Group 1: Pearl Milk Tea

Ian Chen | Carl Francis John Reyes | Imam Jurjawi
Chi-An Li | Aivje Tribiana
Luis Daniel Pambid | Yuya Tagaya

Title:

Smart CCTV Security System using Facial Recognition through Deep Learning
深度學習之智慧臉部安全監控

Abstract(摘要):

The project consists of a 720p camera connected to a Raspberry Pi model 3B located at the entrance of a certain location. It is created with a Graphical User Interface (GUI) to show the real-time recording of the camera. It has a feature that indicates when the last (wanted) person was seen in the location of the camera it captures the face, and records a video for a short period of time (9-10 seconds). At the same time it will create logs on a notepad, that indicates who is detected along with the time and date that he/she was last seen, as well as the location. Frontal Face Haar Cascade Algorithm is used for the face detection while deep learning algorithms: Histogram of Oriented Gradients (HOG) and Linear Support Vector Machine (SVM) are used to recognize the face.

此專題使用了720p攝像頭，連接樹梅派3B模型，且它使用圖形用戶界面（GUI），可即時顯示攝像機的實時記錄；我們額外附加一個功能，指示何時在攝像機位置看到最後一個（想要的）人物，它會捕捉到臉部，並在短時間內（9-10秒）錄製視頻。同時，它將在記事本上創建檔案，將檢測到的人以及他/她上次看到的時間、日期及位置。Frontal Face Haar Cascade算法用於人臉檢測，而深度學習算法：Oriented Gradients（HOG）直方圖和線性支持向量機（SVM）用於識別人臉。

Motivation and Purpose(動機和目的):

This smart CCTV project is made for the purpose of security as it detects people's faces and its location where he/she was last seen. Current commercial CCTVs can only record real-time video of certain places and hasn't been integrated with facial recognition algorithms yet. Police officers are having a hard time detecting/finding people-of-interest since they're still doing it manually. This project is also a low-cost implementation of deep learning and facial recognition algorithms through the use of a Raspberry Pi.

智慧監控器是為了安全起見，因為它可以檢測到人們的臉部及其最後一次出現的位置。目前的商業CCTV只能記錄某些地方的實時視頻，尚未與面部識別算法集成。由於警察仍在手動操作，因此很難發現、找到具危險性的人。該項目還通過使用樹梅派實現了深度學習和麵部識別算法的低成本實施。

Introduction(介紹):

In the modern society, people need security everywhere. Every building, houses, streets, and other places need a Closed Circuit TV also known as CCTV for security purposes. A Closed-Circuit Television, is a TV system in which signals are not publicly distributed but are monitored, primarily for surveillance and security purposes. [1] Facial recognition is also somehow a necessity for a CCTV, wherein the program will detect faces of different people using different algorithms available in modern technology. Facial recognition can help users to identify the person-of-interest. One innovation in today's technology is the deep learning. Deep learning, a subset of machine learning, is an artificial intelligence function that copies the human brain's workings in data processing, specifically creating patterns for decision making use. [2]

在現代社會中，安全往往是人們最注重的課題。每個建築物、房屋、街道和其他地方中，都能輕而易舉地發現監視器的存在。監視器是一種安全系統，其中信號不是公開發布的，而是受到控制的，主要用於監視和安全目的。[1]面部識別在某種程度上也是監視器的必需品，其中程序將使用現代技術中可用的不同算法來檢測不同人的面部，而其中面部識別可以幫助用戶識別感興趣的人。當今技術的一項創新是深度學習。深度學習是機器學習的一個項目，也是一種人工智能功能，可以複製人腦在數據處理中的工作。 [2]

Function or Apply Technique Description(功能及應用技術說明):

The project uses image processing, and the use of the Haar Cascade Algorithm, Histogram of Oriented Gradients, and Linear Support Vector Machine to achieve image processing on digital images. [3] The image processing that will be used focuses on facial recognition, which is a biometric software application capable of uniquely identifying or verifying a person by comparing and analyzing patterns based on the person's facial contours [4], integrated with Haar Cascade Algorithm to determine which face is present that the camera senses along with the program. Haar Cascade algorithm is a machine learning object detection algorithm used to identify objects in an image or video and based on the concept of features. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. [5] Histogram of Oriented Gradients (HOG) is a feature descriptor used to detect objects in computer vision and image processing. The HOG descriptor technique counts occurrences of gradient orientation in localized portions of an image - detection window, or region of interest (ROI). [6] Machine learning involves predicting and classifying data and to do so we employ various machine learning algorithms according to the dataset. SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes. [a] SVM is implemented in the project together with HOG to classify the person's face.

此專題使用使用Haar級圖像處理，方向梯度直方圖和線性支持向量機來實現數字圖像的圖像處理。 [3]將使用的圖像處理側重於面部識別，面部識別是一種生物識別軟件應用程序，能夠通過比較和分析基於人的面部輪廓的模式來唯一地識別或驗證人[4]，與Haar Cascade算法集成確定相機與程序一起感知哪個面部。 Haar Cascade算法是一種機器

學習對象檢測算法，用於識別圖像或視頻中的對象並基於特徵的概念。它是一種基於機器學習的方法，其中級聯功能由許多正面和負面圖像訓練。 [5] Oriented Gradients（HOG）直方圖是一種用於檢測計算機視覺和圖像處理中的對象的特徵描述符。HOG描述符技術計算圖像的局部部分中的梯度方向的出現 - 檢測窗口或感興趣區域（ROI）。 [6]機器學習涉及預測和分類數據，為此我們根據數據集採用各種機器學習算法。 SVM或支持向量機是分類和回歸問題的線性模型。它可以解決線性和非線性問題，並且可以很好地解決許多實際問題。 SVM的思想很簡單：算法創建一條線或超平面，將數據分成類。 [a] SVM與HOG一起在項目中實施，以對人的面部進行分類。

