

長庚資工軟硬體專題期末報告書

專題名稱：

請你跟我這樣畫 兒童精細動作評估系統

執行期間： 112 年 9 月 至 114 年 1 月

指導教授： 魏志達 教授

計畫參與人員：

B1029012 蕭友翔
B1029020 陳昱佑
B1029031 區迅初
B1029054 葉祐齊



執行單位：長庚大學 資訊工程學系

中華民國 2024 年 12 月 24 日

目錄

壹、摘要.....	1
貳、背景、動機與目的.....	2
一、研究背景	2
二、研究動機.....	2
三、研究目的.....	2
參、國內外相關研究（如文獻、產品說明）	3
肆、研究目標.....	5
一、系統使用對象.....	5
二、系統功能說明介紹	5
三、適用時機與方式.....	6
伍、研究方法與進行步驟.....	6
陸、系統具體成果.....	12
柒、結論與未來展望.....	25
捌、成員心得.....	29
玖、參考文獻.....	32

壹、摘要

此份專題報告書旨在說明我們的專題與林口長庚紀念醫院早期療育中心合作之成果及心得。首先，我們將簡略說明此次專題與早期療育中心合作的目標，本專題旨在協助林口長庚紀念醫院早期療育中心開發一套以 Android App 為載體的精細動作能力評估系統，用於早期偵測兒童發育遲緩情況。我們的目標是透過電子化測驗流程與標準化評分機制，降低評估差異，並進一步建立年齡段常模，提供客觀的評分以降低相關醫療人員之間的內部評估差異，為臨床研究與醫療人員提供可靠參考依據。我們依據這個主題與陳嘉玲醫師及陳姍好學姐共同討論並安排以下幾個部分的工作內容：

1. 閱讀相關文獻，設計一套可適用於手機上的電子化測驗流程
2. 選取合適的測驗題目予以實作，並同時設計可用於程式的分數評估方式
3. 討論其他需要實作於 App 上的功能，同時進行資料庫系統的開發與建置
4. 將整個系統整合，確認 App 能否實際相容於多數較新版本的 Android 手機
5. 基於現有的模型進行優化，包括預處理、資料集擴充等，提升分數評估的精確度
6. 聚焦收案時的使用者回饋，及時調整以優化使用者體驗
7. 與醫師和學姊定期討論，進行系統功能、資料庫等方面的新增與調整

在合作期間，我們每個月會挑大家都有空的時間，向醫師、教授、學姐及其他合作人員報告進度並進行討論，其餘的時間就各自進行小組討論與實作，若有設計上的相關問題亦可隨時向學姐討論預計在暑假結束時能夠將 App 的功能大致完成，待開學後學姐便能實際到國小進行收案，確認該 App 的評估準確性，並利用其資料庫系統根據不同年齡段的受試者對於該測驗的分數差異性來建立常模，以提供醫療人員更進一步的參考資訊，以進行臨床上之研究，然而目前因專家效度還在進行中，因此收案部分會延到寒假來進行，在此期間也會持續的完善系統功能。

這份報告中也將呈現我們所學的專業知識及技術，文中會說明工作中主要所使用的程式語言及開發平台，並挑選合適的 API 及 Library (TensorFlow、Keras、OpenCV、ML Kit、Firebase、Jetpack DataStore 等...) 來進行 App 的系統開發，我們會利用 Anaconda Prompt 建置虛擬環境，並以 Visual Studio Code 作為主要的程式編輯器，我們會利用 Python 的各種 Package 來進行模型設計與訓練、資料處理、圖像處理，並利用 OpenCV 進行電腦視覺的相關操作，因為最終主要載體為 Android 手機，故會使用 Android Studio 進行 App 開發與設計，以 Java 作為主要程式語言，結合前述所提及的 API 來完成我們的系統。

這份報告亦會記錄我們工作的各項資料文件，包括我們所規劃的電子化設計流程、參考文獻、系統現況以及一些在開發時會遇到的問題統整。最後，附上這幾個月以來的合作心得，期盼這些內容能夠對於未來進行軟硬體專題的同學們有所幫助！

貳、背景、動機與目的

一、研究背景

引起兒童在動作能力發展上遲緩的原因有很多，其中包括環境因素、社會文化因素、心理因素及腦神經、肌肉系統疾病等…，動作能力大致上可分成兩種類型，精細及粗大動作能力，由於精細動作能力的觀察，較不像一般的粗大動作能力容易以肉眼直接辨識，導致大多數的家長對兒童的精細動作能力發展遲緩現象缺乏警覺性，大多是等到入學後與其他孩子相比才發現，錯失及早治療的機會。以下定義兩者動作能力之間的差異：

