

# 基於人工智慧的轉移學習之人臉辨識以建立學生點名系統

## Face Recognition Based on Artificial Intelligence Transfer Learning to Establish a Student Roll Call System

沈慧宇<sup>1</sup> 陳永欽<sup>2</sup> 張晉銜<sup>3</sup>

Hui Yu Shen<sup>1</sup>, Yeong-Chin Chen<sup>2</sup> JHANG, JIN SIAN<sup>3</sup>

<sup>1</sup> 建國科技大學 資訊與網路通訊系 副教授

<sup>1</sup> Chienkuo Technology University Department of Information and Network Communication Associate Professor

E-mail : [wyshe@ctu.edu.tw](mailto:wyshe@ctu.edu.tw)

<sup>2</sup> 亞洲大學 資訊工程學系研究所 教授

<sup>2</sup> Asia University Graduate School of Computer Science and Information Engineer Professor

E-mail : [ycchenster@gmail.com](mailto:ycchenster@gmail.com)

<sup>3</sup> 亞洲大學 資訊工程學系研究所 研究生

<sup>3</sup> Asia University Graduate School of Computer Science and Information Engineer Student

E-mail : [107121007@gm.asia.edu.tw](mailto:107121007@gm.asia.edu.tw)

### 摘要

本研究建置一個 AIoT(AI & IoT)系統平台，並運用 IoT 技術將樹莓派(Raspberry Pi)結合 RFID、LCD 與人臉辨識技術的方式收集學生證資料與學生人臉辨識結果，其中 MQTT 負責傳送學生證資料與學生人臉辨識結果至後端之雲端大數據(Big Data)伺服器。後端所使用大數據平台技術包括利用 Apache Kafka 進行資料暫存，並運用發佈/訂閱功能將學生人臉照片發佈至 Apache Hadoop HDFS 儲存，同時學生證資料與辨識結果則發佈至 MongoDB 儲存，若辨識結果低於預期則觸發轉移學習(Transfer Learning)以重新訓練(Retraining)人工智慧模型。

透過以上技術所設計的學生人臉辨識模型訓練機制，本篇論文依據學生人臉辨識結果判斷三種不同的資料處理流程。首先，若辨識結果高於重新訓練的預設值時，則只需進行儲存相關點名記錄；其次，若辨識結果介於預設值與最低下限值時，則後端系統需將人工智慧模型進行重新訓練並將訓練好的模型更新到前端樹莓派；最後，若辨識結果低於最低下限值，則後端系統將發送一則 Email 至教室管理人員信箱，以預警通告 RFID 數據可能與人臉辨識結果不一致，並以此方式達到辨識自動化之效果。

**關鍵字：**Apache Hadoop、Apache Kafka、Apache NiFi、MongoDB、轉移學習

### Abstract

In this research, an AIoT (AI & IoT) system platform is established. By using IoT (Internet of Things) technology, we combined the Raspberry Pi with RFID, LCD and face recognition technology together to collect student ID data and student identification results. Additionally, we used MQTT to send student ID data and student face recognition results to the back-end Big Data server. The big data platform

technology used in the back-end includes Apache Kafka for the use of publish / subscribe functions for student face photos in Apache Hadoop HDFS storage. At the same time, the student ID data and identification results are published to MongoDB storage, if the recognition result is lower than expected, then transfer learning is triggered to retrain the artificial intelligence model.

By way of the retraining mechanism of the student's face recognition model designed by the above technology, three different data processing processes based on the results of student's face recognition are implemented. First, if the recognition result is higher than the preset value of retraining, only the relevant roll call records are stored; Second, retrain and update the trained model to the front-end Raspberry Pi if the recognition result is located between retraining preset value and recognition low bound; Finally, if the recognition result is lower than the recognition low bound, the back-end system will send an email to the classroom management personnel to warn and notify that the RFID data and the recognition results are inconsistent. In such a way, the automation recognition is achieved.

**Keywords :** Apache Hadoop 、 Apache Kafka 、 Apache NiFi 、 Transfer Learning 、 MongoDB

## 壹、前言

隨著物聯網 (Internet of Things, IoT) 與人工智慧 (Artificial Intelligence, AI) 的技術成熟, 加上大數據 (Big Data) 及機器/深度學習的不斷發展, 所涉及的應用領域也隨之增加。目前有許多校園將人工智慧的人臉辨識系統導入圖書館的門禁系統或是自動借書系統, 此方式不僅能讓忘記帶學生證的同學也能進入圖書館進行借閱書籍。使用此系統前需向圖書館人員進行申請, 圖書館人員會將學員照片進行人臉辨識模型訓練。剛開始使用人臉辨識門禁系統時, 辨識學員人臉的準確率都能維持很高, 然而時間一久, 準確率也隨之下降, 此時圖書館人員必須重新訓練人臉辨識模型以提升準確率。在學員數量不多時, 管理人員能輕鬆以手動方式進行, 然而圖書館是許多學員收集資料以及提取知識的地方, 若每次都以此方式進行, 人力和時間將是一大問題。

物聯網在校園之中, 除了剛所提到的門禁系統應用外, 還應用於考勤系統等。目前這些設備所產生的數據, 以傳統資料庫進行儲存或處理, 則沒甚麼問題。但未來, 將有更多物聯網設備導入校園, 數據裡可能包含結構化和非結構化資料, 若以傳統資料庫進行處理, 勢必只能購買硬體設備或是記憶體以提升處理速度, 將來所花費的成本將難以估計。

本篇建置一個 AIoT 系統平台, 運用物聯網技術, 將 Raspberry Pi 結合 RFID 與人臉辨識系統, 以及後端結合大數據 (Big Data) 平台進行整合。目的是節省人力與時間, 進而達到自動化之效果。同時該系統具備可擴展性、穩定性與可靠性等

特性。本篇將系統應用於學生點名系統。

在第貳章節介紹文獻探討，第參章介紹硬體、軟體以及 AIOT 系統架構設計，第肆章展示系統實驗結果，第五章討論整體系統架構。

## 貳、文獻探討

本篇論文所採用的是[1]所提出的 OpenFace 深度學習人部識別模型，它基於[9]提出的 FaceNet 深度卷積神經網絡。在 Labeled Faces in the Wild (LFW)數據集上，FaceNet 提供了 99.63%的新記錄準確性，並在 YouTube Faces DB [9]上提供了 95.12%的新記錄。本篇論文透過 FaceNet 提取 128 個維度的人臉特徵。特徵提取後，將支持向量機(SVM)[2]進行分類。[5]該篇提到用於人臉識別時，支持向量機 (SVM) 是一種比多層感知器 (MLP) 更好的學習演算法。

[4]使用 Apache NiFi 設計資料流流程、Apache Kafka 負責管理與發送資料流、MongoDB 儲存資料、使用 Apache Storm 對數據進行處理，以此建置出一個能即時處理資料流的物聯網系統平台，能快速且完整的收集到數據。它可以部署在可擴展的設備網絡中，並用於處理數百萬個節點。[7] 提出一個可擴展的與容錯的資料流管理框架，該框架可以用於結構化和非結構化數據，將 Kafka 與 HDFS 進行整合並由 Apache NiFi 執行資料流系統。