The project uses Raspberry Pi 3B, credit-card sized computer, coded using Python programming language assisted with OpenCV, PyQt4, dlib (Deep Learning Library), PIL (Python Imaging Library), NumPy, and face_recognition libraries. OpenCV (Open Source Computer Vision Library) is an open source computer vision and & machine learning software library, built to provide a common infrastructure for computer vision applications. [7] The PyQt4 is a comprehensive set of Python bindings for Digia’s Qt cross platform GUI toolkit. [8] PyQt4 supports Python v2 and v3. The GUI or Graphical User Interface used in this project is done with the use of the PyQt4 Library. The dlib is a general purpose cross-platform software library written in the programming language C++. [9] Inside the dlib library, Histogram of Oriented Gradients (HOG) together with Support Vector Machine (SVM) are used for face recognition. The Python Imaging Library (PIL) is a free library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats. [10] The library is used for labeling the images with text. The NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. [11] The library is used to support the dlib library for Image Processing. The face_recognition library is the world’s simplest face recognition library, and built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark. [12]

該項目使用樹梅派3B，信用卡大小的計算機，使用Python編程語言編寫，輔以OpenCV，PyQt4，dlib（深度學習庫），PIL（Python成像庫），NumPy和face_recognition庫。 OpenCV（開源計算機視覺庫）是一個開源計算機視覺和機器學習軟件庫，旨在為計算機視覺應用提供通用基礎設施。 [7] PyQt4是Digia的Qt跨平台GUI工具包的全套Python綁定。 [8] PyQt4支持Python v2和v3。本項目中使用的GUI或圖形用戶界面是使用PyQt4庫完成的。 dlib是用編程語言C ++編寫的通用跨平台軟件庫。 [9]在dlib庫中，Oriented Gradients直方圖（HOG）與支持向量機（SVM）一起用於人臉識別。 Python Imaging Library（PIL）是一個免費的Python編程語言庫，它增加了對打開，操作和保存許多不同圖像文件格式的支持。 [10]該庫用於用文本標記圖像。 NumPy是Python編程語言的庫，增加了對大型多維數組和矩陣的支持，以及大量的高級數學函數，可以在這些數組上運行。 [11]該庫用於支持圖像處理的dlib庫。 face_recognition庫是世界上最簡單的人臉識別庫，

使用dlib最先進的人臉識別功能構建而成。該模型在Wild標記面上的精確度為99.38％。[12]

The project begins with the training of faces in the program, wherein the person will manually input his/her name in the Trainer View. Afterwards, the program will take a photo of the person and when a face is detected it will be considered as registered face to be placed in the database.

此專題從項目中的面部訓練開始，其中該人將在訓練者視圖中手動輸入他/她的名字。之後，程序將拍攝該人的照片，並且當檢測到面部時，將其視為將被放置在數據庫中的註冊面部。

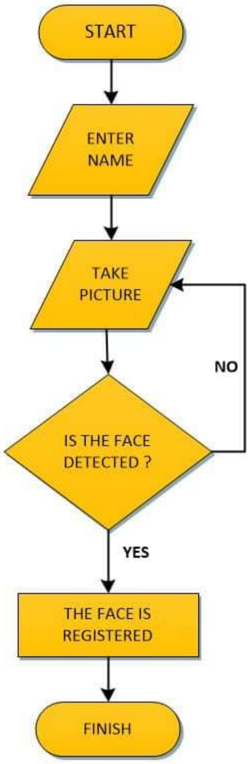


Figure 4.1 Flow Chart of Face Registration using Trainer View

After registering different faces in the database, the program will be in the CCTV View. It will run continuously until the user decides to execute the program. And when the camera detects a person, it will then determine if the said face is in the database. If the face is registered in the database, it will process the image processing algorithms. Then detects if the tolerance is greater than 0.4. If the tolerance is below 0.4, the program will repeat the image processing algorithms until it reaches a value greater than 0.4. After the program verifies that the value is above 0.4, the captured image will be saved in the program and a 10-second video will be recorded as well. Aside from the image and video acquisition, its location will also be recorded using the notepad.

在數據庫中註冊不同的面部後，該程序將在CCTV視圖中。它將持續運行，直到用戶決定執行該程序。當相機檢測到一個人時，它將確定所述面部是否在數據庫中。如果面部在數據庫中註冊，它將處理圖像處理算法。然後檢測公差是否大於0.4。如果公差低於0.4，程序將重複圖像處理算法，直到達到大於0.4的值。在程序驗證該值大於0.4之後，捕獲的圖像將保存在程序中，並且還將記錄10秒的視頻。除了圖像和視頻採集，它的位置也將使用儲存記錄。

