

國立彰化師範大學

APP 程式設計&人工智慧

期末專題報告

發票助手

指導老師: 陳巧旻老師

作者：數四乙 林祐辰 (S0822132)

作者：數四乙 王之炫 (S0822135)

民國 112 年 6 月 20 日

目 錄

一、 前言	3
二、 系統設計與架構.....	3
三、 Android studio application.....	4
四、 API Sever	5
五、 Web server&Database	7
六、 YoloV5 模型實作.....	9
七、 YoloV8 模型實作.....	11
八、 Google Cloud	13
九、 問題與討論	14
十、 成果展示.....	15
十一、 參考資料.....	22

一、前言

在我們日常的生活中心，我們每天都會前往商店購買生活必需品。在交易的過程中，商家會提供給我們交易的發票，每張發票都擁有獨特的號碼。而財政部每兩個月都會公布中獎的號碼，如果我們的發票中了獎，就能獲得獎金。基於這樣的背景，我們希望開發一個能夠協助整理和紀錄發票的應用程式。

目前，最常見的發票類型包括電子發票和傳統發票。電子發票上面印刷了包含資訊的 QR code，讓我們能夠輕鬆地讀取發票的資訊。然而，傳統發票上面只有印刷發票的資訊，沒有提供快速讀取的媒介。所以我們為了增強傳統發票的輸入功能，透過結合了人工智慧的技術，讓我們能夠快速地新增發票。

二、系統設計與架構

在我們的專題製作中，考慮到目前財政部的電子發票服務平台 API 僅限企業使用，為了在我們的 APP 中實現相應功能，我們決定自行開發我們所需要的 API。除了開發 APP 和相關 API 之外，我們融合了 YOLO (You Only Look Once) V5 物件偵測演算法在我們的專題中，透過該演算法可以在照片中取得傳統發票的發票號碼位置，並利用 Google Cloud 中的 Vision API 進行 OCR（光學字元辨識）的處理，對於使用者來說，只須要對發票拍一張照片，就可以新增發票。

為了使我們開發的 API 能夠順利運行，我們申請了 Lionfree 所提供的網站服務，並將 API 部屬在申請的網域下面，提供應用程式進行呼叫。此外，我們也簡單設計了一個網站和管理員後台，除了提供用戶了解我們所設計的 APP，更重要的是讓管理員可以透過網站界面更新資料，讓管理員不用反覆進入資料庫進行操作。

為了實現對傳統發票的 AI 辨識功能，我們需要一台可以運行 Python 的伺服器，為了讓這個功能更加完整，那個伺服器還需要能夠提供固定 ip 來給我們的應用程式進行呼叫。因此最後我們選擇使用 Google Cloud 的 Compute Engine 作為我們 AI 發票辨識的伺服器，用於接收圖片進行發票號碼的辨識。

三、Android studio application

應用程式簡介

發票助手是以會員、發票、首頁、兌獎與分析共五個功能頁面所組成的發票管理軟體，其中包含了完整的登入系統、後端資料庫應用與前端功能介面，方便使用者迅速管理發票、分析消費習慣與發票兌獎。

ZXing(Zebra Crossing)

由 Google 開發的一個條碼解析與生成的開源庫，包含多種一維與二維條碼規格。在本專案中利用此開源庫依照指定規格(code39)來生成手機載具條碼，以供用戶於消費時使用。

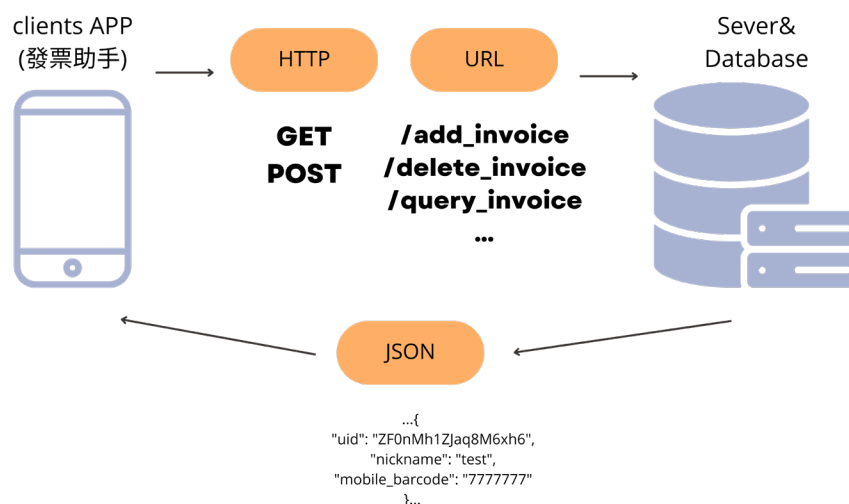
ML Kit

Google ML Kit (前身為 Firebase ML-Kit)是由 Google 所開發的機器學習開發套件，其支援了各項常見機器學習相關功能。在本專案中利用其條碼辨識之功能來實現電子發票的讀取，方便用戶快速將發票內容加入資料庫中。