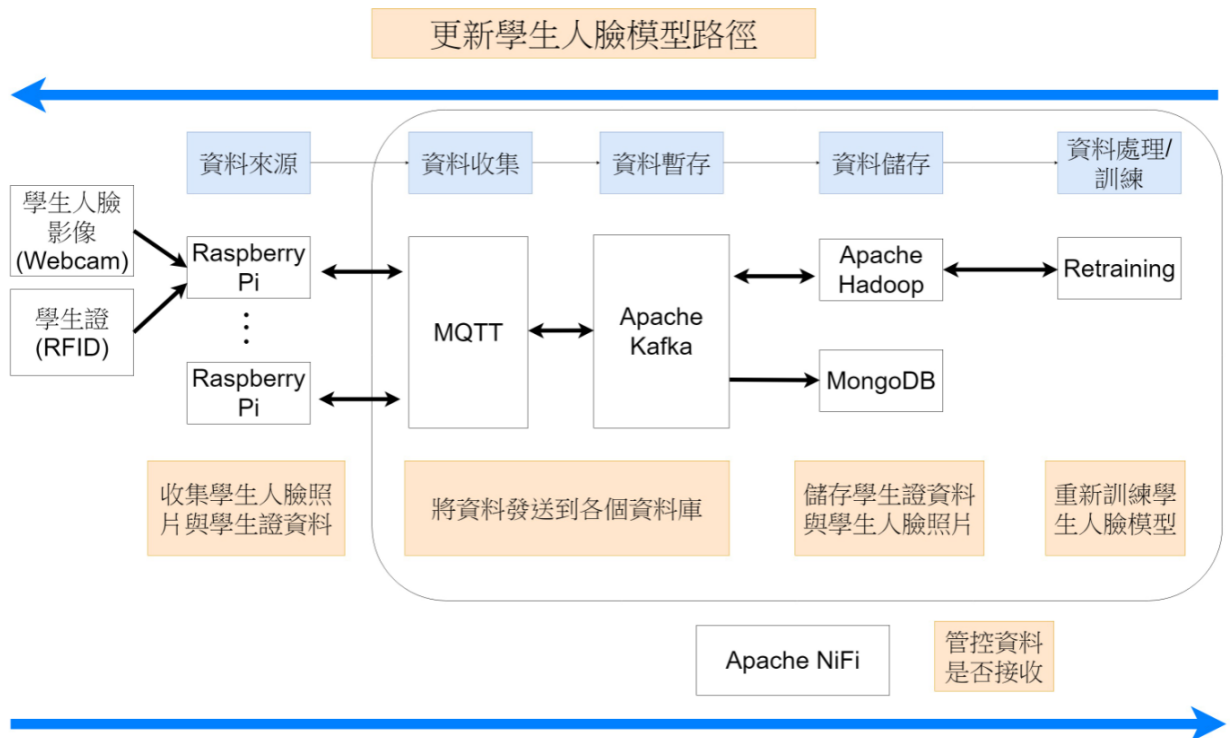
[8]將 Raspberry Pi 以及集群的方式配置在智慧工廠中，並評估 Raspberry-Pi 的群集的可用性、經濟實惠的特性與性能，而[6]將 Raspberry Pi 中加入人臉辨識技術，利用 Haar Cascades 進行人臉偵測與特徵臉進行人臉辨識。[3]提出一個基於 MQTT 通訊協議的邊緣裝置來實現數據收集的 AIoT 架構，設計出一個能有效減少數據縮減的方法，能夠減少來自物聯網數據收集的一些冗餘數據並減少存儲所需的開銷。

本篇論文建構 AIoT 的系統平台，它提供了可靠性、可擴充性及穩定性之重要特點，並經由人臉辨識技術做為應用，以驗證人工智慧結合物聯網的運算機制，並藉以瞭解整體資料處理流程與設計原理。

## 參、研究實驗與設計

### 一、系統模組設計

本研究運用 Raspberry Pi 執行 Python 程式，運用 Webcam 收集學生人臉照片後進行人臉辨識，RFID 收集學生證資料，MQTT 將學生證資料、學生人臉照片以及學生人臉辨識結果發送至後端資料庫。後端部分運用 Apache Kafka 發送資料到各個系統，Apache Hadoop 負責儲存學生照片，MongoDB 負責儲存學生證資料與學生人臉辨識結果，Retraining 主機負責重新訓練學生人臉模型，架構如圖 1 所示。



## 重新訓練學生人臉模型路徑

圖 1.系統架構設計

### (一)硬體與各子系統介紹

#### 1. Raspberry Pi

圖 2 為 Raspberry Pi 4 Model B，是源自一個開放源程式碼的硬體專案平台，是一款基於 Linux 系統的單板機電腦，只需一張 SD 卡即可運行於 Linux 作業系統。其應用非常廣泛，例如文字編輯器、遊戲以及能播放高畫質影片等等。同時具備簡單 I/O 功能的電路板。Raspberry Pi 還可以用來開發交互產品，例如溫溼度感測訊號、控制電燈等等各式各樣設備。我們在 Raspberry Pi 上執行 Python 程式，程式裡包含辨識學生證的卡號、收集學生的人臉照片以及辨識學生的人臉。



圖 2. 樹莓派 (Raspberry Pi 4 Model B)

### (二)RFID 進行學生證資料收集之介紹

圖 3 為 RFID (Radio Frequency Identification) 或稱為無線射頻辨識，是一個無線通訊技術且非接觸式的自動辨識系統。主要由利用射頻訊號以及無線方式傳送

及接收數據資料。RFID 系統組成元件包含電子標籤(Tag)、讀取器(Reader)與天線(Antenna)[10]，讀取器可讀取電子標籤內存的資訊並將資料送給資料庫進行後續應用。學生證本身為 RFID 卡，本研究運用學生證來進行刷卡動作，抓取學生證的資料與學生人臉辨識結果進行比對，比對後將在 LCD(圖 4)上顯示能否通過。



IC 感應卡



圖 4. LCD 液晶螢幕顯示器

圖 3. RFID-RC522 IC 卡感應模組

### (三)學生臉部辨識之介紹

人臉辨識技術是利用攝像頭收集人臉圖像或是影片，並自動在圖像或是影片中檢測及跟蹤人臉位置，之後將偵測到的人臉進行比對。一般在對圖形或影像辨識前需要做三個步驟，第一人臉偵測(face detection)、第二是特徵提取(feature extraction)以及第三人臉辨識(face recognition)。

#### 1. 人臉偵測

人臉偵測負責尋找人臉位置或是找出人臉的細微特徵，例如眼睛、嘴巴、眉毛等位置，並忽略不需要的地方，如車子、動物等。通常會使用 Haar 特徵與 Adaboost 級聯分類器[11]對圖像進行每一區塊的分類，如通過級聯分類器，則筆試該位置有人臉影像。

#### 2. 特徵提取

特徵提取處理圖像中不同角度，將它轉到同一個基準上，之後將人臉轉換為一串數字。人臉特徵有分為幾類，例如基於幾何特徵法和模板匹配法，是指眼睛、鼻子和嘴巴等臉部特徵之間的幾何關係，如距離、面積或角度等。

#### 3. 人臉辨識

將提取的人臉特徵與資料庫的人臉進行比對，根據相似度進行分類，常用的分類器有支持向量機。在此運用基於深度神經網絡的開源人臉識別系統 OpenFace 進行學生人臉辨識。

### (四)MQTT 發送學生證資料與學生辨識結果之介紹

MQTT (Message Queuing Telemetry Transport) 是一個訊息佇列遙測傳輸，透過發布/訂閱的方式進行資料傳送，能將訊息有效的發送給一個或多個伺服器，它建構於 TCP/IP 之上。MQTT 開發目的為了針對物聯網的特性，好處是體積小、低功耗、數據包較少以及頻寬低或低耗能的條件下傳送資料，這些特點使它適用範圍非常廣泛，目前智慧家庭以及小型化設備都已廣泛使用。我們運用 MQTT 將學生證的卡號與學生人臉辨識的結果發送到後端資料庫。