※ 精細動作能力	※ 粗大動作能力
◎ 使用細小肌肉或肌肉群（手部小肌肉）收縮與控制的動作	◎ 需要大塊肌肉或肌肉群（軀幹與四肢肌肉）收縮與控制的動作
◎ 手部的活動：抓握、捏、放，手眼協調以及工具操作	◎ 維持姿勢的能力：躺、坐、跪、站、蹲 ◎ 改變姿勢的能力：坐到站、站到蹲 ◎ 移動能力：爬、走、跑、跳、投擲

大腦在嬰幼兒期（從出生到六歲之間）發育速度最快、可塑性也較佳；另一方面，神經肌肉系統隨著年齡增長及與環境間的互動，也會增進身體及心智各方面的成熟度。故對於發展遲緩兒童來說，若能透過早期療育的介入，我們便可以提早發現並及時治療，把握發展遲緩兒童的黃金治療期，即能有效避免將來不可逆的發展障礙。

二、研究動機

有鑑於目前早期療育大多採取人工評估為主，陳嘉玲醫師便提案開發一套兒童精細動作評估系統，在App上設計一系列電子化測驗，並藉由電腦視覺方法及AI模型來計算出相對客觀的分數，協助醫療人員來初步評估兒童是否有發育遲緩的現象，並配合醫療人員的指引來協助治療，以幫助他們能夠逐漸康復。雖然本系統在短期內實在難以達到全面，但我們期望能夠達到易於理解且便於操作，以及具備較良好的擴充性，倘若以後需要更動系統上的內容（如測驗題目、分數評估方式、模型、資料庫、App介面與功能等…），能在本系統的基礎上更容易的實作。

三、研究目的

在這次合作中，我們負責的是兒童精細動作評估系統開發，以Android App為載體，供早期療育人員進行收案以實際驗證成效。前期的工作內容主要是查詢測驗、發展量表並閱讀相關文獻，討論出精細動作評估中較適合用於電子化測驗的題目，並研究如何利用AI模型及電腦視覺方法來協助我們進行分數上的評估，以及這種方式是否能與人工評估的方式產生

一致性，並討論實作上的可行性。後期便開始結合先前所規劃的內容，著手進行 App 的開發，包括介面及系統設計，且由於 Android 手機的多樣性，我們亦要處理版面配置、權限及相容性等問題。最後便將 App 與資料庫系統進行整合，即能供早期療育人員實際進行收案以協助他們在臨床上之研究。

參、 國內外相關研究（如文獻、產品說明）

由於我們缺乏相關的醫療背景知識，因此我們在合作期間閱讀了一些相關文獻，其主要內容大致圍繞在兒童精細動作的發展里程碑（Developmental Milestones）[1]，評估兒童精細動作能力的相關測驗，以及如何利用 AI 模型及電腦視覺方法評估分數等...。

首先是精細動作發展里程碑，對於已經長大的我們來說，使用雙手進行精細動作是十分容易的，但對於學齡前的嬰幼兒來說，其實在每個階段都有許多不同的指標來評估他們是否發育正常，因此我們在設計電子化測驗的內容時，應當考慮題目在 App 上操作的難易度，以適應不同成長階段的兒童，且由於在 App 上操作的侷限性，我們只能挑選如仿畫、臨摹等精細動作測驗來進行設計，而不是如堆疊積木、使用餐具等無法在 App 上進行操作的測驗內容（這部分在未來可能會搭配穿戴式裝置進行評估）。