MPAndroidChart

MPAndroidChart 是一款專門用於移動端圖表繪製的強大第三方套件。在本專案中利用其繪製圓餅圖，來幫助使用者快速了解自身消費習慣，協助使用者管理與優化收支內容。

四、API Sever



開發動機

當我們開發發票助手這個 APP 時，我們發現財政部的 API 現在只提供給企業使用，不支援一般的開發者。為了克服這個限制，我們不得不自己開發了一系列的 API 來協助我們的應用程式運作。

透過自己開發 API，我們能夠獨立地建立一個功能完整且豐富的應用程式。更重要的是，我們的應用程式就不需要因為財政部 API 的限制，對於功能進行取捨。這使得我們能夠自由地設計和定制各種功能，以滿足使用者的需求，例如發票對獎、消費分析、傳統發票 AI 輸入等。

此外，這也給予我們更大的靈活性和擴展性。我們能夠根據使用者的回饋與前端的需求不斷優化和增強我們的 API，提供更多有價值的功能和整合，讓我們的發票助手更加強大和實用。

API 設計

我們總共開發了 11 個 API，其中 10 個為普通的 Web API，主要是提供應用程式和 Lionfree 上託管的 Web server 以及 Database 進行互動。最後一個 API 是特殊的 API，他是我們部署在 Google Cloud 上的 Python 伺服器，可以接收發票圖片並進行分析以及回傳結果，過程中不需要人為的控制，如果辨識的結果不符合發票號碼的格式，也會回傳對應的通知給使用者。

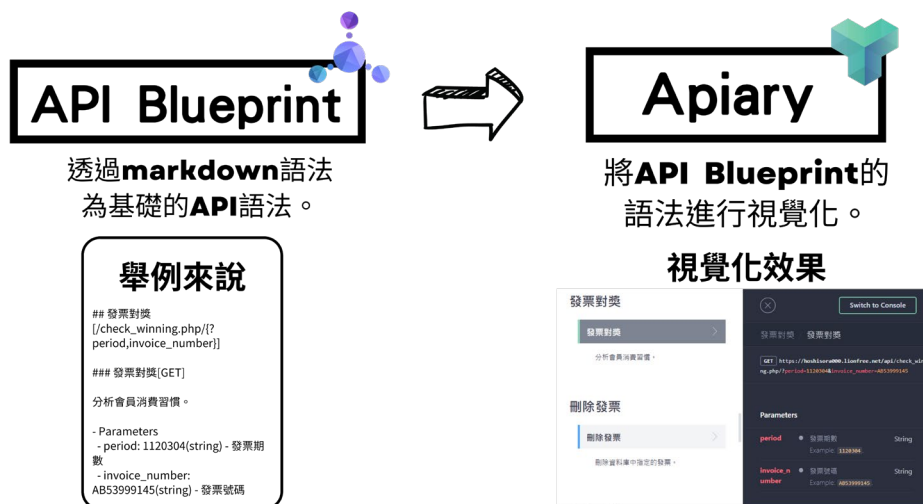
普通 API(呼叫連結"https://hoshisora000.lionfree.net/api")

編號	API	URL	Request	Response
1	新增發票	/add_invoice.php	POST-Form-data	JSON
2	新增會員資料	/add_member_inf.php	POST-Form-data	JSON
3	消費分析	/analysis.php	POST-Form-data	JSON
4	會員中獎檢查	/check_invoice.php	POST-Form-data	JSON
5	發票對獎	/check_winning.php	GET	JSON
6	刪除發票	/delete_invoice.php	POST-Form-data	JSON
7	取得時間	/get_time.php	GET	JSON
8	查詢發票	/query_invoice.php	GET	JSON
9	查詢會員資料	/query_member.php	GET	JSON
10	更新會員資料	/update_member_inf.php	POST-Form-data	JSON

Python API(部屬於 Google Cloud Compute Engine)

編號	API	URL	Request	Response
11	傳統發票辨識	http://34.96.209.0:3000/	POST-Form-data	text

APIARY



在本次專題製作中，我們使用了 Apiary 來呈現 API 說明文件，以促進前端和後端開發者之間的協作。透過這個方式，後端開發者可以展示 API 的使用方式給前端開發者，讓他們知道如何與後端程式進行互動，加速開發的效率。此外，由於 Apiary 是部屬在網路上的，因此可以即時同步版本，相對於傳統的書面說明，降低了前後端開發者文件版本不同步的可能性。

