

VCHECK

專題編號：NTPUCSIE-109-002

執行期間：2020 年 9 月至 2021 年 4 月

1. 摘要

在現代，各類電子產品已然成為了生活的必需品，雖然它帶來了生活上的方便性，卻也造成了當代人過度用眼，導致視力異常。

視力異常，主要表現為屈光不正，屈光不正為常見的眼部疾病。常見的表現為：近視、散光、老花及其他眼部疾病，進而可能造成視力的損害，嚴重者甚至可能會直接導致失明。而台灣近視人口比例高居全球第一，根據統計顯示，台灣兩千三百萬人之中就有一千萬人近視，因此定期的視力檢查儼然成為了現代人所必需要做的例行公事，且大概是最多每間隔半年就得要做一次檢查。在這半年之中，並非所有人都可以正確掌握自己眼睛的健康狀況，其中甚至會有人罹患嚴重的疾病，在下一次的定期檢查發現時已經錯過了治療的黃金時間。這些情形，就成為了我們這次專題的動機。

2. 簡介

(一) 研製背景

為了能夠讓現代的人們隨時掌握自己眼睛的健康狀況，以及避免錯過治療的黃金時間，我們決定製作出一個視力檢查的軟體，其中包含最基礎的視力檢查，以及額外的散光、色弱、老花、黃斑部病變、對比度等簡

易測試。而為了增加整個軟體使用上的普遍性以及方便性，我們就想到了將此軟體做成 app，讓使用者可以隨時隨地做測試。

(二) 研究目標

本專題是想透過手機 APP，融合動作捕捉系統，讓使用者在家就能做簡單的視力檢測，使得使用者能了解到自己目前的視力的大概狀況。再加入如散光、色弱、老花以及黃斑部病變等簡單的測驗供給使用者做檢測，讓使用者能夠更廣泛性的了解自己眼睛的健康與否。為了使得使用者所有的測試結果都可以被記錄下來，我們希望能建立出一個資料庫，將所有的測驗結果儲存在裡面。並且考慮到使用者並非只有手機持有人，我們希望能另外儲存每個使用者的個人資料，包含使用者帳號密碼到資料庫裏面，藉由使用者名稱區分開每個人的測試結果。配合完整的使用者登入系統，我們預期可以完全區隔開每個人的個人資料等等，讓每個人都能夠輕鬆查詢到自己的歷史測試結果，又不怕資料會外洩出去。有了這款 APP，人們就能隨時隨地追蹤眼睛的健康情況了。

(三) 主要預期成果

- 使用穩定的動作捕捉系統，來正確判別使用者測視力時手勢比的方向，並以程式判定出測試者大略的視力，

導出結果存入資料庫。

- 以簡單的選擇題或填充題的方式，進行散光、老花等多項測試，完成後導出結果存入資料庫。
- 使用 SQLite 資料庫系統，以表格化的方式保存各個使用者的個人資料以及測試記錄。
- 透過設計使用者帳號系統，以支援多名使用者使用，也使得測試、個人資料不會被其他使用者看到。

3. 專題進行方式與技術

(一) Android Studio

Android Studio 是一個 Android 整合開發工具，基於 JetBrains 軟件公司開發的 IntelliJ IDEA 工具軟件，類似 Eclipse 的 ADT，Android Studio 提供了整合的 Android 開發工具。相較於 Eclipse，Android Studio 不僅運行速度更快，提示補全也更加智能、人性化，並且整合了 Gradle 構建工具、強大的 UI 編輯器，使其更適合用於開發和除錯。

(二) PoseEstimationForMobile

為辨別使用者的手勢方向，我們使用了動作捕捉技術，而此 repo 使用 TensorFlow 技術實現 Convolutional Pose Machine 簡稱(CPM)和 Hourglass 模型。

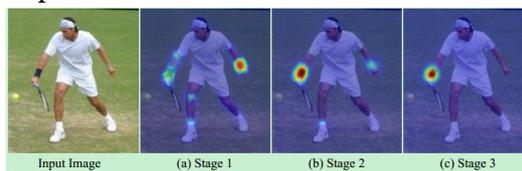
Model	FLOPs	PCKh	Inference Time
CPM	0.5G	93.78	~60 FPS on Snapdragon 845
Hourglass	0.5G	91.81	~60 FPS on iPhone XS (need more test)