Age	Developmental Milestones	Possible implications if milestones not achieved
0 – 6 months	<ul style="list-style-type: none"> • Reflexive grasp (at birth) • Global ineffective reach for objects (3 months) • Voluntary grasp (3 months) • 2 handed palmar grasp (3 months) • 1 handed palmar grasp (5 months) • Controlled reach (6 months) 	<ul style="list-style-type: none"> • Poor muscle development and control • Delayed ability to play independently • Delayed sensory development due to delayed interaction with toys and other sensory objects
6 – 12 months	<ul style="list-style-type: none"> • Reaches, grasps, puts object in mouth • Controlled release of objects • Static Pincer grasp (thumb and one finger) • Picks things up with pincer grasp (thumb and one finger) • Transfers objects from one hand to another • Drops and picks up toys 	<ul style="list-style-type: none"> • Poor development of hand and finger strength • Poor manipulation of objects resulting in delayed play skills • Delayed sensory development due to lack of sensory play experiences
1-2 years	<ul style="list-style-type: none"> • Builds tower of three small blocks • Puts four rings on stick • Places five pegs in pegboard • Turns pages two or three of a book at a time • Scribbles • Turns knobs • Paints with whole arm movement, shifts hands, makes strokes • Self-feeds with minimal assistance • Able to use signing to communicate • Brings spoon to mouth • Holds and drinks from cup independently 	<ul style="list-style-type: none"> • Poor development of hand and finger strength • Delayed independent play skills • Delayed development of self care skills (such as eating) • Delayed manipulation skills
2-3 years	<ul style="list-style-type: none"> • Strings four large beads • Turns single pages of a book • Snips with scissors • Holds crayon with thumb and fingers (not fist) • Uses one hand consistently in most activities • Imitates circular, vertical, and horizontal strokes • Paints with some wrist action, makes dots, lines, circular strokes • Rolls, pounds, squeezes, and pulls playdough • Eats without assistance 	<ul style="list-style-type: none"> • Delayed self-care skills (such as eating) • Delayed pre-writing skill development • Delayed manipulation of small objects such as toys, pencils and scissors • Frustration when manipulating small toys and objects
3 – 4 yrs	<ul style="list-style-type: none"> • Builds tower of nine small blocks • Copies circle • Imitates cross • Manipulates clay material (rolls balls, makes snakes, cookies) • Uses non-dominant hand to assist and stabilise the use of objects • Snips paper using scissors 	<ul style="list-style-type: none"> • Delayed pre-writing skill development • Frustration and/or avoidance of pencil based tasks • Poor pencil grasp and pencil control • Poor self-care skills (such as eating) • Delayed drawing skills

4 -5 yrs	<ul style="list-style-type: none"> Cuts on line continuously Copies cross Copies square Writes name Writes numbers 1-5 Copies letters Handedness is well established Dresses and undresses independently 	<ul style="list-style-type: none"> Difficulties holding and manipulating a pencil Difficulties learning to write name and other letters of the alphabet Dependence on caregivers for every day activities such as dressing Frustration and/or avoidance of pencil based tasks
5 – 6 yrs	<ul style="list-style-type: none"> Cuts out simple shapes Copies triangle Colours within lines Uses a 3 fingered grasp of pencil and uses fingers to generate movement Pastes and glues appropriately Can draw basic pictures 	<ul style="list-style-type: none"> Difficulties learning to form letters and numbers correctly Poor handwriting Difficulties demonstrating academic ability on paper Fatigue during pencil based tasks Frustration and/or avoidance of pencil based tasks
6 -7 yrs	<ul style="list-style-type: none"> Forms most letters and numbers correctly Writes consistently on the lines Demonstrates controlled pencil movement Good endurance for writing Can build Lego, K'nex and other blocks independently Ties shoelaces independently 	<ul style="list-style-type: none"> Difficulties getting ideas down on paper Experiences fatigue during handwriting tasks Difficulty keeping up in class due to slow handwriting speed Poor legibility of handwriting May impact on self-esteem when comparing work to peers Possible frustration and/or behavior difficulties due to avoidance of pencil based tasks
7- 8 yrs	<ul style="list-style-type: none"> Maintains legibility of handwriting for entirety of a story 	<ul style="list-style-type: none"> Difficulty completing handwriting tasks in a timely manner Experiencing fatigue during handwriting tasks Poor academic achievement due to difficulty getting ideas down on paper Difficulties due to avoidance of pencil based tasks

This chart was designed to serve as a functional screening of developmental skills per age group. It does not constitute an assessment nor reflect strictly standardised research.

The information in this chart was compiled over many years from a variety of sources. This information was then further shaped by years of clinical practice as well as therapeutic consultation with child care, pre-school and school teachers in South Australia about the developmental skills necessary for children to meet the demands of these educational environments. In more recent years, it has been further modified by the need for children and their teachers to meet the functional Australian Curriculum, Assessment and Reporting Authority (ACARA) requirements that are not always congruent with standardised research.