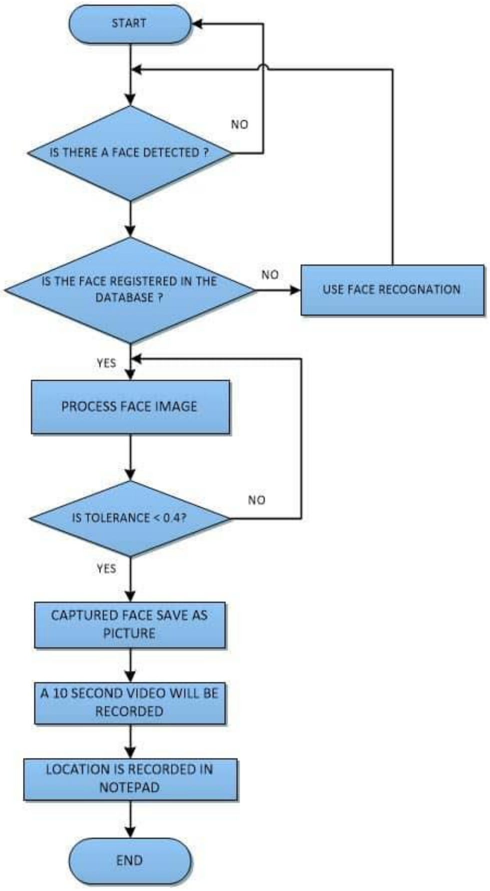
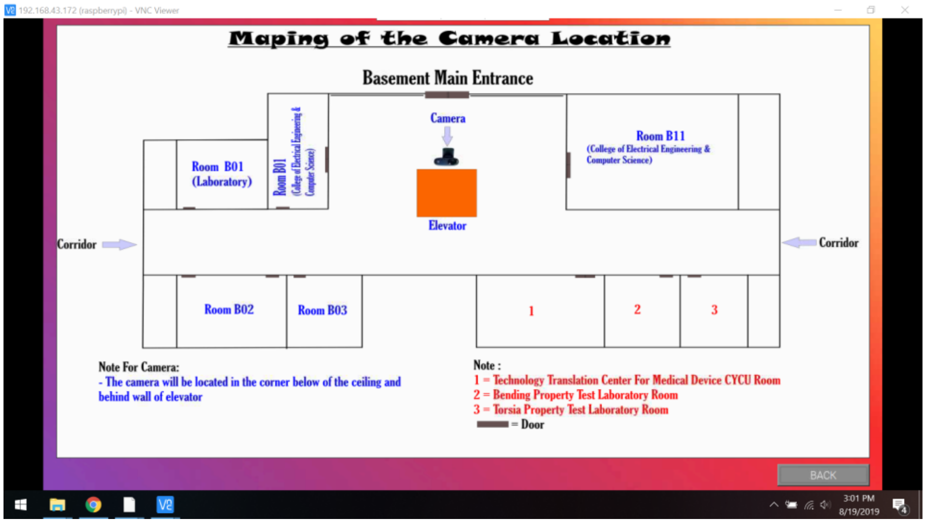

START
IS THERE A FACE DETECTED ?
NO
IS THE FACE REGISTERED IN THE DATABASE ?
NO
USE FACE RECOGNATION
YES
PROCESS FACE IMAGE
IS TOLERANCE < 0.4?
NO
YES
CAPTURED FACE SAVE AS PICTURE
A 10 SECOND VIDEO WILL BE RECORDED
LOCATION IS RECORDED IN NOTEPAD
END

Figure 4.2. Flow Chart of the CCTV View of the Program

The Graphical User Interface (GUI) of the program is built using python programming language together with the PyQt4 Library. There are three User Interfaces (UIs), and they are categorized as: CCTV View, Trainer View and Map View. The CCTV View UI does real-time image processing and face recognition as well as recording videos for 9-10 seconds, capturing faces and writing time logs. The CCTV View features are: When a face is recognized, it captures the face then saves the picture into the *captured* folder; When a face is recognized, a video is recorded for 9-10 seconds (depends on the processing time of the face recognition) then saves the video into the *recorded* folder; When a face is recognized, informations (date, time and location) are recorded into the *log* folder named <date>.txt. Key functions are applied and used as navigation of the UIs: Pressing ‘Q’ would quit the program; pressing ‘T’ would show the Train View UI; and pressing ‘M’ would show the Map View UI.

程式的圖形使用者界面（GUI）使用python編程語言和PyQt4庫構建。有三個界面（UI），它們分類為：CCTV視窗，開發者視窗和地圖視窗。 CCTV View UI可進行實時圖像處理和人臉識別，並可錄製9-10秒的視頻，捕捉面部和編寫時間。CCTV視窗功能包括：識別到臉部後，它會捕捉臉部然後將圖像保存到捕獲的文件夾中；當識別出臉部時，錄製視頻9-10秒（取決於臉部識別的處理時間），然後將視頻保存到錄製的文件夾中； 識別到面部時，信息（日期，時間和位置）將記錄到名為<date> .txt的文件夾中。 關鍵功能被應用並用作UI的導航：按'Q'將退出程序；按'T'將顯示Train View UI；並按'M'將顯示地圖視圖UI。


192.168.43.172 (raspberrypi) - VNC Viewer
Maping of the Camera Location
Basement Main Entrance
Camera
Room B11
(College of Electrical Engineering & Computer Science)
Room B01
(Laboratory)
Room B01
(College of Electrical Engineering & Computer Science)
Elevator
Corridor
Corridor
Room B02
Room B03
1
2
3
BACK
3:01 PM
8/19/2019

Note For Camera:
- The camera will be located in the corner below of the ceiling and behind wall of elevator

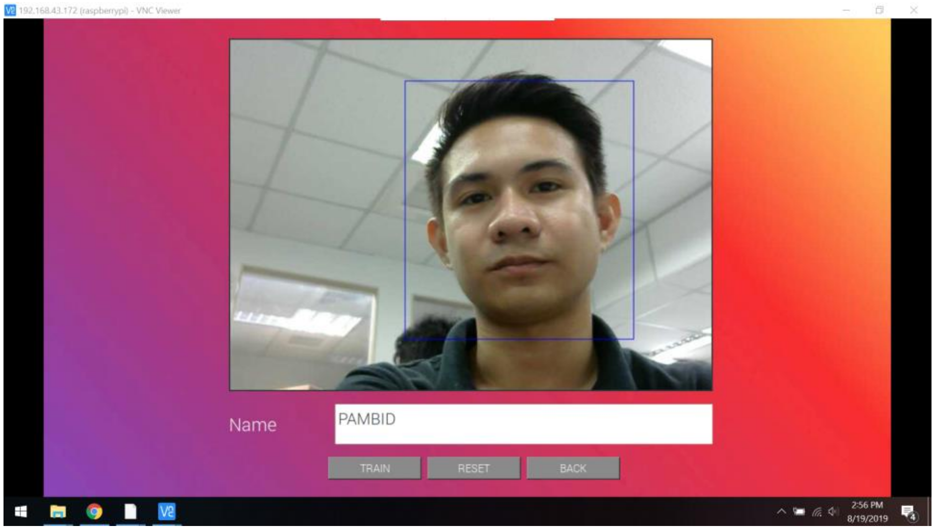
Note :
1 = Technology Translation Center For Medical Device CYCU Room
2 = Bending Property Test Laboratory Room
3 = Torsia Property Test Laboratory Room
▬ = Door