### (五)Apache NiFi 監控後端學生資料之流向介紹

Apache NiFi 是一種開放式的資料流系統，主要用於處理和分發資料。NiFi

提供了圖形介面方式(如圖 5 所示)，用於創建、監控以及將各系統之間的資料流進行串連，同時具備高度可配置與可修改的資料流過程。透過自動化和可視化管理系統之間的資料流，並允許及時跟蹤資料流。另外也可對資料進行擷取(extract)、轉換(transform)和載入(load)後送至目的端。不僅能在一般伺服器上架設，也可配置於有網路的邊緣設備(例如:Raspberry Pi 等)來進行資料收集。我們運用此技術來查看資料是否有無傳送到資料庫。

### (六)Apache Kafka 發佈學生證資料與學生人臉辨識結果之介紹

是一個開源的分散式串流平台(distributed streaming platform)。它提供了一個分散式消息訊息佇列(Publish/Subscribe)系統[12]，主要特點是能夠將訊息持久化的功能，在 TB 級以上資料也能確保常數時間複雜度的存取效能。高吞吐量，能在低接的電腦上提供每秒十萬筆以上的訊息傳輸。同時支援離線資料處理與即時資料處理，另外以不中斷的方式進行水平擴充。我們運用發佈/訂閱的方式將學生證卡號、辨識結果與圖像等發送給不同的資料庫。

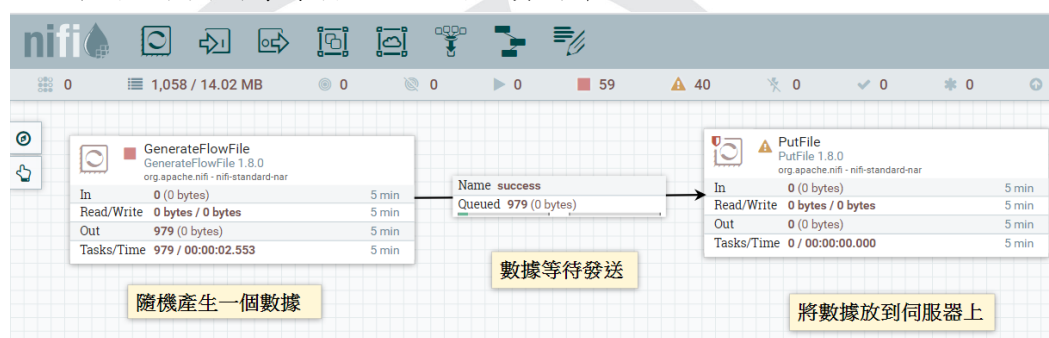


圖 5. Apache NiFi 圖控流程

### (七)Apache Hadoop HDFS 儲存學生照片之介紹

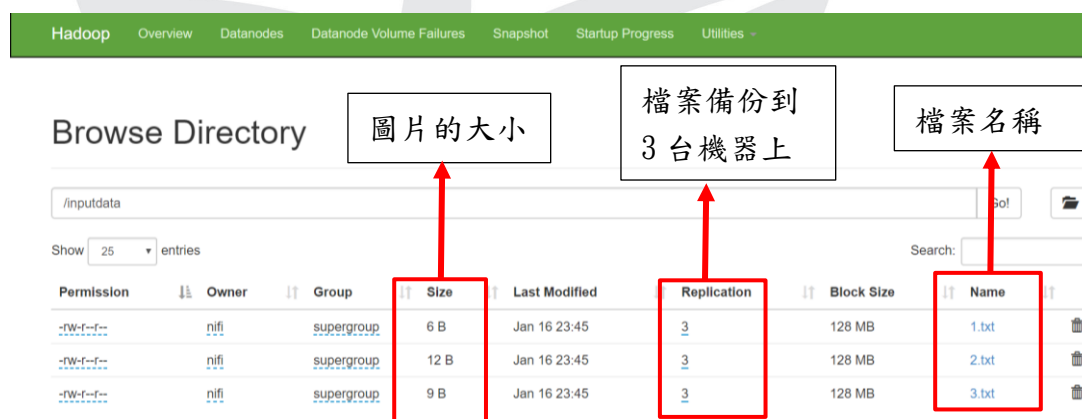


圖 6. Apache Hadoop 圖控介面

HDFS 是一個分散式檔案儲存系統，目的是用於儲存超大資料集。它能佈署於廉價的設備上並透過叢集的方式能將單一伺服器擴展到多台機器上[13]，同時具備高容錯率、可擴展性、高吞吐量的數據訪問等特性，這使得在設備出現故障時也可確保檔案不會丟失。在這裡我們將使用 Apache Hadoop HDFS 來儲存人臉圖像，如圖 6 所示。我們運用它儲存學生照片。

## (八)MongoDB 儲存學生證資料與學生人臉辨識結果之介紹

MongoDB 是一種文件導向的分散式資料庫管理系統[14]，以文本的方式進行儲存。MongoDB 的資料結構以 Key 跟 Value 組成的，其數據是用二進制的 JSON 格式 BSON 存儲的，它沒有固定的資料表，所以可以讓數據的存儲以及數據結構更靈活，存儲速度更加快。適合儲存非結構化資料(JSON)與半結構化資料(XML)。我們用來儲存學生證卡號與學生人臉辨識的紀錄檔。

## 二、AIoT 系統架構

### (一)前端感測、偵測與辨識系統介紹

透過 Raspberry Pi 執行 Python 程式，裡面功能有以下五個流程。第一是偵測 RFID 有無進行刷卡動作。第二當有 RFID 刷卡時運用 Webcam 拍攝學生人臉照片，之後將學生照片與學生人臉模型進行比對以及辨識。第三將 RFID 的資料與辨識結果進行比對，第四將比對的結果顯示在 LCD 上，並顯示通過或是不通過的訊息。第五透過 MQTT 將 RFID 資料、辨識結果與圖片上傳到後端 Apache NiFi，流程如圖 7 所示。

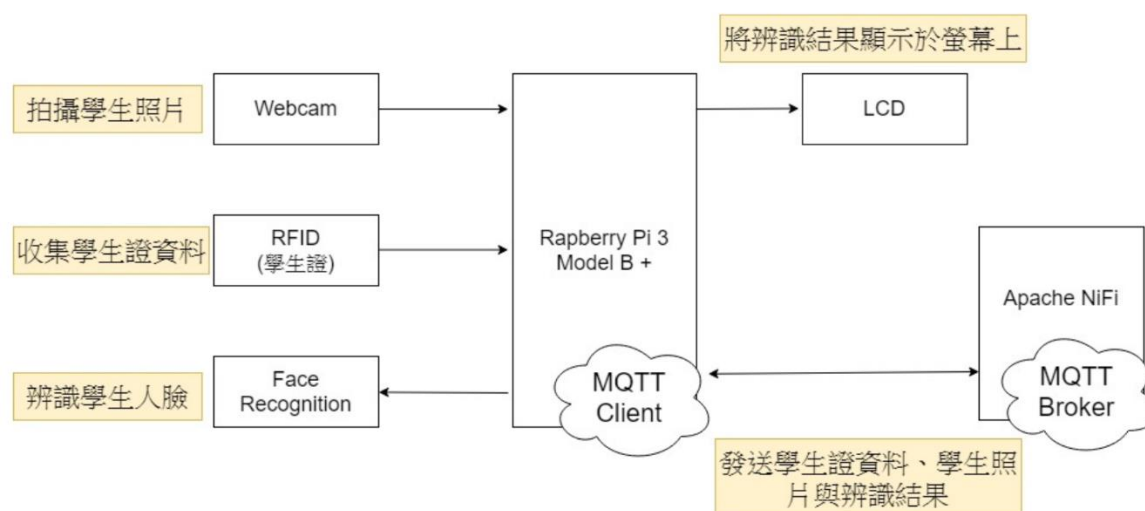


圖 7. 前端系統架構

### (二)後端儲存與再訓練系統介紹

運用 MQTT 接收資料，之後將資料送給 Apache Kafka，運用 Apache Kafka 發佈/訂閱的方式將學生照片送至 Apache Hadoop 進行儲存、將 RFID 資料與辨識結果儲存至 MongoDB。如有需要將模型進行重新，則把圖片送至 Retraining 的主機，運用人工智慧的轉移學習將學生人臉模型進行重新訓練。透過 Apache NiFi 將以上整個資料流串接起來，並運用可視化的方式進行即時監控，流程如圖 8 所示。