Figure 1. 兒童精細動作發展里程碑之表格

接著是評估兒童精細動作能力的相關測驗，根據早期療育中心提供給我們的相關資料，我們參考了許多市面上已經存在的發展量表，包括 Peabody Developmental Motor Scales | Second Edition (PDMS-2)、Bruininks-Oseretsky Test of Motor Proficiency | Second Edition (BOT-2)、Bayley Scales of Infant and Toddler Development | Third Edition (Bayley-III)、Beery-Buktenica Developmental Test of Visual-Motor Integration | Sixth Edition (Beery VMI) 等...。



Figure 2. PDMS-2、BOT-2、Bayley-III、Beery VMI 實際產品圖

根據這些市面上發展量表中有關於精細動作能力測驗的部分，我們和早期療育中心進行討論後，挑選出了三個較典型且適合用於電子化測驗的題目，分別是：

1. 圖形仿畫
2. 路徑內臨摹
3. 文字謄寫

我們針對三種類型的精細動作能力測驗，設計了不同的分數評估方式，根據 Ruimin Li

等人的研究[2]，使用帶有時序濾波器的 CNN 模型評估經攝影機錄製的精細動作是可行的，不同的是我們採用 CNN 模型進行圖形仿畫分數評估，現今的 CNN 模型在影像辨識上非常成熟，甚至可以做到比人類還要精準的程度，但這必須仰賴於龐大的訓練集，由於早期療育中心的圖形仿畫中僅有「多菱形」有 5584 筆仿畫資料，又因多菱形是我們自行設計的圖形，且評分細項較為複雜，故適合利用 AI 來協助分數評估。

路徑內臨摹是採用 IoU 的方式進行評估，不同於 Ruimin Li 等人的研究[2]，我們是設定筆觸能夠將路徑近乎填滿，並同時考量路徑內、外的比例來進行分數評估，此方式較為嚴格，但能夠更準確地反映出精細動作能力，因為所有筆觸的路徑偏移均會被路徑外的比例所記錄，並給予扣分。

文字謄寫的部分，我們參考了 Seth Polsley 等人的研究[3]，他們使用機器學習的方式來評估英文字的一些文本指標，如曲線、筆畫、勾、閉合度等…，來進行精細動作能力評估，不同的是我們希望能評估中文字，對臺灣的兒童來說會較為熟悉，有鑑於中文字的複雜度，我們決定以 OCR、文本指標（如筆畫）、圖形相似度等數個評估方法，將其進行加權統計，綜合來進行分數的計算，我們認為這樣能兼顧整體性及局部性的評估。

肆、 研究目標

一、 系統使用對象

本系統主要使用對象包括兒童及家長、早期療育人員及醫師等…。兒童是直接受益者，為主要參與測驗的對象，旨在評估他們的精細動作能力，家長與早期療育人員可從旁協助測試，如發現成績不盡理想，便可尋求醫師或其他專業人員提供協助，確認問題點並提供醫療建議、擬定治療計畫等，協助兒童動作能力的恢復。我們也會根據個人資料中慣用手的設定來調整精細動作測驗的版面，使得左慣用手的兒童也能便於進行測驗。

二、 系統功能說明介紹

我們製作的兒童精細動作評估系統，主要可應用於早期療育人員進行小規模收案（如班級、社團等…）。本系統除了精細動作能力測驗外，亦提供粗大動作能力測驗錄製、個人基本資料填寫功能，系統除了會將資料保存至本機端，也會將資料上傳至我們連接的 Firebase 資料庫，以利早期療育中心進行分析，並根據不同年齡段受試者的分數差異性來建立常模，以利於後續進行臨床上之研究。

本系統不僅可用於收案，亦適合任何人進行個測，受試者可利用支援本系統的手機進行測驗，評估自身的精細動作能力（家長可從旁協助），若發現結果不如預期，便可尋求相關醫療人員提供協助，配合醫師的指引來協助治療，以幫助他們能夠逐漸康復，本系統則在旁擔任輔助的角色。

三、 適用時機與方式

本系統適用於一般目的性的收案流程，抑或是單純測試兒童的精細動作能力，倘若某受試者的精細動作能力發展遲緩現象已被發覺，則便是該系統適當之使用時機，然而本系統使用的方式主要是以 App 來進行操作，仍有許多可能（如不習慣手機操作、注意力不集中等...）造成測驗分數不理想，因此若旁邊能有早期療育人員或醫師來協助評估，才是最妥善的使用時機與方式，因此在定義上，本系統主要還是扮演一輔助的角色，減輕傳統上人工評估的不便之處。