Figure 4.3. Mapping of the camera location using Map View GUI

The Trainer View UI does face image training by first detecting the face using Frontal Face Haar Cascade Algorithm. When the Train Button is clicked and a face is detected, it will take a picture then saves into the *dataset* folder, the name of the (wanted) person is then saved into the *DATABASE.txt* file. These images are then processed by image encoding, converting an image to numbers, and the encoded image is saved into the *ENCODINGS.txt* for further image processing. Also, the Train View has also a Reset Button, deletes all the data as well as the database of people's faces, saved data sets, captured faces, recorded videos and logs. The purpose of Map View is to show the whole vicinity of the location and where the camera is located. In addition, a Back Button is added into Train View and Map View so that the CCTV View UI can be navigated again.

Trainer View UI通過首先使用Frontal Face Haar級聯算法檢測面部來進行面部圖像訓練。單擊“訓練”按鈕並檢測到面部時，將拍攝照片然後保存到數據集文件夾中，然後將（想要的）人員的名稱保存到DATABASE.txt文件中。然後通過圖像編碼處理這些圖像，將圖像轉換為數字，並將編碼圖像保存到ENCODINGS.txt中以進行進一步的圖像處理。此外，列車視圖還有一個重置按鈕，刪除所有數據以及人臉，保存的數據集，捕獲的面部，

錄製的視頻和日誌的數據庫。地圖視圖的目的是顯示位置的整個附近和攝像機所在的位置。此外，在“列車視圖”和“地圖視圖”中添加了“後退”按鈕，以便可以再次導航CCTV視圖UI。



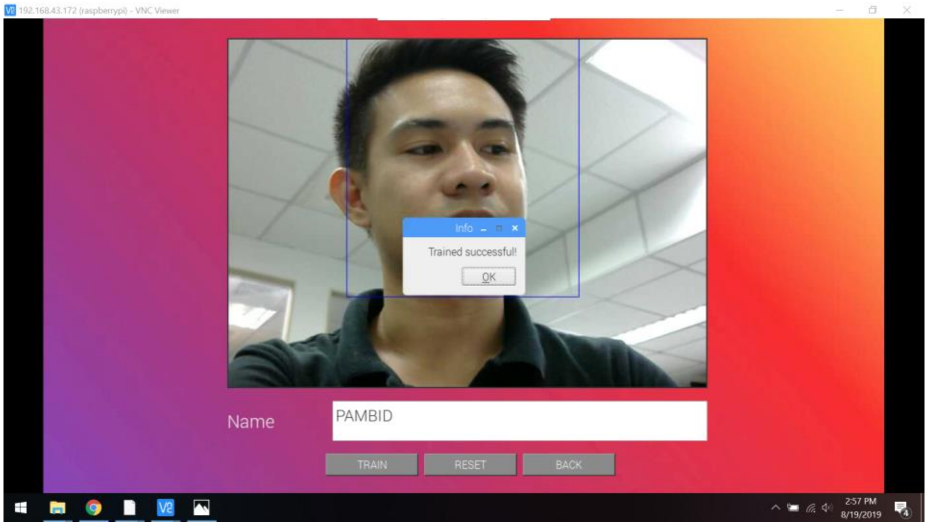
Name PAMBID

Figure 4.3. Trainer View GUI

Results(結果):

In the Trainer View part of the program, the person's face is accurately registered in the database. It will register 3 pictures of the same person during the training. The purpose of registering 3 pictures is to feed more data in the deep learning process. We limit to 3 pictures because training would take a longer time to process if there would be more. The samples of each face can be adjusted inside the program itself, and it depends on the user how many samples he/she wants.

在程序的開發者視窗部分中，人員的面部準確地在數據庫中註冊。它將在培訓期間註冊同一個人的3張照片。註冊3張圖片的目的是在深度學習過程中提供更多數據。我們限制為3張圖片，因為如果有更多的話，培訓機器需要更長的時間來處理。每個面部的樣本可以在程序內部進行調整，這取決於用戶他/她想要的樣本數量。



Trained successful!

Figure 5.1. Person is successfully registered in the Trainer View

In the CCTV View, as the registered face is detected in the camera and it reaches a tolerance of 0.4 (60% confidence), the program will record automatically a 9-10 second video showing the date and time when the person detected, as well as its recorded location in the Notepad.

在CCTV視圖中，由於在相機中檢測到註冊的面部並且其容差達到0.4（60％置信度），程序將自動記錄9-10秒的視頻，顯示檢測到的人的日期和時間。作為其在記事本中的記錄位置。