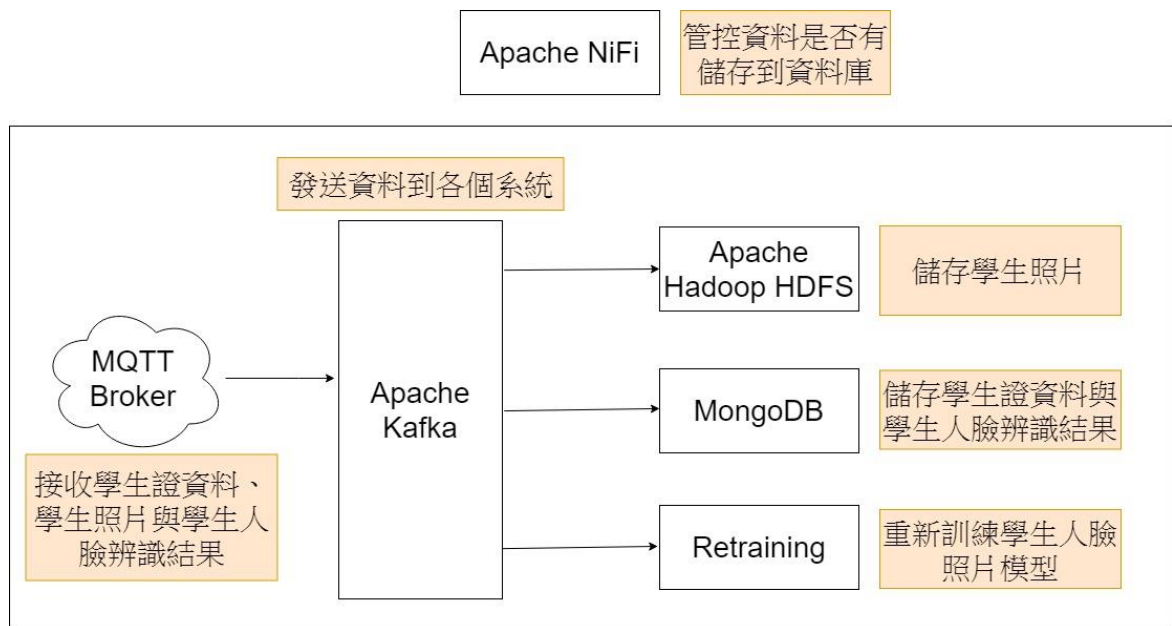


圖 8. 後端系統架構

### (三)資料處理流程介紹

前端 Raspberry Pi 執行 Python 程式進行學生人臉辨識，依據學生人臉辨識結果，將資料發送至三種不同的資料處理流程。第一，辨識結果高於重新訓練的預設值；第二，辨識結果介於預設值與最低下限值；第三，辨識結果低於最低下限值。

#### 1. 辨識結果高於重新訓練的預設值

MQTT 接收資料後送至 Apache Kafka，運用 Apache Kafka 的發佈/訂閱的機制，將圖片送至 Apache Hadoop 進行儲存，另外將準確率與 RFID 資料送至 MongoDB 儲存，如圖 9 所示。

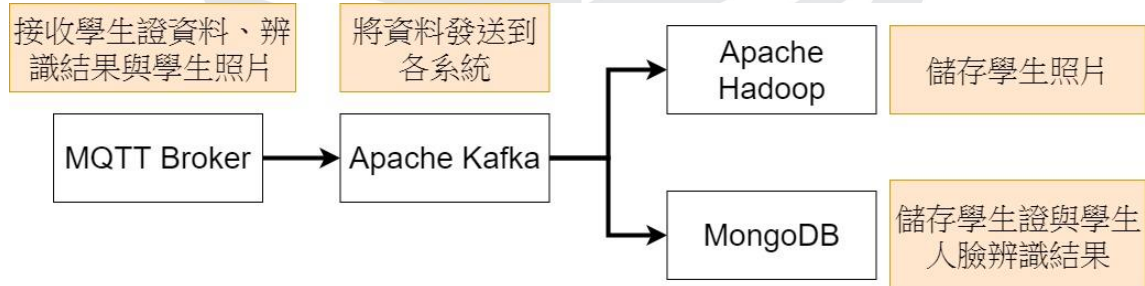


圖 9. 學生人臉辨識準確率高於預設值之資料處理流程

#### 2. 辨識結果介於預設值與最低下限值

MQTT 接收資料後送至 Apache Kafka，將圖片送至 Apache Hadoop 進行儲存，將準確率與 RFID 資料送至 MongoDB 儲存，另外圖片需送至訓練模型的主機。運用轉移學習方式進行人臉模型重新訓練，如圖 10 所示。另外將訓練好的模型以原本的資料流送回邊緣裝置進行模型更新，另外一條流向 Apache Hadoop 儲存訓練好的模型，如圖 11 所示。

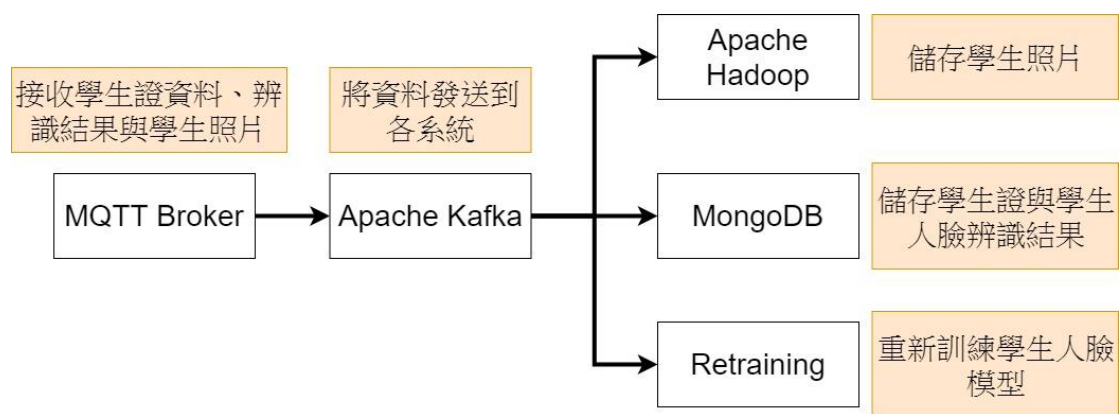


圖 10. 學生人臉辨識準確率介於預設值之間之資料處理流程

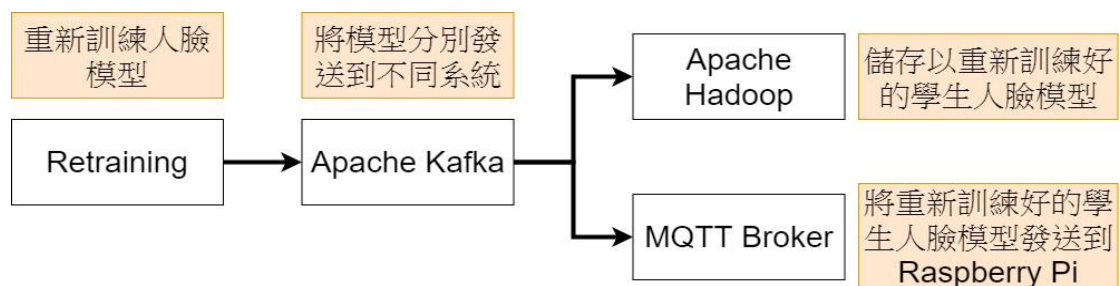


圖 11. 更新學生人臉辨識模型到 Raspberry Pi

### 3. 辨識結果低於最低下限值

MQTT 接收資料後送至 Apache Kafka，將圖片送至 Apache Hadoop 進行儲存，將準確率與 RFID 資料送至 MongoDB 儲存，另外新增一條寄送 Email 的資料流，如圖 12 所示。

