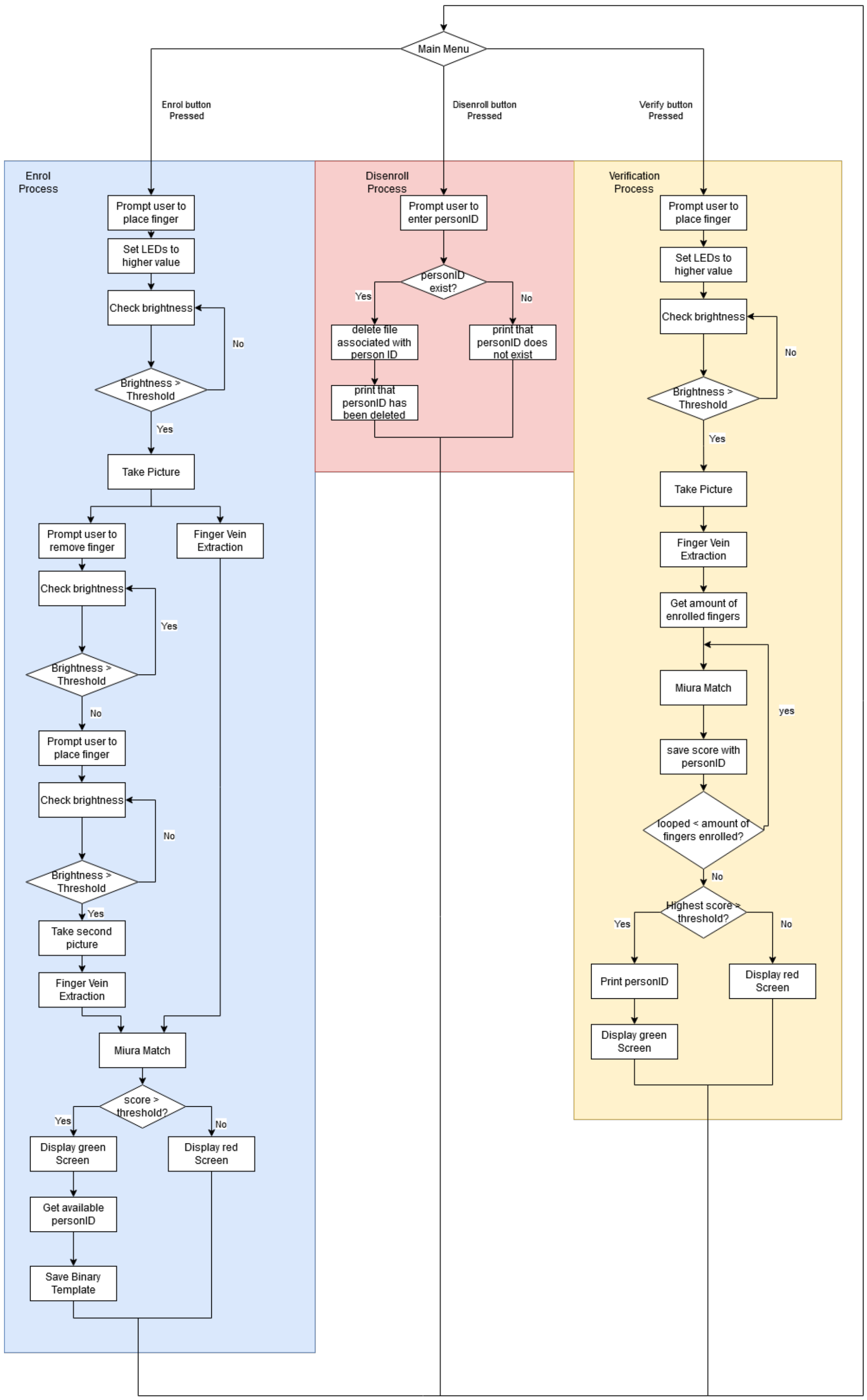


Fig. A.1: Flow diagram