五、Web server&Database

Lionfree

Lionfree 作為一個台灣本土提供免費虛擬主機服務的平台，其經營團隊是由一群社群志工組成，希望透過提供網站空間託管服務，為台灣的資訊教育基礎建設提供幫助。

在本次專題實作中，我們開發的 10 個普通 API 全部都部署在 Lionfree 網站上，此外我們還運用了 Lionfree 所提供的資料庫服務。如果應用程式需要存取資料庫都是透過我們的 API，而非由應用程式直接連結資料庫，透過這樣的架構安排，可以確保我們的資料庫 IP 以及登入帳號密碼都不會洩漏出去。更重要的是，因為 Lionfree 的機房提供了相當多的安全保障，讓我們對於我們資料的安全有著更多的信心。

Database

在我們的專題中，我們選擇了 MariaDB 作為我們的資料庫。MariaDB 是一個開源的關聯式資料庫管理系統，作為 MySQL 的一個分支版本，有著跟 MySQL 高度的相容性。與 MySQL 相比，MariaDB 提供了更多的功能和改進，同時保持了高度的穩定性和可擴展性。

在我們的資料庫中總共包含四個資料表，分別是 bulletin_board(網站留言板資料表，儲存訪客在我們網站的留言)、member(會員資料表，記錄會員資料)、member_invoice(會員發票資料表，記錄會員的發票)以及 winning_numbers(中獎號碼資料表，儲存中獎號碼)。

表 1 網站留言板資料表

表格名稱		bulletin_board				
欄位名稱	中文名稱	資料型態	長度	可否空值	預設值	主鍵
id	留言編號	varchar	11			V
name	姓名	varchar	20			
email	電子信箱	varchar	40			
phone	電話號碼	varchar	20			
message	訊息	text				

表 2 會員資料表

表格名稱	member					
欄位名稱	中文名稱	資料型態	長度	可否空值	預設值	主鍵
uid	會員 id	varchar	60			V
nickname	暱稱	varchar	10			
mobile_barcode	手機條碼	varchar	10	V	NULL	

表 3 會員發票資料表

表格名稱	member_invoice					
欄位名稱	中文名稱	資料型態	長度	可否空值	預設值	主鍵
uid	會員 ID	varchar	60			V
invoice_number	發票號碼	varchar	10			V
date	消費日期	date				
time	消費時間	time				
money	消費金額	int	11			

表 4 中獎號碼資料表

表格名稱	winning_numbers					
欄位名稱	中文名稱	資料型態	長度	可否空值	預設值	主鍵
period	發票期數	int	7			V
super_special	特別獎	varchar	8			
special	特獎	varchar	8			
head1	頭獎 1	varchar	8			
head2	頭獎 2	varchar	8			
head3	頭獎 3	varchar	8			

Website

在這個專案中，除了開發我們的應用程式之外，我們還額外建立了相關的網站，讓用戶可以更深入地了解我們的應用程式。同時，我們也提供了一個管理員頁面，讓管理員能夠透過網頁存取資料庫的內容。這樣一來，在進行資料更新時，管理員就不需要直接進入資料庫進行操作，透過直觀的操作介面，讓管理員可以輕鬆的更新資料。

六、YoloV5 模型實作

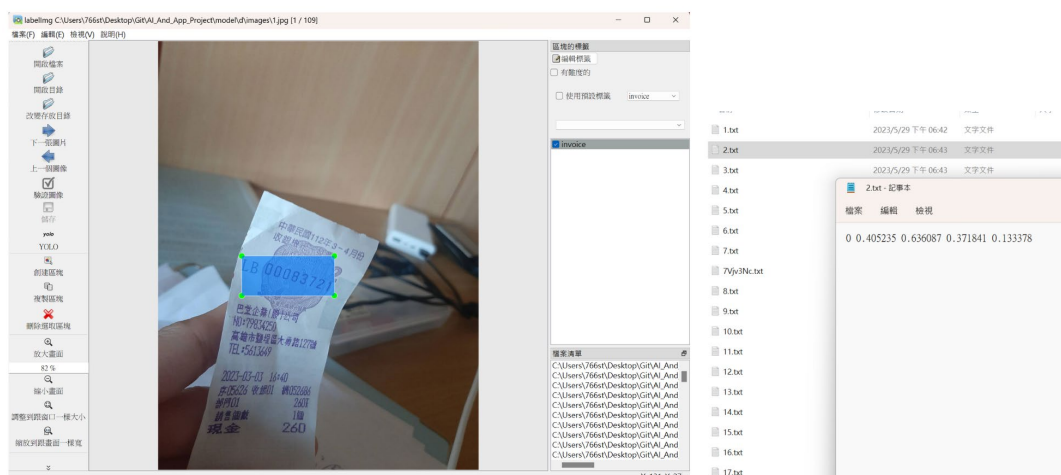
模型簡介

YOLO (You Only Look Once) V5 是一種物件偵測演算法，它能夠快速且準確地識別圖像中的物件。YOLO V5 是由 Alexey Bochkovskiy、Chien-Yao Wang 和 Hong-Yuan Mark Liao 於 2020 年開發的。