伍、 研究方法與進行步驟

接下來便是如何利用 AI 模型及電腦視覺相關方法評估分數，根據我們所閱讀的相關文獻，使用 Convolutional Neural Network (CNN) 模型來進行評估是一個可行的方案，現今的 CNN 在影像辨識上非常成熟，甚至可以做到比人類還要精準的程度，但這必須仰賴於龐大的訓練集。由於電子化測驗的實際收案樣本數不夠多，因此早期療育中心提議讓我們以仿畫多菱形當作主要訓練圖形，該圖形是由他們所自行設計的，目前尚未出現在其他測驗中，故我們以這個仿畫圖形來進行訓練，既可以測試它對於精細動作評估的效度是否足夠，又因為它是較複雜的仿畫圖形（18 個評分細項），在人工評估上也較容易產生誤差，因此利用 CNN 模型來協助醫療人員降低內部評估差異便顯得有價值，至於其他的仿畫圖形，待訓練集足夠多時，便能利用相似的方法來進行評分。

如前所述，根據早期療育中心的提議，我們將以多菱形仿畫圖形作為 CNN 模型的訓練對象，該圖形共有 6 個評分項目（基本圖形、大小、邊、方向、重疊、開合度），總共 18 個細項，每個細項符合標準得 1 分，不符合便得 0 分。其中除了方向是考慮直、橫，重疊是考慮上、下、左、右，其餘 4 個評分項目均是考慮外、中、內三個菱形，方向及重疊的評分方式如下所示：

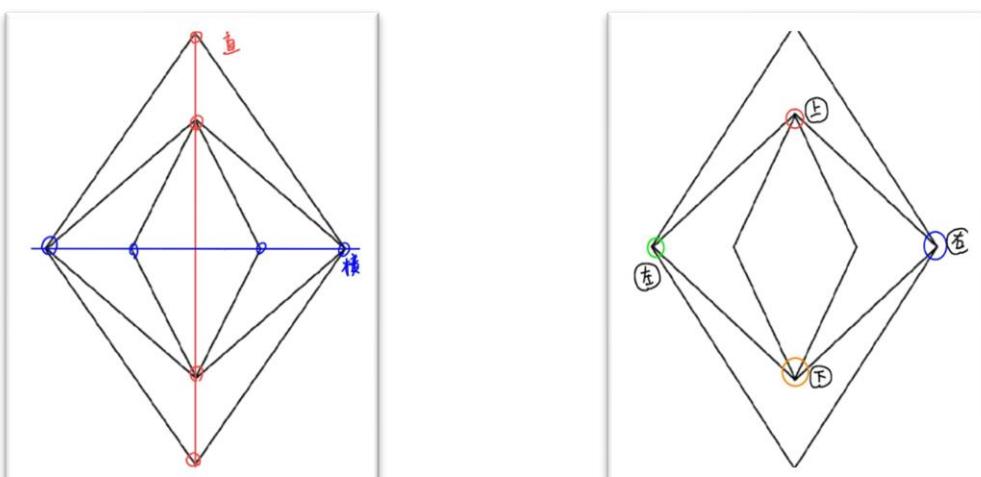


Figure 3. 多菱形仿畫圖形的方向及重疊評分方式

早期療育中心提供給我們共 4301 筆不同樣式的多菱形仿畫圖形，而我們自己添加了 1283 筆與多菱形仿畫圖形幾乎完全不同的干擾資料（Noising），共 5584 筆資料當作訓練集，提供給我們的 CNN 模型進行訓練。首先對仿畫圖形進行 Data Preprocessing，主要分成以下四個步驟：

處理方式	說明
大小縮放	將圖形縮放至某一固定大小（目前的尺寸為：328 * 232）
灰階轉換	將圖形的 Channel 數設為 1，轉換成為 0~255 的灰階圖形
影像增強	利用 Gaussian Blur 對圖形進行平滑處理，並利用 Laplacian Operator 進行邊緣檢測，以進行圖形銳化
統一格式及正規化	利用 MinMaxScaler 將讀取到的原始圖形數據正規化 ($[0,255] \rightarrow [0,1]$)

多菱形仿畫圖形處理完畢後，接著我們利用 Pandas 來進行資料處理，根據早期療育中心提供給我們的仿畫圖形與評分結果對應之 Excel 檔，讀取其資料並儲存至記憶體，待模型建立好後，便準備送入 CNN 模型進行訓練。