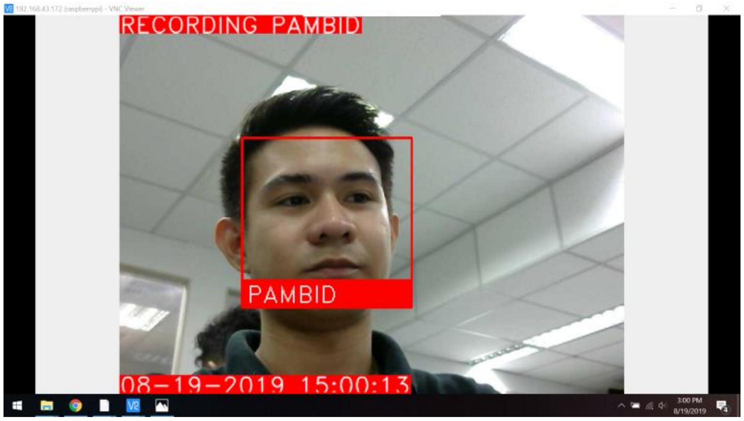
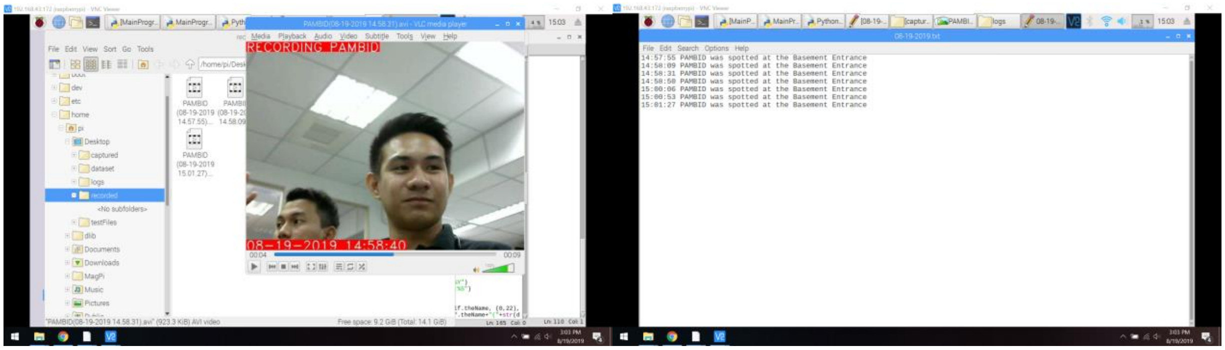


Figure 5.2. Recording of the detected person.



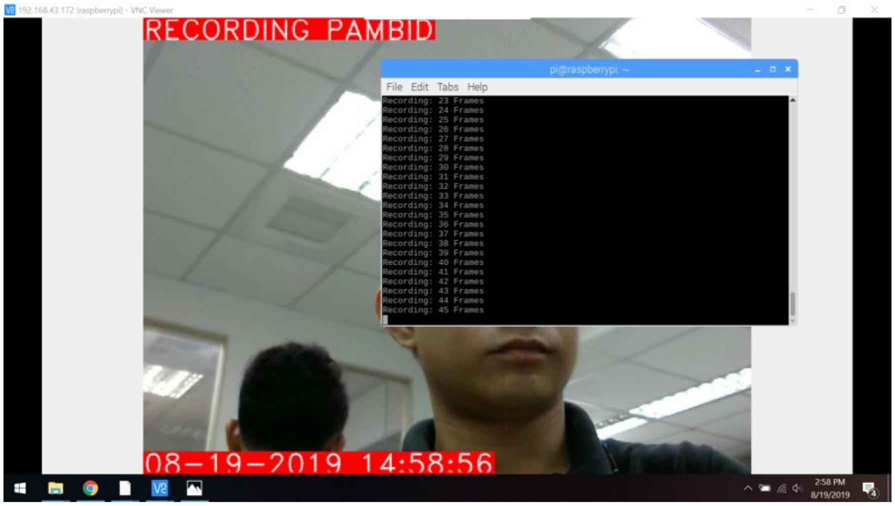
```
14:57:55 PAMBID was spotted at the Basement Entrance
14:58:09 PAMBID was spotted at the Basement Entrance
14:58:31 PAMBID was spotted at the Basement Entrance
14:58:50 PAMBID was spotted at the Basement Entrance
15:00:06 PAMBID was spotted at the Basement Entrance
15:00:53 PAMBID was spotted at the Basement Entrance
15:01:27 PAMBID was spotted at the Basement Entrance
```

Figure 5.3. Video record (left) and logs (right) with

timestamps and location of the detected person

Figure 5.4 shows the frames per second; and as shown in the figure, 100 frames are recorded. In recording the video, OpenCV is used together with its built in *cv2.VideoWriter()* function. 10 frames/second is set resulting to have a 9-10 (because of image processing time) seconds video.

圖5.4顯示了每秒，如圖所示，記錄了100幀。在錄製視頻時，OpenCV與其內置的cv2.VideoWriter（）函數一起使用。設置10幀/秒導致具有9-10（由於圖像處理時間）秒視頻。



```
Recording: 23 Frames
Recording: 24 Frames
Recording: 25 Frames
Recording: 26 Frames
Recording: 27 Frames
Recording: 28 Frames
Recording: 29 Frames
Recording: 30 Frames
Recording: 31 Frames
Recording: 32 Frames
Recording: 33 Frames
Recording: 34 Frames
Recording: 35 Frames
Recording: 36 Frames
Recording: 37 Frames
Recording: 38 Frames
Recording: 39 Frames
Recording: 40 Frames
Recording: 41 Frames
Recording: 42 Frames
Recording: 43 Frames
Recording: 44 Frames
Recording: 45 Frames
```

Figure 5.4. Printing the recording frames

Furthermore, the program will also take a picture when the registered face is detected, and it is stored into the *captured* folder. This would be the basis for identifying the person. The program uses a library called PIL (Python Imaging Library), and it is used to attach texts on the image as well as the shapes (rectangle) and colors (red).