I. Convolutional Pose Machine

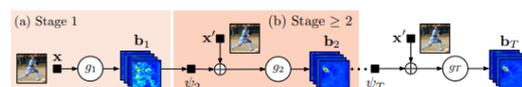
Convolutional Pose Machines 基本上算是 Pose Machines 的延伸，Pose Machines 是一種序列化的預測框架，可以學習資訊豐富的空間模型。而 Convolutional Pose Machine 是將 Convolutional Network 整合進 Pose Machines 中，將其中的特徵擷取以及預測器的部分，改以 CNN 的架構直接取代掉，以學習圖像特徵和圖像相關

的空間模型，提供了對大量的空間模型進行順序性預測的功能。此方法提供自然的學習來加強過程監督，並且補充了反向傳播的梯度，調節了學習的過程，成功解決了訓練過程中梯度消失的難題，達到更為精確的預測。

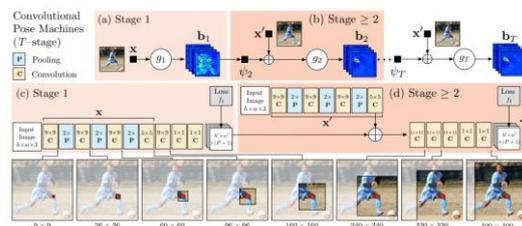
CPM 由一系列的 CNN 組成，並重複輸出每一個關節點的 2D confidence map，每一個 stage，採用圖像特徵和上一個 stage 輸出的 2D confidence map 作為輸入。以下面的圖為例，關節點 confidence map 的資訊讓後續的 stage 提供了線索資訊，因此，CPM 的每個 stage 都可以輸出越來越精細的關節點的 confidence map。



Pose Machine 由 multi-class-predictors 所組成，並利用 gradient boost 對人體每個關節點進行預測，如下圖：



CPM 同時利用 CNN 的優點，和 Pose Machine 的空間建模，如下圖：

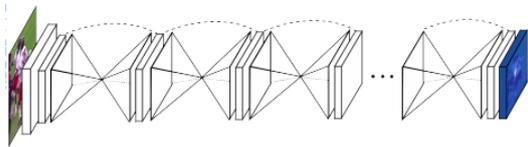


從上圖我們得知，隨著 Convolution 的層數越來越加深，圖片能中偵測關節點的範圍就越大，而越大的偵測範圍則可以同時得到不同關節點的位置，這樣可以提升網路針對不同關鍵點的預測精度。

II. Stacked Hourglass

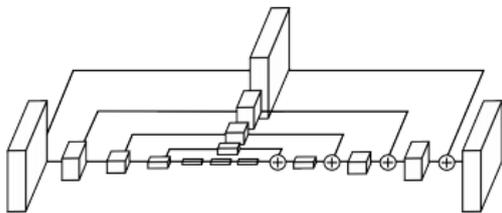
為一種新穎的捲積網絡體系結構，用於對人體的動作進行預測。藉由捕獲與身體相關的各種空間關係，重複地進行 bottom-up、top-down 處理，並以 Intermediate Supervision 的方式提高網絡性能，此架構被稱為 Stack Hourglass 網絡。而該網絡基於 pooling 和 upsampling 的連續步驟而完成。

至於為什麼會稱之為 Stack Hourglass 網絡，主要是因為它長得很像堆疊起來的沙漏，如下圖：



這種堆疊在一起的 Hourglass 結構是對稱的，使用 bottom-up 過程能將圖片從高分辨率降到低分辨率，使用 top-down 過程能將圖片從低分辨率升到高分辨率，pooling 可以將圖片降成很低的分辨率，upsampling 則可以結合多個分辨率的特徵。