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	基本圖形(外)	基本圖形(中)	基本圖形(內)	大小(外)	大小(中)	大小(內)	邊(外)	邊(中)	邊(內)	方向(直)	方向(彎)	重疊(上)	重疊(下)	重疊(左)	重疊(右)	組合度(外)	組合度(中)	組合度(內)
1	檔案名稱																	
2	Data (1)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	Data (2)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	Data (3)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	Data (4)	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1
6	Data (5)	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1
7	Data (6)	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1
8	Data (7)	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1
9	Data (8)	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	1
10	Data (9)	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1
11	Data (10)	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1
12	Data (11)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	Data (12)	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
14	Data (13)	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	1	1
15	Data (14)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	Data (15)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	Data (16)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	Data (17)	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1
19	Data (18)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	Data (19)	1	1	1	1	0	0	0	0	0	0	1	1	0	0	1	1	1
	Data (20)	1	1	1	1	0	0	0	0	0	0	1	1	0	0	1	1	1

Figure 4. 多菱形仿畫圖形與評分結果對應之 Excel 表格節錄

接著是我們 CNN 模型的架構圖，如下所示：

/content/Project: 8		
Model: "sequential_2"		
Layer (type)	Output Shape	Param #
conv2d_14 (Conv2D)	(None, 232, 328, 8)	80
conv2d_15 (Conv2D)	(None, 232, 328, 16)	1,168
conv2d_16 (Conv2D)	(None, 230, 326, 16)	2,320
max_pooling2d_6 (MaxPooling2D)	(None, 115, 163, 16)	0
dropout_8 (Dropout)	(None, 115, 163, 16)	0
conv2d_17 (Conv2D)	(None, 115, 163, 24)	3,480
conv2d_18 (Conv2D)	(None, 113, 161, 24)	5,208
max_pooling2d_7 (MaxPooling2D)	(None, 56, 80, 24)	0
dropout_9 (Dropout)	(None, 56, 80, 24)	0
conv2d_19 (Conv2D)	(None, 56, 80, 32)	19,232
conv2d_20 (Conv2D)	(None, 52, 76, 32)	25,632
max_pooling2d_8 (MaxPooling2D)	(None, 13, 19, 32)	0
dropout_10 (Dropout)	(None, 13, 19, 32)	0
flatten_2 (Flatten)	(None, 7904)	0
dense_4 (Dense)	(None, 1024)	8,094,720
dropout_11 (Dropout)	(None, 1024)	0
dense_5 (Dense)	(None, 18)	18,450
Total params: 8,170,290 (31.17 MB)		
Trainable params: 8,170,290 (31.17 MB)		
Non-trainable params: 0 (0.00 B)		
continue		