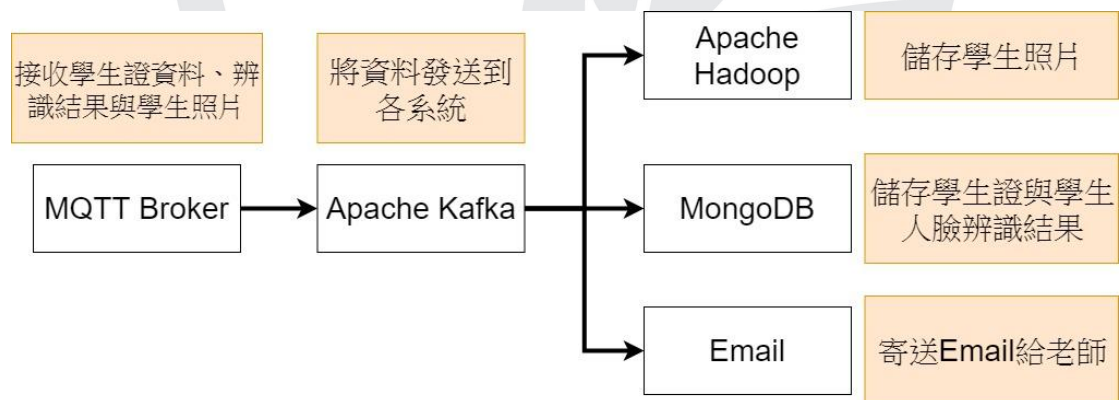


圖 12. 學生人臉辨識準確率小於預設值的資料處理流程

## 三、AIoT 概念介紹

是由物聯網(IoT)與人工智慧(AI)兩項技術所結合，透過物聯網產生數據或影像、大數據平台進行儲存，最後將收集好的運用人工智慧來進行決策。以生產線判斷產品的良率為例，可以透過物聯網收集影像並由大數據平台儲存，最後經由人工智慧進行決定產品是某汰換。經由這樣的形式可以達到自動化的效果。

## 肆、實驗結果

### 一、辨識結果高於重新訓練的預設值

當學生證與辨識人臉的結果一致時，如圖 13 所示，辨識出的結果為 91.28%，則讓此人進入且模型無需重新訓練並將學生證資料、學生人臉辨識結果與學生資料發送給後端。之後在 LCD 上顯示此位同學姓名，如圖 14 所示。後端 Apache Kafka 透過發佈/訂閱的方式(圖 15)將學生證資料送給 MongoDB 儲存(圖 16)，將學生照片儲存在 Apache Hadoop HDFS (圖 17)。