YOLO V5 的基本原理是將物件偵測問題轉化為一個回歸問題。它將圖像分為一個固定大小的網格，每個格子預測該格子內是否有物件以及物件的類別和邊界框的位置。在訓練過程中，YOLO V5 使用了標註的圖像數據集進行監督學習。通過計算預測結果與真實標註之間的差異，使用反向傳播算法來調整模型的參數，從而使模型能夠更好地識別物件。

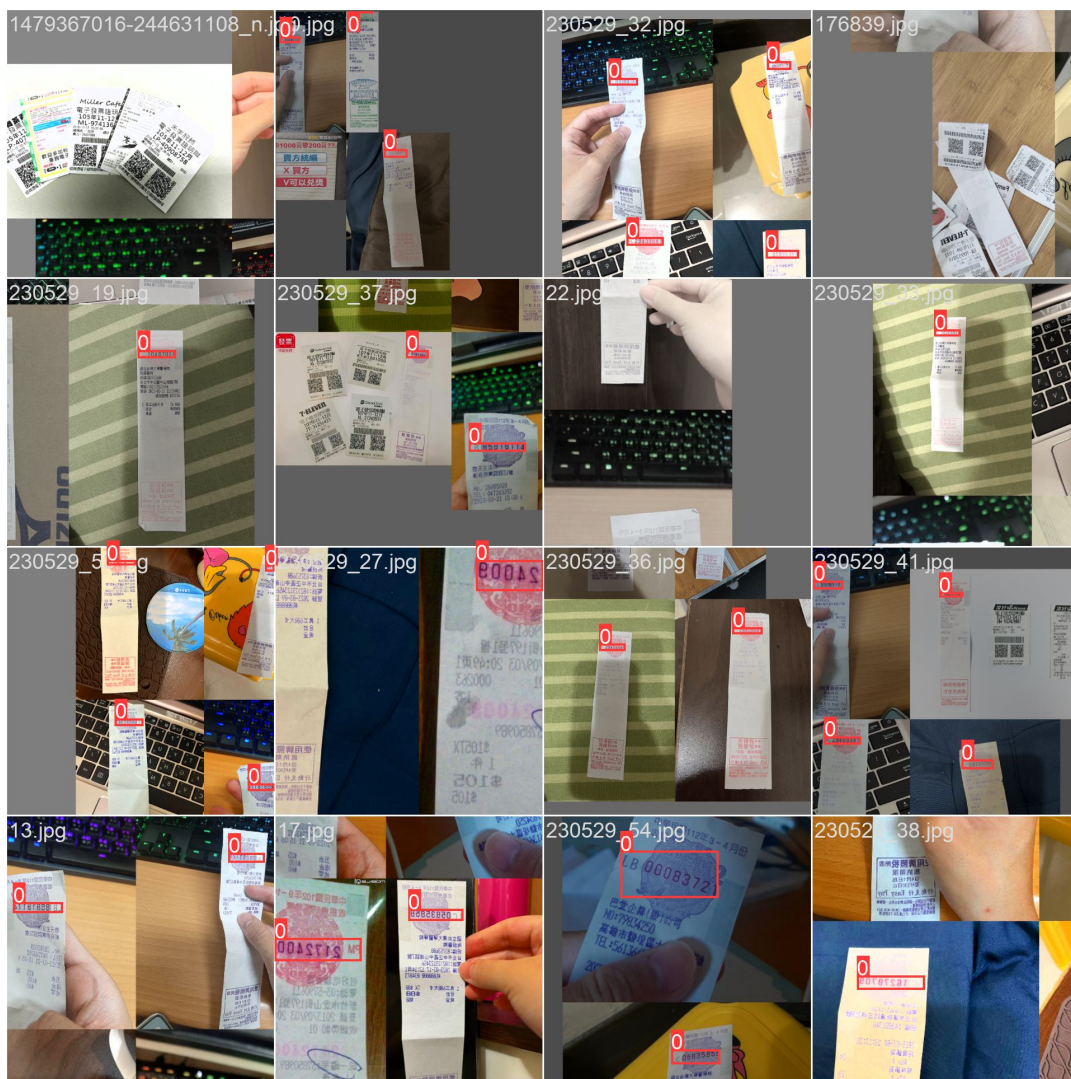
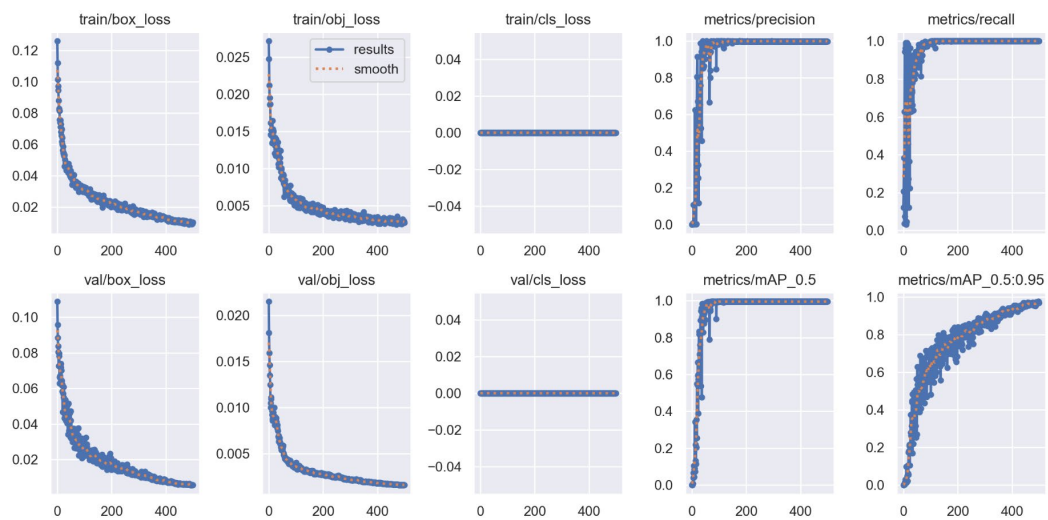
訓練流程