Figure 5. CNN 模型建立完成後的 Summary

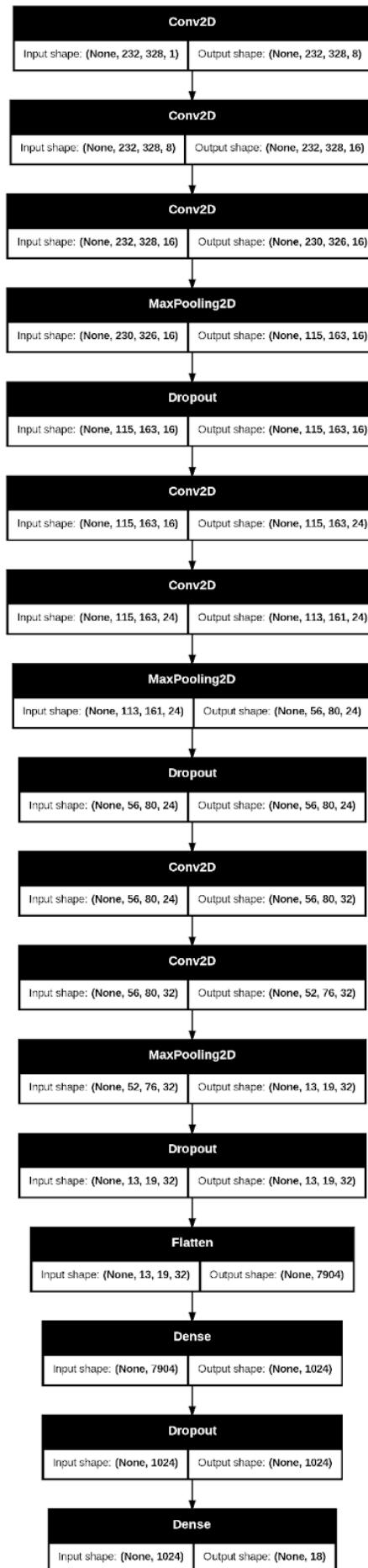


Figure 6. CNN 模型架構圖

基本上我們的 CNN 模型和一般常見的 CNN 最大的不同在於 Output Layer 的 Activation Function 並不是採用傳統的 Softmax Function，而是採用 Sigmoid Function，原因在 Softmax 適合用於基於機率的多分類問題，通常用於影像分類上，而我們的電子化測驗中，需要的是評估 18 個細項的得分與否，此時 Sigmoid 的特性便十分適合用來進行評估，我們會設定一合適的閾值 τ ，目前是設定 $\tau = 0.5$ ，若 τ 小於 0.5 則該細項便得 0 分，否則該細項便得 1 分。

以下則是我們 CNN 模型的超參數細項：

模型重要參數	設定值
Activation Function	ReLU
Activation Function (Output Layer)	Sigmoid
Learning Rate	0.002
Dropout	0.5
Loss Function	Binary Crossentropy
Optimizers	NAdam
Metrics	Binary Accuracy

比較值得注意的是，我們的 Loss Function 採用的是 Binary Crossentropy，一般常用於二分類問題，對於我們 Output Layer 的規劃來說將會十分合適，畢竟我們期望每個細項的輸出不是 0 (錯誤) 即是 1 (正確)，因此要盡可能地減少 Sigmoid Function 輸出靠近 0.5 這種模棱兩可的數值。

接著是我們的訓練結果，如下所示：

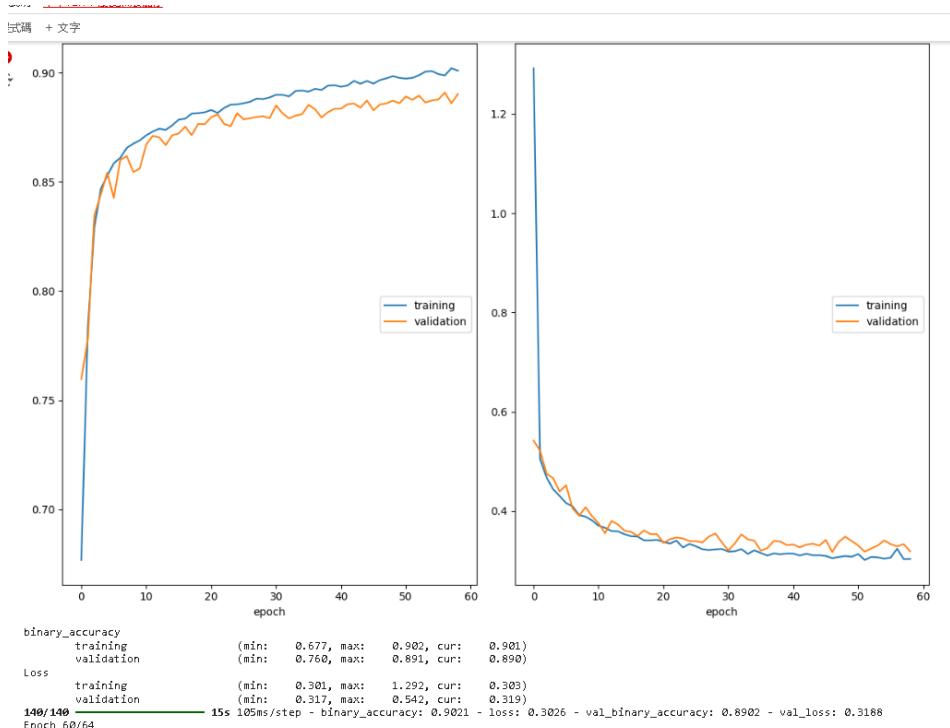


Figure 7. CNN 模型訓練結果（訓練集及驗證集之 Accuracy 及 Loss）

我們共訓練了 100 個 epoches，而 Batch Size 我們設定成 32，在每批次的訓練中，我們也利用 Keras 的 ImageDataGenerator 來進行 Data Augmentation，以及利用 Keras 的 ReduceLROnPlateau 來進行 Adaptive Learning Rates，如下圖所示：

```
from keras.preprocessing.image import ImageDataGenerator
gen = ImageDataGenerator(featurewise_center = False,
                         samplewise_center = False,
                         width_shift_range = 0.08,
                         shear_range = 0.3,
                         height_shift_range = 0.08,
                         zoom_range = 0.08,
                         data_format = "channels_last")
train_generator = gen.flow(x_train, y_train, batch_size = 3)
```