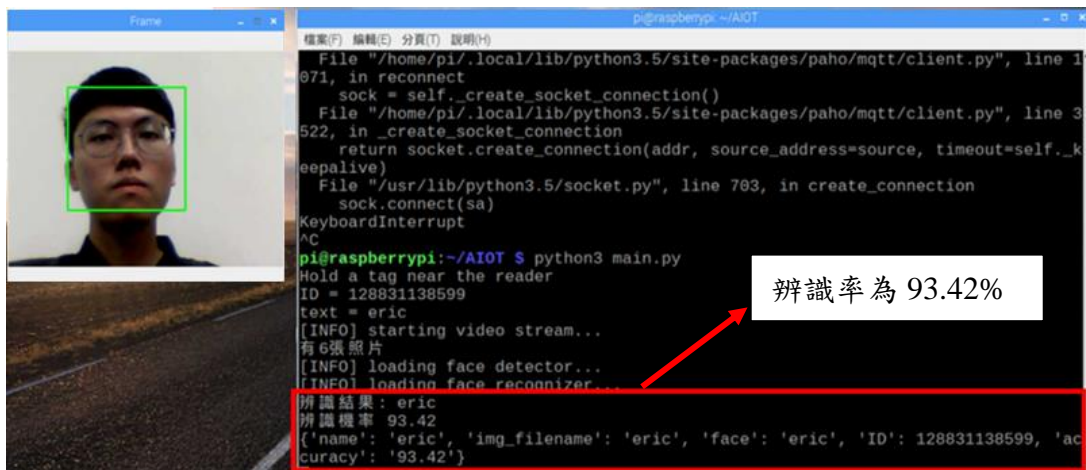


圖 13. 學生人臉辨識結果

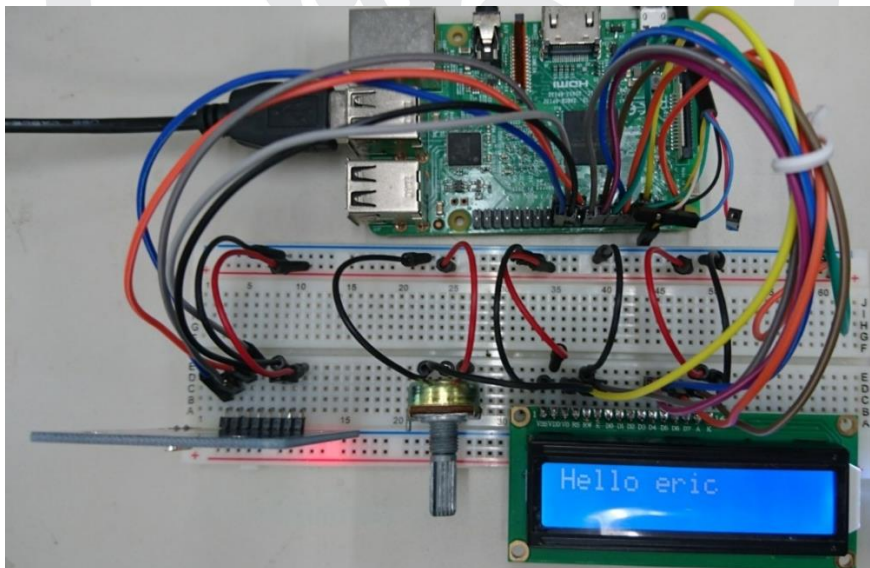


圖 14. 學生人臉辨識後之結果顯示於 LCD

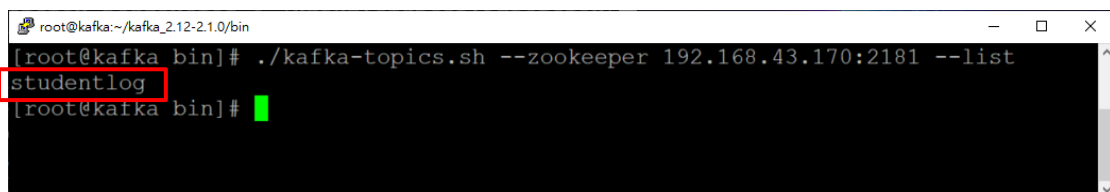


圖 15. 學生資料暫存於 Kafka 之畫面

```
hadoop@mongodb:~$ > show dbs
admin          0.000GB
config        0.000GB
local         0.000GB
studentDB     0.000GB
> use studentDB
switched to db studentDB
> db.studentlog.find()
{ "_id" : ObjectId("5e09c532dd8f0701c7d2ce1f"), "card name" : "eric", "img_filename" : "eric.jpg",
"card_id" : 128831138599, "accuracy" : 91.28, "face" : "eric" }
{ "_id" : ObjectId("5e09c5f9dd8f0701c7d2ce20"), "card name" : "jack", "img_filename" : "jack.jpg",
"card_id" : 128355935990, "accuracy" : 93.42, "face" : "jack" }
{ "_id" : ObjectId("5e09c67add8f0701c7d2ce21"), "card name" : "michael", "img_filename" : "michael.jpg",
"card_id" : 136842397452, "accuracy" : 91.14, "face" : "michael" }
{ "_id" : ObjectId("5e09c6a8dd8f0701c7d2ce22"), "card name" : "ken", "img_filename" : "ken.jpg",
"card_id" : 150134879562, "accuracy" : 90.38, "face" : "ken" }
{ "_id" : ObjectId("5e09c6dbdd8f0701c7d2ce23"), "card name" : "andy", "img_filename" : "andy.jpg",
"card_id" : 134507895035, "accuracy" : 90.97, "face" : "andy" }
{ "_id" : ObjectId("5e09c8ed3490358ab8bc6561"), "card name" : "alan", "img_filename" : "alan.jpg",
"card_id" : 165789420354, "accuracy" : 92.42, "face" : "alan" }
>
```

圖 16. MongoDB 儲存學生資料與辨識結果

## Browse Directory

/user/nifi/student\_img

Show 25 entries

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	nifi	supergroup	485.92 KB	Apr 10 21:37	3	128 MB	alan.jpg
-rw-r--r--	nifi	supergroup	475.8 KB	Apr 10 21:08	3	128 MB	andy.jpg
-rw-r--r--	nifi	supergroup	523.64 KB	Apr 10 21:46	3	128 MB	eric.jpg
-rw-r--r--	nifi	supergroup	492.79 KB	Apr 10 21:25	3	128 MB	jack.jpg
-rw-r--r--	nifi	supergroup	493.82 KB	Apr 10 21:32	3	128 MB	ken.jpg
-rw-r--r--	nifi	supergroup	435.32 KB	Apr 10 21:21	3	128 MB	michael.jpg

Showing 1 to 6 of 6 entries

圖 17. Apache Hadoop HDFS 儲存學生照片

## 二、辨識結果介於預設值與最底下限值

當戴上配件時準確率下降到 75.63%，此時會將圖像送往後端進行重新訓練，如圖 18 所示。圖 19 為更新後的模型並再進行一次辨識，此時可以看到準確率從 75.63% 提升到 93.28%。另外再測試沒配件時人臉辨識是否有誤，如圖 20 所示，用原始樣貌所辨識出來的結果為 90.55%。我們將五位實驗者的人臉辨識結果做平均(表 1)，原本未訓練戴口罩的模型時準確率只到 71.4%，訓練過後的模型準確率可達 89.6%，而未戴口罩時的人臉辨識都能達到 90% 以上的準確率。後端將學生證資料儲存在 MongoDB(圖 21)、將照片儲存在 Apache Hadoop HDFS，之後將訓練好的模型儲存在 Apache Hadoop HDFS(圖 22)，之後 Apache Kafka 將圖片送到 Raspberry Pi(圖 23)。

表 1.平均學生人臉辨識結果

實驗者 訓練前/後	實驗者一 (jack)	實驗者二 (michael)	實驗者三 (alan)	實驗者四 (andy)	實驗者五 (ken)	平均辨識結果
原始學生人臉 (訓練前)	93.42%	91.14%	92.42%	90.97%	90.38%	91.6%
戴口罩 (訓練前)	75.63%	69.26%	68.24%	68.35%	75.58%	71.4%
戴口罩 (訓練後)	93.28%	95.39%	94.62%	84.10%	80.93%	89.6%
原始學生人臉 (訓練後)	90.55%	96.99%	90.40%	91.97%	84.63%	90.9%