1. git clone <https://github.com/ultralytics/yolov5.git>
2. git clone <https://github.com/heartexlabs/labelImg.git>
3. 蒐集訓練用圖片並使用 labelImg 對訓練用圖片標註標籤



4. 依照規範，在 yoloV5 目錄之下建立 dataset.yaml
5. 使用指令 "python train.py --img 320 --batch 16 --epochs 500 --data dataset.yaml --weights yolov5s.pt" 開始訓練模型

訓練結果



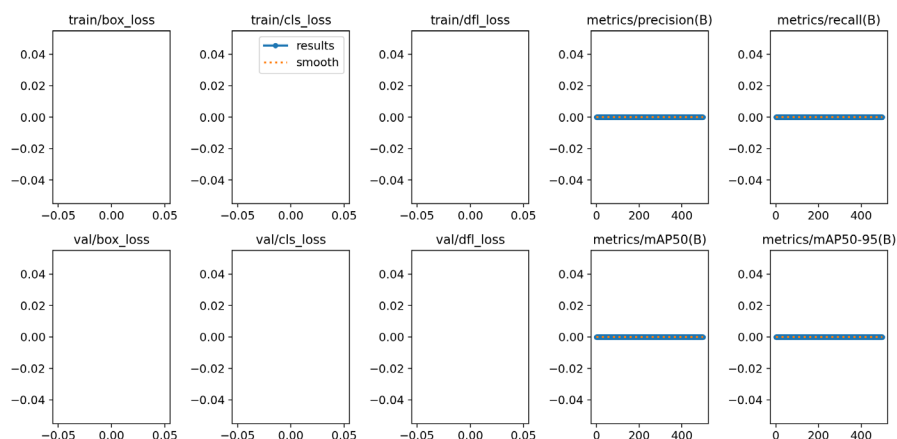
七、YoloV8 模型實作

模型簡介

YOLOV8 是 ultralytics 公司在繼 YOLOV5 後，最新發布(2023 年 1 月 10 日開源)的一個重大更新版本，相較於先前版本只有物體偵測的功能，在本次版本中新增了圖像分類和實例分割的功能。不過 ultralytics 官方在 Github 上並非直接將其命名為 YOLOV8，而是直接使用 ultralytics 這個詞，原因是因為官方希望將它定位為一個算法框架而非特定的算法，因此如果實際去 Github 上查看他的內容會發現裡面除了 V8 版本之外，還包含了早期發布的 YOLO 版本。

NVIDIA GTX16XX 顯卡訓練異常

在我們本次實作的過程中，當我們使用 NVIDIA GTX1660TI 的顯卡訓練模型時，發現了一個很有趣的議題，就是訓練的結果相當失敗(參考下圖)，然而如果使用 CPU 去進行訓練，出來的結果就相當正常。當我們仔細的去檢查訓練過程時，發現訓練過程的 loss 的是 nan，這算是一個相當不正常的過程，因此我們對於這個問題進行了些許的探究。



Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size			
1/5	0.659G	nan	nan	nan	20	128: 100%	7/7	[00:03<00:00, 1.85	
C:\Users\GF75\anaconda3\envs\AI_class\lib\site-packages\torch\optim\lr_scheduler.py:138: UserWarning: Detected call of `lr_scheduler.step()` before `optimizer.step()`. In PyTorch 1.1.0 and later, you should call them in the opposite order: `optimizer.step()` before `lr_scheduler.step()`. Failure to do this will result in PyTorch skipping the first value of the learning rate schedule. See more details at https://pytorch.org/docs/stable/optim.html#how-to-adjust-learning-rate									
		Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	4/4 [00:04<0
		all	109	107	0	0	0	0	
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size			
2/5	0.656G	nan	nan	nan	22	128: 100%	7/7	[00:02<00:00, 2.94	
		Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	4/4 [00:00<0
		all	109	107	0	0	0	0	
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size			
3/5	0.656G	nan	nan	nan	20	128: 100%	7/7	[00:02<00:00, 3.31	
		Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	4/4 [00:00<0
		all	109	107	0	0	0	0	

網路上對於這個問題的討論非常的多，但大部分的解決方法都是靠降低 pytorch 以及 CUDA 的版本去解決，可是這樣的解決方法我們認為並沒有太大的實用性，因為降低他們的版本很容易發生系統不支援等等的問題。因此我們希望去找尋他發生的原因以及比較理想的解決辦法，為此我們翻閱了許多網路上的資料，也分別進行測試，最後發現問題可能是 Automatic mixed precision 的問題。

深度學習模型的訓練目前通常都是利用 GPU 強大的浮點數運算能力，來完成神經網路參數的學習過程。現在的模型訓練過程基本上都是使用單精度和雙精度，Automatic mixed precision 是一種用於加速深度學習訓練的技術，透過在訓練的過程中動態使用不同精度的數據來提高效能，在確保準確度的前提下，達到節省時間和資源的效果。

雖然對於這個問題實際的成因我們並未找尋出來，但是我們還是有得到一個不太理想的解決方法，透過輸入禁用 Automatic mixed precision 功能的指令，我們可以成功的訓練出模型。然而比較可惜的部分是，他只解決了訓練的部分，而驗證的部分這個問題會依舊存在，但我們如果將訓練出來的模型拿去進行預測，可以發現預測的表現並沒有問題。

測試環境：

CPU: Intel(R) Core(TM) i7-9750H

GPU: NVIDIA GeForce GTX 1660 Ti

Pytorch 版本:2.0.1

CUDA 版本: 11.8

泛化能力討論

在我們本次專題製作的過程中，我們分別以相同的數據，分別訓練了 V5 和 V8 的模型，而在我們測試的過程中發現，V5 模型泛化能力非常優秀，基本上預測的結果與位置都相當準確，反觀 V8 雖然大多數情況也都能順利判斷出來，但準確性以及範圍卻沒有 V5 那麼優秀，考量到專題的穩定性，我們最後還是選擇使用了 V5 的模型進行發票號碼位置辨識。

八、Google Cloud

Google Cloud 是由 Google 建立的雲端平台，提供一系列的雲端服務和工具，包含了計算、網路、儲存、機器學習以及資料分析等眾多產品。相對於傳統的實體主機伺服器，採用這類雲端伺服器可以降低人力以及設備的租借成本，更重要的是所有的服務都是採用按須付費，可以靈活的調配所需要的資源，達到降低成本的目的。

Compute Engine

Compute Engine 是 Google Cloud 提供的基於虛擬機器的計算服務。它允許使用者在 Google 的全球性資料中心中建立和管理虛擬機器。更重要的是 Compute Engine 提供了靈活的虛擬機器，使用者可以根據需求選擇不同的虛擬機器配置，包括處理器、記憶體、儲存和網路等等，使用者可以根據需求，自訂虛擬機器的規模和配置，並建設所需要的作業系統以及軟體環境。

Compute Engine 還提供了豐富的網路功能和工具，讓使用者能夠輕鬆管理和配置虛擬機器實例的網路連接，像是我們可以自訂防火牆規則和負載平衡設定，以確保伺服器的穩定和安全性。

Vision API

Vision API 是 Google Cloud 提供的一項強大的影像分析服務，它利用先進的機器學習技術，能夠自動識別和解析圖像中的內容。在本次專題中，我們主要使用的是 OCR（Optical Character Recognition，光學字符識別）這一個功能。OCR 是一項技術，可以將圖像中的文字轉換為可編輯和可搜索的文本，而 Vision API 的 OCR 功能可以自動檢測圖像中的文字，並將其識別和提取出來，生成可供應用程式處理和分析的文本數據。