為了捕捉每個尺度下的資訊，像是捕捉臉或手時會需要用到局部的特徵，而要捕捉整個人體時，要用到整張圖片的資訊，通常的做法是使用多個 pipeline 分別單獨處理不同尺度下的信息，然後後面再組合這些特徵。



(三) SQLite

SQLite 是遵守 ACID 的關聯式資料庫管理系統，它包含在一個相對小的 C 程式庫中。與許多其它資料庫管理系統有些不同，SQLite 不是一個客戶端/伺服器結構的資料庫引擎，而是被整合在使用者程式中。SQLite 遵

守 ACID，實現了大多數 SQL 標準。它使用動態的、弱類型的 SQL 語法，作為嵌入式資料庫，是應用程式在本地/客戶端儲存資料的常見選擇。與其他一般資料庫相比，其優點是不需要建立一個資料庫系統，也不需要再安裝資料庫伺服器軟體，要使用的時候，只需要將 SQLite 的程式庫編入就可以使用。

4. 遇到困難與解決辦法

困難 1：

最初的時候我們其實並不是打算做成 APP，而是打算實作在電腦上的。但是在尋找可用資源時，一直處處碰壁，最常碰到的問題就是找到可以使用的專案和 model 無法安裝，光是配置環境就會花掉很多時間，且不一定能成功運行。能成功運行的部分也因為摸索使用方法所花掉的時間過長，導致我們在起頭就用掉了大把的時間，但完全沒有任何進展。

解決辦法 1：

我們訂下了一個月的時間，若是配置環境到摸索使用方法花掉的時間超過了一個 month，我們就立刻換下一個專案。在準備換專案的同時，指導教授也剛好建議我們可以嘗試往製作手機 app 的方向邁進，增加其方便性。因此，我們往 app 的方向邁進的同時，分成了兩組摸索不同的專案和 model，時間也縮短成三個星期，增加我們整體的效率。幸運的是，我們很快就嘗試出了一個成功的專案，且大家摸索的速度很快，最終決定使用成功的專案，也就 是 PoseEstimationForMobile。

困難 2：

動作捕捉方面由於我們找到的 model 不太穩定，而且我們又是以 model 抓到的關節點座標進行使用者手勢的判定，因此只要 model 稍微跑

掉一下，我們的手勢判斷就會出錯。

解決辦法 2：

以程式做調控，首先判定的手勢要維持三秒完全相同才會進行最後判定，並且放寬判定的限制，增加誤差值的容許範圍。因上述的調整，雖然造成了在使用上時會有延遲感，但減少了因為不穩定而造成判定錯誤的機會，藉此能夠達到正確的動作判斷。

困難 3：

一開始儲存使用者資料的時候，我們是使用 Android Studio 裡面一個叫 SharedPreferences 的方法進行儲存。這個方法雖然也可以儲存使用者資料，但只要資料的數量變的龐大，很容易就會使得整個 APP 的運行速度下降。並且由於使用此方法時測試紀錄都會被覆寫掉，導致我們做歷史紀錄查詢時只能查詢到最近一次測試的結果。

解決辦法 3：

在尋找儲存資料的方法時，我們剛好在上資料庫系統課程，因此選擇了與 android studio 契合度佳的 SQLite 建立本地的資料庫。除了能用表格化的方式儲存測試結果之外，也排除掉了 SharedPreferences 使用過多導致 app 變慢，降低使用品質的問題，且能將我們所需的使用者個人資料一併進行儲存，透過資料的連結實現多使用者互相看不到彼此資料的功能。

困難 4-1：

一開始儲存測試結果時，我們是以 string 的型態將結果文字直接儲存下來，方便記錄查詢時將結果顯示出來，例如視力檢查是儲存"您的右眼為 1.2"。但此方法並未真正的將結果儲存下來，例如視力檢查應該是以 float 的型態儲存 1.2，且被指導教授

強力反駁。

解決辦法 4-1 & 困難 4-2：

依照指導教授的建議，我們將各個結果依照其適合的型態另外進行儲存，原本 string 型態的欄位作為查詢顯示保留下來。

最後共有三種型態，因此歷史紀錄的表上預定會再多出三個欄位，但一筆歷史紀錄裡只會用到其中一個欄位，導致另外兩個欄位需要插入 null 值，造成 null 值會有過多的問題。