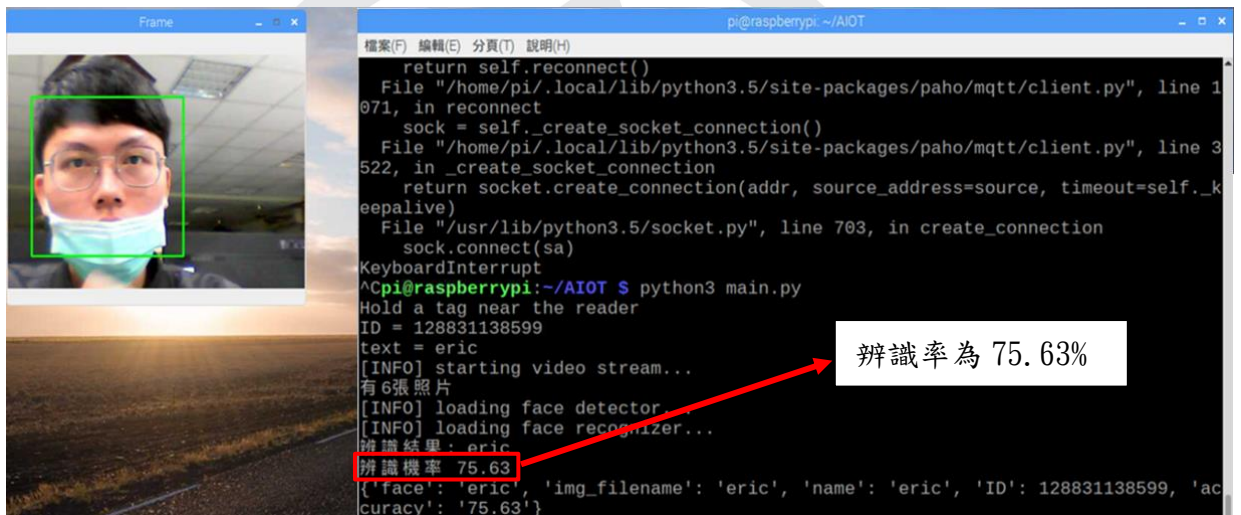


圖 18. 學生人臉模型未訓練前

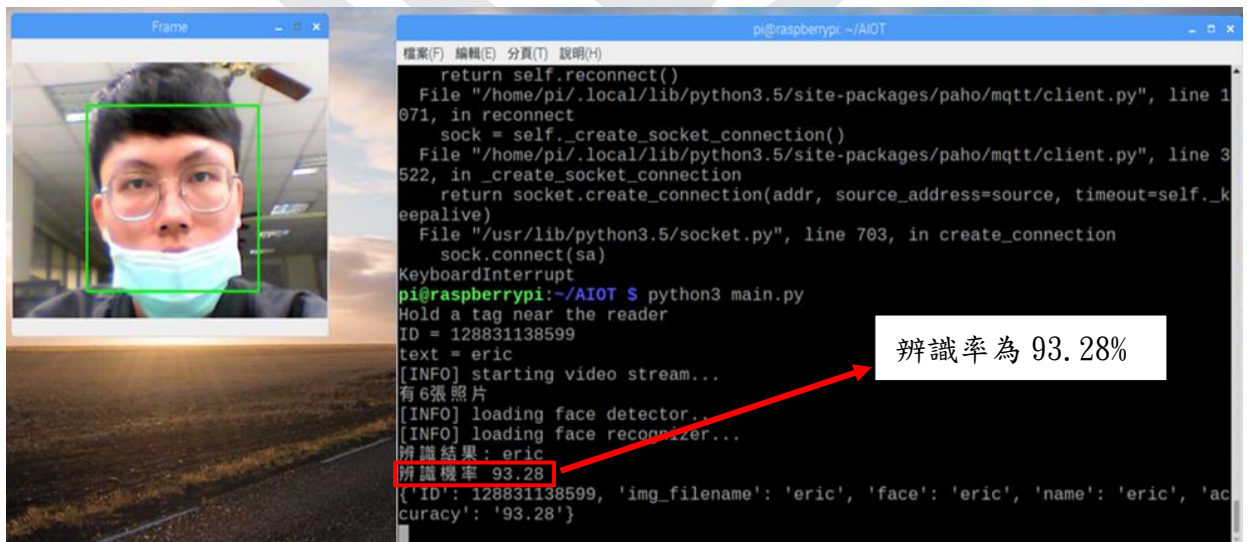


圖 19. 學生人臉模型訓練後

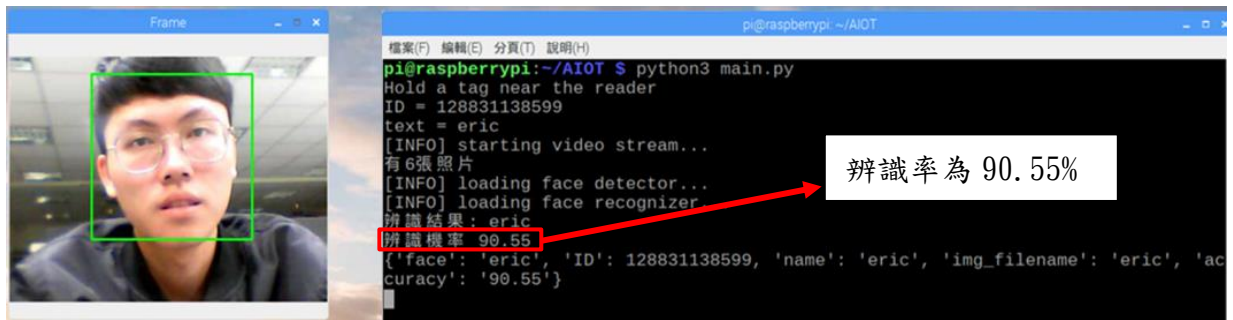


圖 20. 測試學生未戴配件時的辨識率

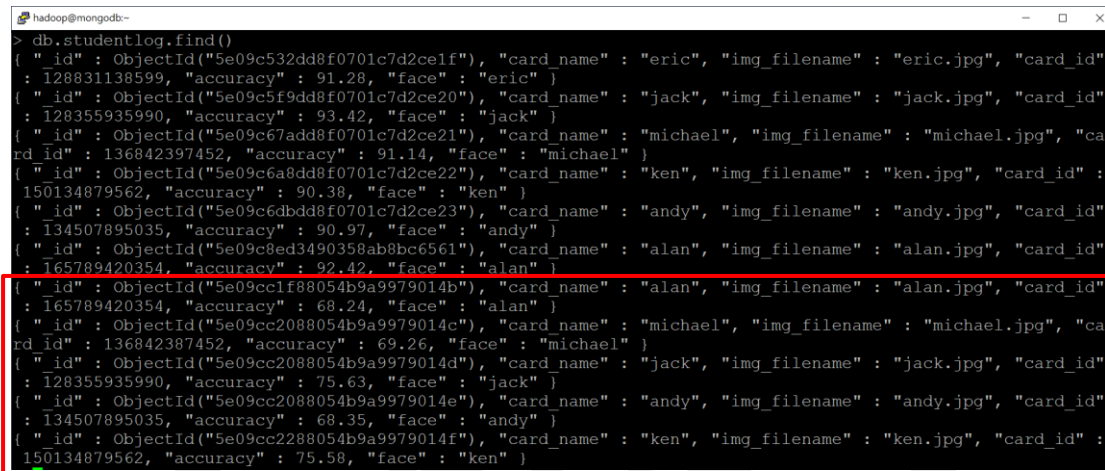


圖 21. MongoDB 儲存學生資料與辨識結果

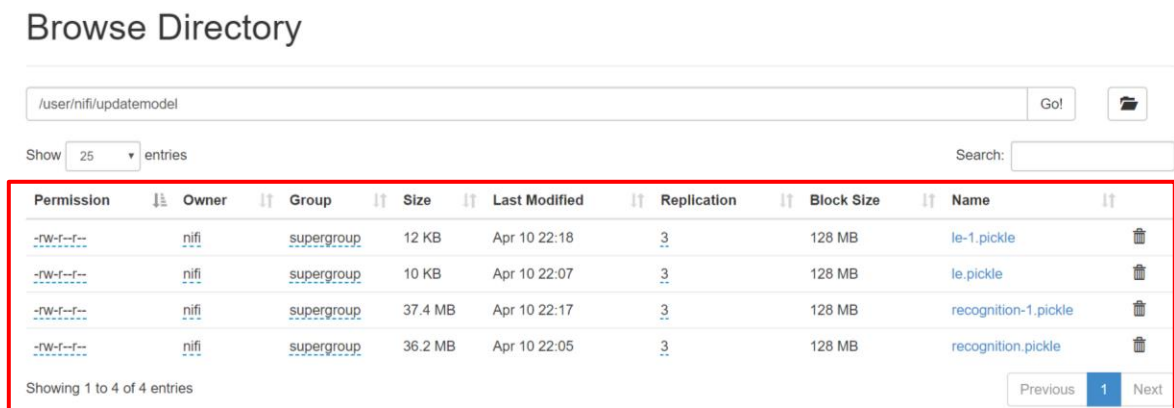


圖 22. Apache Hadoop HDFS 儲存更新後的學生人臉辨識模型

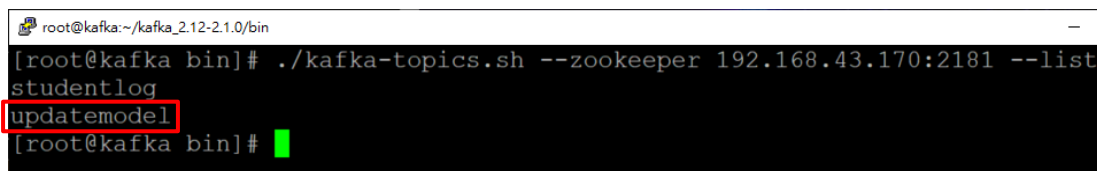


圖 23. 資料暫存於 Apache Kafka 之畫面

### 三、辨識結果低於最低下限值

從圖 24 發現辨識出來的結果與學生證資料不符時，會在 LCD 上顯示無法通過的訊息(圖 25)。之後將辨識結果、學生證資料儲存於 MongoDB(圖 26)，照