如果我們使用 Vision API 的 OCR 功能其實非常簡單，我們只需將圖像傳遞給 Vision API，它將返回一個包含識別到的文字和其對應位置的結果。在我們的專題實作中，因為我們只需要對發票號碼做 AI 辨識，為了避免發票上其他資訊影響辨識結果，當伺服器接收到一張圖片時，會先利用 YOLO 模型進行辨識，擷取出包含發票號碼的區域後，才將我們的處理過的圖片傳送給 Vision API 進行 OCR 辨識。

九、問題與討論

1. 由於 android studio 不允許在主線程進行 UI 更新，以及當呼叫 API 時會使用不同線程，因此當同時進行 UI 操作與 API 呼叫時，可能會造成資料尚未更新或是呼叫元件未初始化等問題，需要注意整體程式運行的邏輯，確保其運行的先後順序。
2. 在 Google Cloud 上的 Compute Engine 運行 yolo 需要安裝 Pytorch 才能執行，然而我們平常都是使用 conda 去進行安裝，但因為系統安全性的問題我們遲遲無法正常的安裝 conda，導致我們無法輕易安裝 Pytorch，因此我們只能選擇使用 pip 的方式進行。但當我們安裝最新版本的 Pytorch 卻會發生錯誤，遲遲無法正常的安裝，但透過指定版本的方法我們才成功安裝成功。