此外，程序還將在檢測到註冊的面部時拍攝照片，並將其存儲在捕獲的文件夾中。這將是識別此人的基礎。該程序使用一個名為PIL（Python Imaging Library）的資料庫，它用於在圖像上附加文本以及形狀（矩形）和顏色（紅色）。

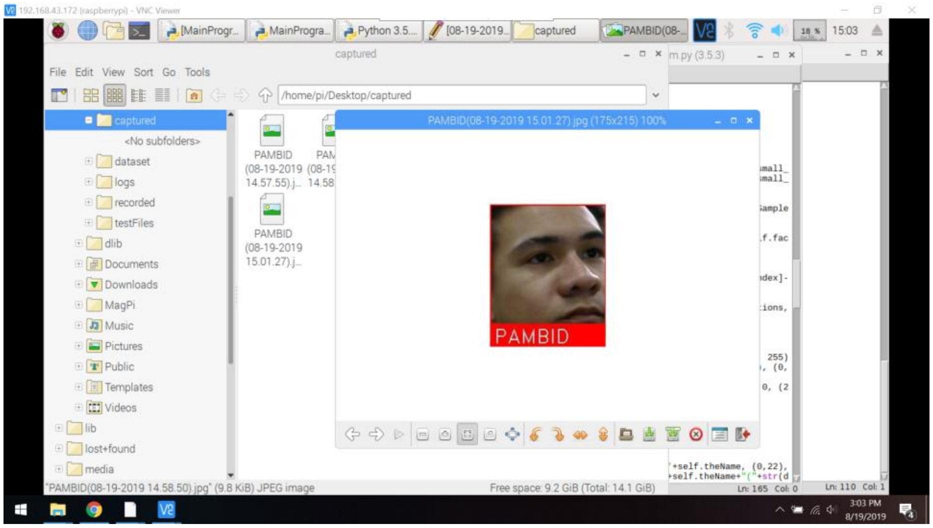

192.168.43.172 (raspberrypi) - VNC Viewer
[MainProgr...
MainProgra...
Python 3.5...
[08-19-2019...
captured
PAMBID(08-...
15:03
captured
File Edit View Sort Go Tools
/home/pi/Desktop/captured
captured
<No subfolders>
dataset
logs
recorded
testFiles
dlib
Documents
Downloads
MagPi
Music
Pictures
Public
Templates
Videos
lib
lost+found
media
PAMBID (08-19-2019 14.57.55).j...
PAMBID (08-19-2019 15.01.27).j...
PAMBID(08-19-2019 15.01.27).jpg (175x215) 100%
PAMBID
"PAMBID(08-19-2019 14.58.50).jpg" (9.8 KiB) JPEG image
Free space: 9.2 GiB (Total: 14.1 GiB)
Ln: 165 Col: 0
Ln: 110 Col: 1
3:03 PM
8/19/2019

Figure 5.5. Captured face with labeled name

As shown in Figure 5.6, we made a miniature version of the place/vicinity in which the camera is located in front of the entrance. The face of the person is not detected because she is not trained in the deep learning process. It shows that the used algorithm is almost accurate since it doesn’t result a false face recognition.

如圖5.6所示，我們製作了一個微型版本的地方附近，攝像機位於入口前面。未檢測到該人的面部，因為她沒有接受深度學習過程的培訓。它表明所使用的算法幾乎是準確的，因為它不會導致錯誤的人臉識別。

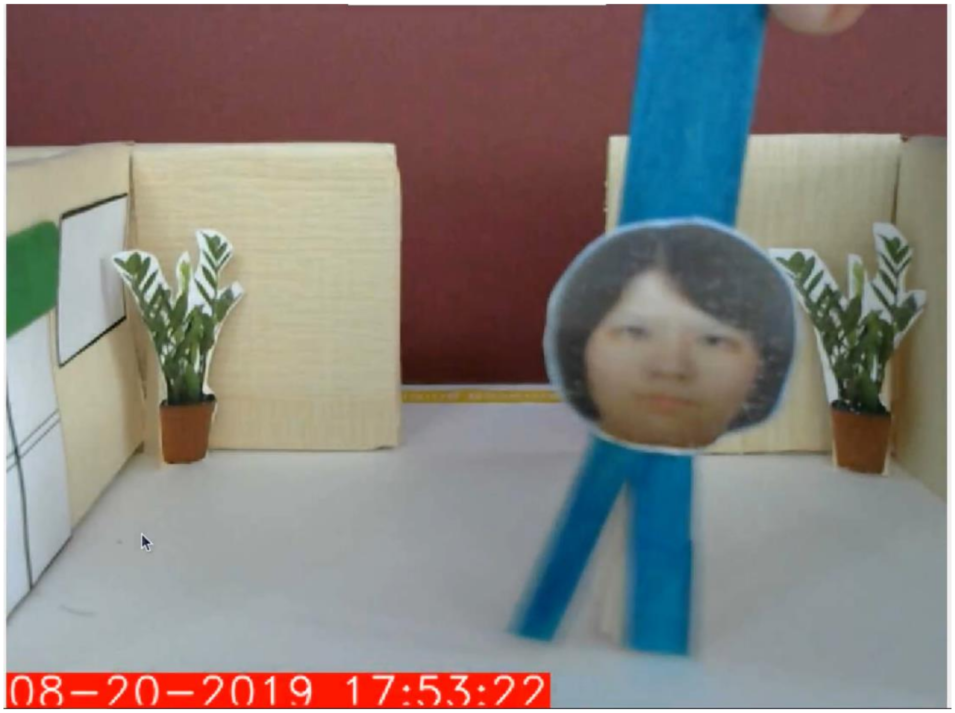


Figure 5.6. An unregistered person being undetected

In addition, it also applies for two or more persons as shown in Figure 5.7. They are not being detected because they are also not trained in the deep learning process. Same with the previous result, it doesn't give a false face recognition.