片儲存在 HDFS 至後端進行儲存，最後 Apache Kafka 發布/訂閱的方式(圖 27)發送一則 Email 給管理人員(圖 28)。

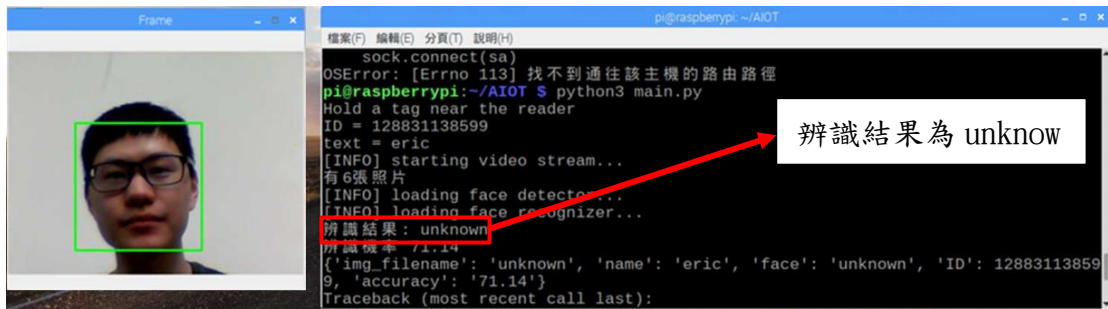


圖 24.學生證非本人持有

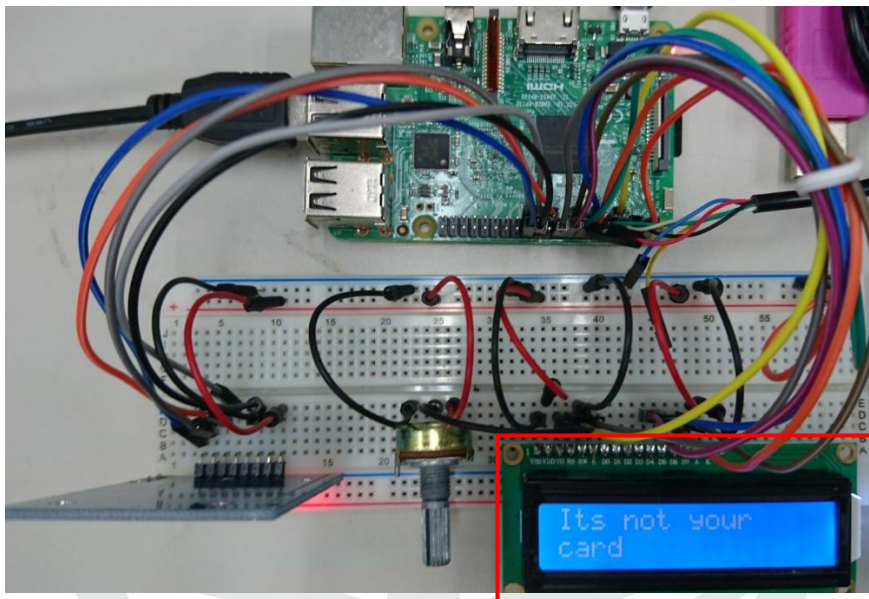


圖 25. 螢幕顯示訊息

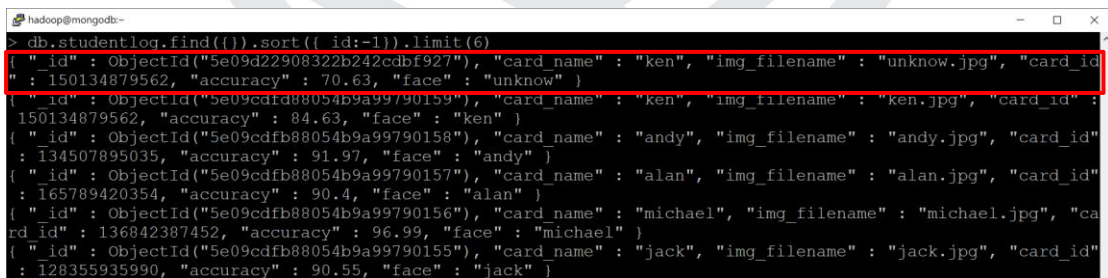


圖 26.通報紀錄檔

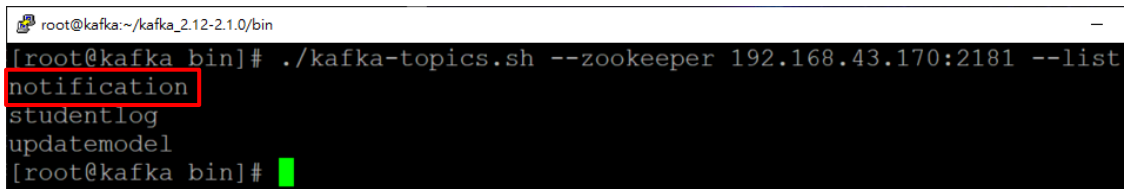


圖 27.資料暫存於 Kafka 之畫面



圖 28. Email 寄送至管理人員信箱

## 伍、結論

本研究論文應用學生點名系統建構一個 AIoT 的系統架構，此系統包含前端 IoT 硬體設備能自動化偵測與人臉辨識、後端大數據平台進行儲存與分析、以及運用人工智慧的方式更新與重新訓練模型，本系統架構由於採用雲端運算的開源軟體系統，因此同時具備穩定性、可靠性以及可擴展性等特色。

本實驗設計雖然是以人臉辨識作為應用基礎，但事實上其應用範圍並不限於人工智慧的人臉辨識，尤其是許多工業應用常常需要執行產品的正常辨識判斷與異常樣品檢測，在正常判斷與異常檢測的辨識範圍之間，人工智慧模型就必須重新再訓練以因應可能的環境變化或產品規格需求，本實驗設計的自動化重新訓練機制都可以滿足這些工業產品的品質管理應用。

## 陸、參考文獻

### 一、中文部分

[10] 應用充電站—RFID 發展歷程與組成元件。上網日期: 2019 年 11 月 29 號，檢自: <https://www.digitimes.com.tw/tw/dt/n/shwnws.asp?cnlid=10&cat=&cat1=&id=56650>

[11] 人臉檢測分析之 Haar-like, Adaboost, 級聯(Cascade)。上網日期: 2019 年 10 月 30 號，檢自: <https://face2ai.com/MachineLearning-Haar-like-Adaboost-cascade>

[12] Kafka 維基百科。上網日期: 2019 年 12 月 15 號，檢自: <https://zh.wikipedia.org/wiki/Kafka>

[13] 認識大數據的黃色小象幫手 - Hadoop。上網日期: 2019 年 9 月 21 號，檢自: <https://www.inside.com.tw/article/4428-big-data-4-hadoop>

[14] MongoDB 維基百科。上網日期: 2019 年 12 月 15 號，檢自: <https://zh.wikipedia.org/wiki/MongoDB>

## 二、英文部分

- [1] Amos, B., Ludwiczuk, B., & Satyanarayanan, M. (2016). OpenFace: A general-purpose face recognition library with mobile applications.
- [2] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*(pp.273–297), 20(3).
- [3] Chang , K. and Chiang, M.(2019). Design of Data Reduction Approach for AIoT on Embedded Edge Node. *2019 IEEE 8th Global Conference on Consumer Electronics* (pp. 899-900), Osaka, Japan.
- [4] Chanthakit , S., Keeratiwintakorn , P. and Rattanapoka , C.(2019). An IoT System Design with Real-Time Stream Processing and Data Flow Integration. *2019 Research, Invention, and Innovation Congress* (pp. 1-5), Bangkok, Thailand.
- [5] Faruqe, M.O., & Hasan, M.A. (2009). Face recognition using PCA and SVM. *2009 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*(pp.97-101).
- [6] Gupta , I., Patil, V., Kadam C. and Dumbre, S.(2016),Face detection and recognition using Raspberry Pi. *2016 IEEE International WIE Conference on Electrical and Computer Engineering* (pp. 83-86.), Pune.
- [7] Isah, H. and Zulkernine , F.(2018). A Scalable and Robust Framework for Data Stream Ingestion. *2018 IEEE International Conference on Big Data* (pp. 2900-2905.), Seattle, WA, USA.
- [8] Kim, C. and Son, S.(2018). A Study on Big Data Cluster in Smart Factory using Raspberry-Pi. *2018 IEEE International Conference on Big Data* (pp. 5360-5362), Seattle, WA, USA.
- [9] Schroff , F., Kalenichenko , D. and Philbin, J.(2015). FaceNet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 815-823), Boston, MA.