解決辦法 4-2：

參照指導教授以及其他教授的建議，如下圖，我們將所有可能結果分成三張表進行儲存，並且將每筆結果利用項目與結果 ID 區分開來，歷史紀錄的表上僅儲存該測試項目及其結果對應到的結果 ID。查詢歷史紀錄雖然需要先尋找項目再尋找結果 ID 才能得到結果文字，但我們克服了表中會產生 null 值的問題，也減少了每筆歷史紀錄的資料儲存量。



5. 主要成果與評估

- 視力檢查部分藉著程式的調控，能夠正確的判別出測試者的動作，雖然 model 本身不能夠算很穩定，仍然可以大約估測出測試者的視力。
- 除了視力檢測外，還有多項小測試如散光、色弱、老花、對比度以及黃斑部病變等可供使用者一同做檢測。檢測方式為簡易的選擇題或是填充題，讓使用者可以及早發現眼睛的其他問題。
- 設計簡易的使用者帳號系統，使用者可以創建帳號，並輸入使用者名稱、密碼、身高、選擇頭像，將這些資料

存入資料庫之中，即算創建帳號完成，登入後就可以針對該帳號儲存各個測試的記錄。為了防止使用者忘記密碼，在註冊時有額外請使用者自定義一個問題，忘記時只要正確回答該問題即可直接更改密碼。

• 藉由 SQLite，完整保存使用者的資料以及測試結果，並且內部也能夠讓使用者修改資料及查詢歷史記錄，使得使用者可以對自己眼睛的健康與否一目了然。

6. 結語與展望

以一個視力檢查的 app 來說，我相信我們 app 的完整度與實用性是很足夠的，但我們其實可以再多加一些額外的功能增加我們的競爭力。像是可以加入一些有關眼睛保健的知識或者一些專業醫師的文章，讓使用者能更了解眼睛的病狀，以達到預防眼睛惡化的效果，亦或是在使用者眼睛被檢查出有病狀的情況下，可以藉由手機衛星定位去尋找最近的治療場所，並將該場所的資訊如地址、連絡電話等提供給使用者，讓使用者可以盡早獲得妥善的治療。

7. 銘謝

感謝指導教授在我們有設備上的需求時，給予我們很多的援助，並且在我們都是第一次製作 APP 的情況下，給了我們許多關於使用者帳號系統以及安全性上的建議，讓我們能及早發現並修正問題，使我們的 APP 顯得更加完善。也感謝大同大學媒體設計學系，遊戲設計組的羅紫涵同學為我們重新設計了整個 app，讓我們的 app 看起來更加的美觀。

8. 參考文獻

[1]Real-time single person pose estimation for Android and iOS.
<https://github.com/edvardHua/PoseEstimationForMobile>

[2]視力測試 4.8 那個 E 字有多大-365 教育網
<http://www.ceshiti365.com/z/137875.html>

[3]每日頭條-只要三分鐘，即可自測視力！
<https://kknews.cc/zh-hk/digital/bkaaajo.html>

[4]老花測試 DIY，輕鬆快速檢測！
<https://helis.tw/blogs/news/presbyopic-vision-test>

[5]色弱測試/色盲檢查-線上工具

<https://www.ifreesite.com/color-blindness/>

[6]Paper of Convolutional Pose Machines

<https://arxiv.org/abs/1602.00134>

[7]人體姿態辨識 Convolutional Pose Machines

<https://williamchiu0127.medium.com/%E8%AB%96%E6%96%87%E9%96%B1%E8%AE%80%E7%AD%86%E8%A8%98-%E4%BA%BA%E9%AB%94%E5%A7%BF%E6%85%8B%E8%BE%A8%E8%AD%98-convolutional-pose-machines-ac6ca08e96bb>

[8]Paper of Stacked Hourglass Networks for Human Pose Estimation

<https://arxiv.org/abs/1603.06937>

[9]Stacked Hourglass Networks 論文筆記

<https://zhuanlan.zhihu.com/p/45002720>