十、成果展示

帳號登入與註冊頁面



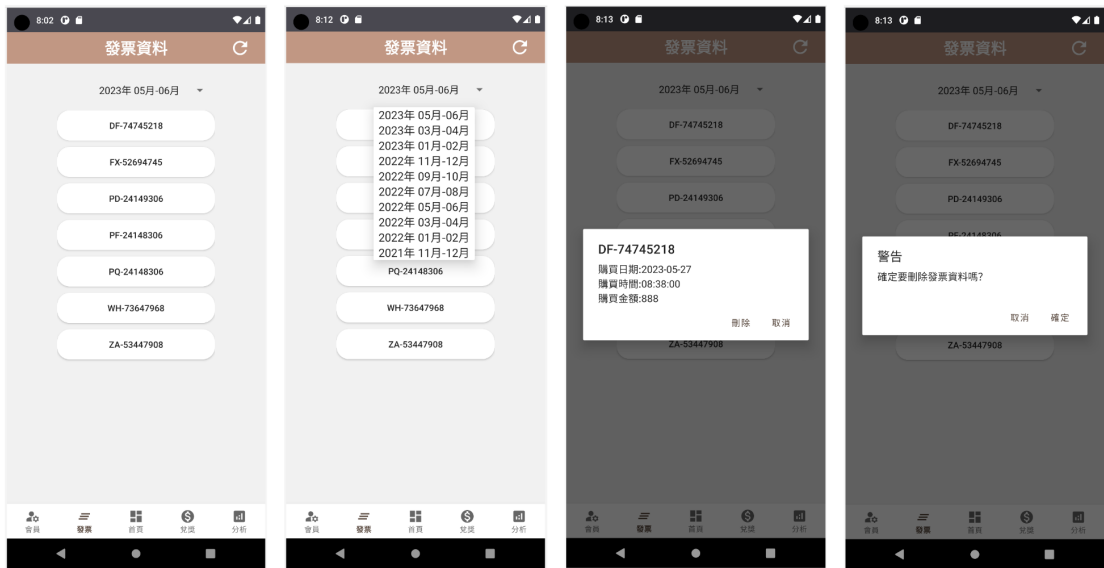
基本頁面



會員頁面



發票頁面

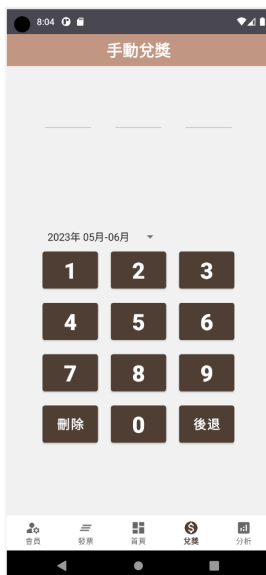


首頁頁面

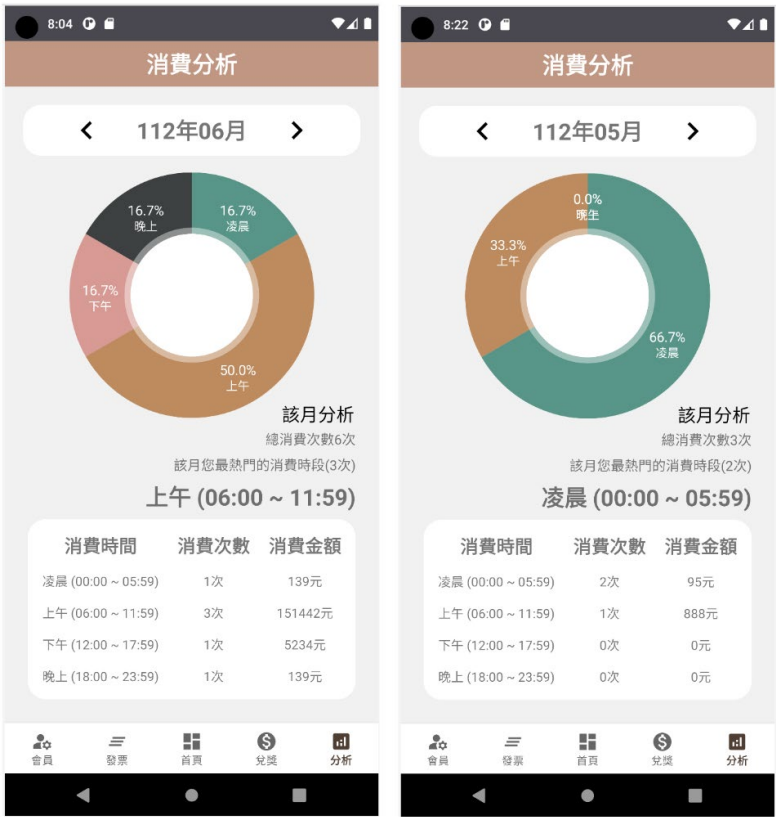




兌獎頁面



分析頁面



網站首頁

發票助手

首頁 關於我們 聯絡我們 管理員後台

發票助手

好的助手，提供你更細緻的消費體驗。

Copyright © ncue_aiapp.project 2023

隱私宣告 · 開發團隊 · 聯絡我們

管理員後台

中獎號碼設定

新增資料

Search:

Show

10

entries

期數	特別獎號碼	特獎號碼	頭獎號碼1	頭獎號碼2	頭獎號碼3	修改/刪除
1120102	06634385	66882140	25722152	93412693	16957025	刪除
1120304	20783987	04135859	94899145	71143793	41055355	刪除

Showing 1 to 2 of 2 entries

Previous1Next

新增發票號碼

發票期數

特別獎

特獎

頭獎1

頭獎2

頭獎3

送出

取消

API 說明文件

Download

API Blueprint

Fork on GitHub

hoshiora000/ai_app

Project

INTRODUCTION

光新的HTTP方法方式:
向服務器發送的消息。

REFERENCE

新增發票

新增會員

會員分析

會員中獎查詢

發票對獎

刪除發票

查詢時間

發票查詢

查詢會員資料

修改會員資料

會員中獎查詢

會員發票中獎查詢

分析會員消費習慣。

發票對獎

發票對獎

分析會員消費習慣。

刪除發票

刪除發票

刪除資料庫中指定的發票。

查詢時間

發票對獎

發票對獎

KEY `https://hoshiora000.11onfree.net/api/check_winning.php?period=1120304&invoice_number=AB53999145`

Parameters

period

發票期數 Example: 1120304

String

invoice_number

發票號碼 Example: AB53999145

String

Request

Production

Raw

Try

Response

200

HEADERS

Content-Type: application/json

十一、參考資料

<HowTo: Creating a Barcode in Kotlin Android>

<https://www.brightec.co.uk/blog/howto-creating-barcode-kotlin-android>

<Android 中 HTTP 网络请求相关问题>

<https://michaelyb.top/2018/08/Android-HTTP/>

<Android 调用摄像头和相册>

https://blog.csdn.net/CSDN_handsome/article/details/121717404?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522167067553616782425177338%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall1.%2522%257D&request_id=167067553616782425177338&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-1-121717404-null-null.142^v68^control,201^v4^add_ask,213^v2^t3_control1&utm_term=kotlin%E8%B0%83%E7%94%A8%E7%9B%B8%E6%9C%BA&spm=1018.2226.3001.4187

<MPAndroidChart-PieChart>

<https://github.com/rezwaneJamee/MPAndroidChart-PieChart/tree/master>

<How to create Settings Page with Modern and Attractive Material UI design in Android Studio>

<https://www.youtube.com/watch?v=Px5u3wz3g-U&t=1263s>

<NaN tensor values problem for GTX16xx users (no problem on other devices)>

<https://github.com/ultralytics/yolov5/issues/7908>

<如何使用 Google GCP 開虛擬主機來架設網站，以及費用公開！>

<https://www.chirue.com/gcp-vm-setting/>

<[apiary][01]設計 API 時好用的工具 - 讓前後端溝通格式不再卡卡 - 概念介紹篇>

<https://blog.alantsai.net/posts/2019/03/apiary-why-user-apiary-different-document-api-format-makes-back-and-front-communicate-easier>