此外，它還適用於兩個或更多人，如圖5.7所示。他們沒有被發現，因為他們也沒有接受深度學習過程的培訓。與之前的結果相同，它不會給出錯誤的人臉識別。

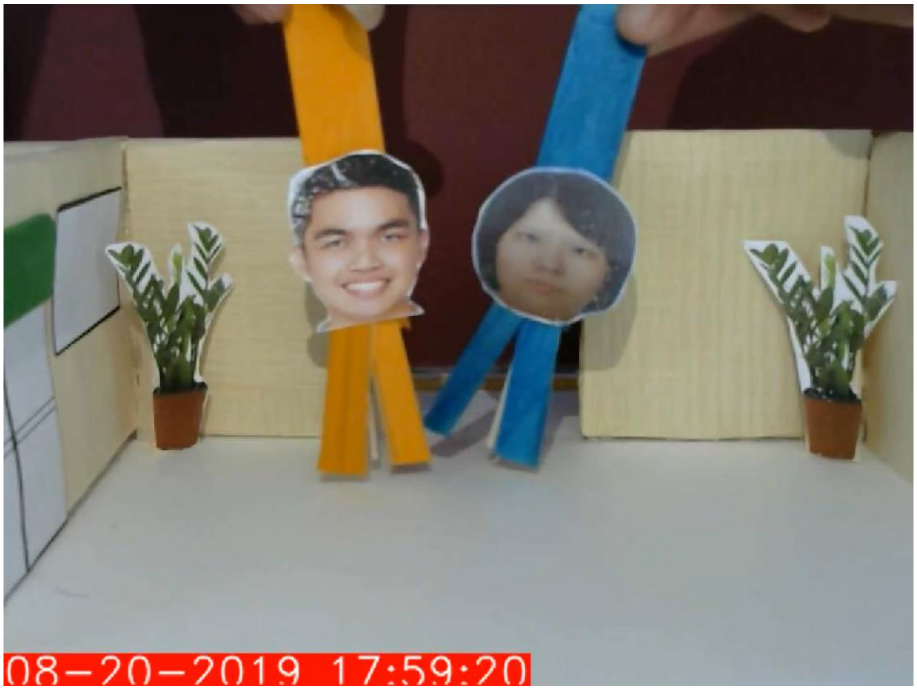


Figure 5.7. Two unregistered person being undetected

When a face of wanted person is recognized, it starts the recording feature as shown in Figure 5.8. The two faces are trained with deep learning hence they are recognized by the program. They are recognized if the trained faces, compared to the current face, reach a 0.4 tolerance value.

當識別出想要的人臉時，它會啟動錄製功能，如圖5.8所示。這兩個面孔經過深度學習訓練，因此被程序識別。 如果訓練面與當前面相比達到0.4誤差值，則識別它們。

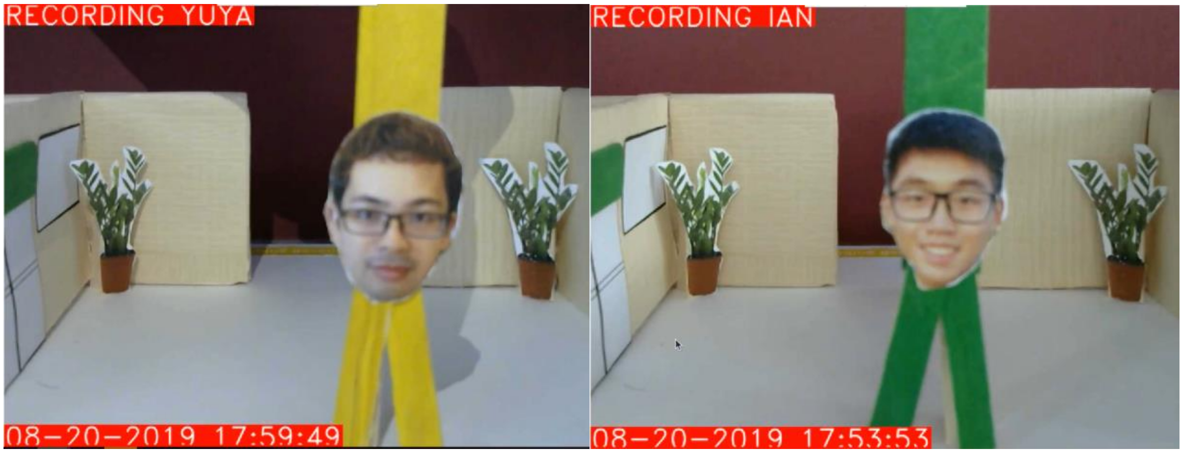


Figure 5.8 Yuya and Ian (registered) are being recorded by the CCTV

Furthermore, we also tested it in a real-life application by intentionally showing our real-life face in front of the camera to test the accuracy of the face recognition. As shown in Figure 5.9, the program is able to recognize the face even though the trained images are from the face of the miniatures.

此外，我們還在真實應用中對其進行了測試，故意在相機前面展示我們的真實面部，以測試人臉識別的準確性。如圖5.9所示，即使訓練的圖像來自微縮模型的面部，程序也能夠識別面部。

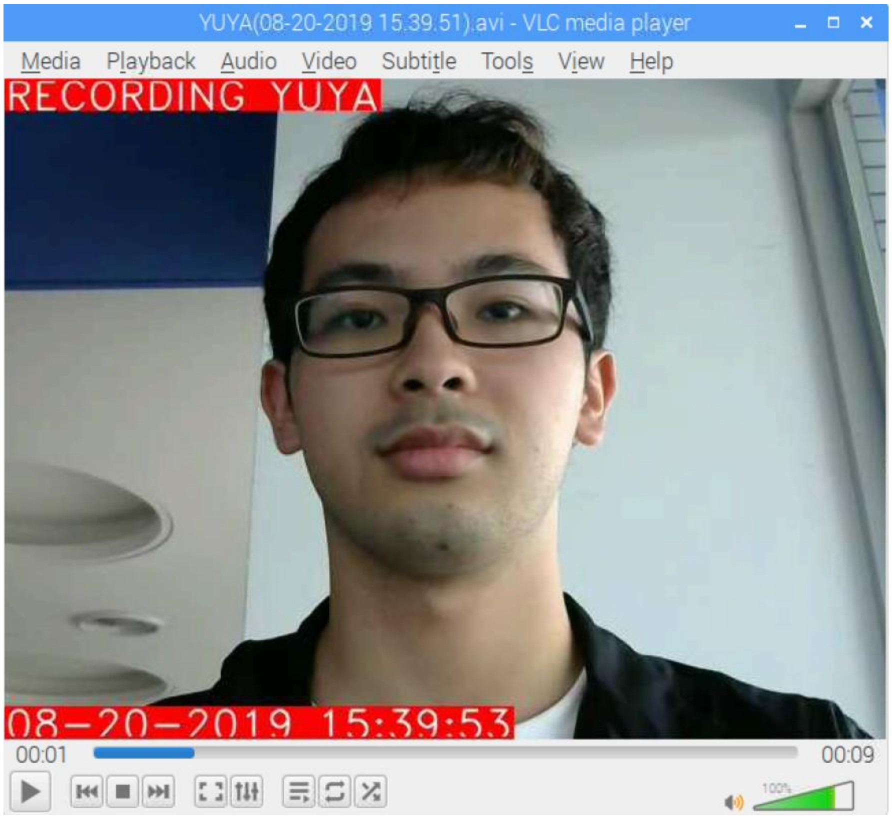


Figure 5.9. Real person detected by the CCTV

Innovation Adoption or Practical(創新採用或實踐):

With the results observed, this project can help future CCTV industries to improve their products as it is proven that with the help of the microcontroller, the CCTV can refine its capabilities especially in detecting the identities of people. Combining the facial recognition algorithms and deep learning algorithms, it can create a more accurate program that will detect people and its location depending on the camera's placement. Practical applications of this Smart CCTV project includes: detection of wanted criminals roaming around the center of a city, location of professors inside and outside the school building, attendance and location of the students in a particular class, improved household CCTV that can detect the location of the family members around the house, and many more.

根據結果，此專題可以幫助未來的CCTV行業改進他們的產品，因為事實證明，在微控制器的幫助下，CCTV可以優化其功能，特別是在檢測人員身份方面。結合面部識別算法和深度學習算法，它可以創建一個更準確的程序，根據攝像機的位置檢測人員及其位置。這個智能安全監控的實際應用包括：檢測在城市中心周圍漫遊的通緝罪犯，校內教授的位置，學生在班級的出席率和位置，改善家庭監控，可以檢測到位置房子周圍的家庭成員，以及更多。

Conclusion(結論):

Facial recognition integrated with deep learning outcomes great improvement in modern technology. The image processing algorithms present in the program were accurate enough to detect real life persons even if its trained images in the database are in the form of pictures; in this project, it’s in the form of miniature pictures. It was observed that the program can detect registered faces immediately, and can also detect the registered images even in a different background with different lighting conditions. It can be said that the Haar Cascade Algorithm and the Histogram of Oriented Gradients (HOG) Algorithm were effective algorithms that allowed the system to detect and recognize faces. Although, it was observed that with only one Raspberry Pi model B, it can only handle a single camera with its processing power, but overall it does its job and the system works smoothly.

面部識別與深度學習相結合，在現代技術方面有了很大的進步。程序中存在的圖像處理算法足夠準確，即使數據庫中訓練好的圖像是圖像形式，也可以檢測現實生活中的人。在這個項目中，它是微縮圖片的形式。觀察到程序可以立即檢測到登記的面部，並且即使在具有不同照明條件的不同背景也可以檢測登記的圖像。可以說，Haar級聯算法和方向梯度直方圖（HOG）算法是允許系統檢測和識別面部的有效算法。雖然，據觀察只有一個樹梅派B型號，它只能處理單個相機，但總體而言它完成它的工作且系統工作順利。

In sum, the study was successful and was able to function the way it was expected to. Yet, the researchers recommend others whom might continue this project, to improve the system by adding more cameras to the system. This is to ensure that the system to be more effective once applied in a real situation. Also, for them to remember that when adding another or more cameras, the processing power of the microcontroller must be improved as well. The researchers suggest to try clustering multiple Raspberry Pis, and make use of: parallel computing, multiprocessing, or multi threading. Also, by trying the latest model of the Raspberry Pi which is Raspberry Pi model 3B+, this will theoretically help the system to handle more processes at a single period of time.

總之，此專題是成功的，且能夠符合預期運作。然而，研究人員推薦可能繼續這個項目的其他人，通過在系統中添加更多攝像頭來改進系統。這是為了確保系統在真實情況下應用後更有效。此外，要讓他們記住，在添加另一台或多台攝像機時，必須提高微控制器的處理能力。研究人員建議嘗試聚集多個樹梅派，並利用-並行計算，多處理或多線程。此外，通過嘗試Raspberry Pi型號3B +的Raspberry Pi的最新型號，理論上這將有助於系統在一個時間段內處理更多過程。

References(資料來源):

[1] CCTV (closed circuit television). Retrieved from https://whatis.techtarget.com/definition/CCTV-closed-circuit-television

[2] Deep Learning. Retrieved from https://www.investopedia.com/terms/d/deep-learning.asp

[3] Digital Signal Processing. Retrieved from https://en.wikipedia.org/wiki/Digital_image_processing

[4] Facial Recognition. Retrieved from https://www.techopedia.com/definition/32071/facial-recognition

[5] Deep Learning Haar Cascade Explained. Retrieved from http://www.willberger.org/cascade-haar-explained/

[6] Histogram of Oriented Gradients (HOG) Descriptor. Retrieved from https://software.intel.com/en-us/ipp-dev-reference-histogram-of-oriented-gradients-hog-descriptor

[a] Support Vector Machines(SVM). Retrieved from https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989.

[7] About OpenCv. Retrieved from https://opencv.org/about/

[8] PyQt4 Project Description. Retrieved from https://pypi.org/project/PyQt4/

[9] dlib C++ Library. Retrieved from http://dlib.net/

[10] PIL documentation. Retrieved from https://pillow.readthedocs.io/en/stable/

[11] Numpy. Retrieved from https://www.numpy.org/

[12] The world's simplest facial recognition api for Python and the command line. Retrieved from https://github.com/ageitgey/face_recognition