Figure 8. 利用 ImageDataGenerator 進行 Data Augmentation

```
from keras.callbacks import ReduceLROnPlateau
learning_rate_function = ReduceLROnPlateau(monitor = 'val_acc', patience = 3, verbose = 1, factor = 0.5, min_lr = 0.00001)
```

Figure 9. 利用 ReduceLROnPlateau 進行 Adaptive Learning Rates

以上這些措施能夠有效防止我們的模型發生 Overfitting 的現象（並且我們也逐步的簡化 CNN 模型架構，將參數的數量調降的同時，同時保證其準確率），我們亦有嘗試以下方法來嘗試解決 Overfitting 的問題，但效果較不顯著，如 Batch Normalization、L1 Regularization、L2 Regularization、將 CNN 模型拆分成 6 個子集合，各自獨立進行訓練等…。

總體而言，我們針對多菱形仿畫圖形所訓練的 CNN 模型，準確率大致上可以達到約八成左右，泛化程度還算不錯，而且並沒有嚴重的 Overfitting 問題，不過我們在進行實際測試時，發現該模型對於某些未曾出現在訓練集的資料，也就是與訓練集差異較大的仿畫圖形，仍可能發生誤判的情形，這部分只能仰賴之後逐漸擴增資料集的規模，或是再針對模型進行修改來進行改善了。

接著是路徑內臨摹，這個測驗的評分方式主要是看受試者能否將畫筆盡可能的描繪在圖形的路徑內部，描繪的越精準分數便會越高，我們與早期療育中心討論後，會先將該測驗圖形繪製一張標準圖形，接著便以受試者所繪製的測驗圖以及標準圖進行 Intersection over Union (IOU) (又稱為 Jaccard Index) 的計算，假設標準圖的畫筆筆觸範圍（黑色部分）記作 A，測驗圖的畫筆筆觸範圍記作 B，則計算公式如下所示：

$$J_i(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

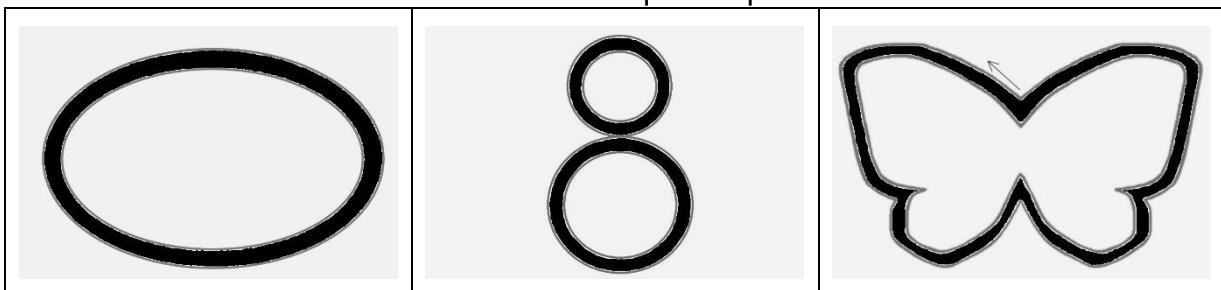


Figure 10. 電子化測驗—路徑內臨摹題目之標準圖形

根據公式的定義， $J_i(A, B)$ 即為畫筆落在圖形的路徑內部之比值，然而根據其他發展量表，我們也必須考量畫筆落在路徑外部的情況，否則受試者亦有可能將整張畫布塗滿且仍然得到滿分，假設整個標準圖定義成字集合記作 U ，則計算公式如下所示：

$$J_o(A, B) = \frac{|B - J_i(A, B)|}{|U - A|}$$

根據公式的定義， $J_o(A, B)$ 即為畫筆落在圖形的路徑外部之比值，根據 Jaccard Index 的定義，可得到 $0 \leq J_i(A, B) \leq 1$ ，且 $0 \leq J_o(A, B) \leq 1$ ， $J_i(A, B)$ 的值越大，受試者的分數越高，同理， $J_o(A, B)$ 的分數越小，受試者的分數越高。然而早期療育中心希望我們能夠將這兩個 Index 合併成一個更好量化的分數，使他們在評估上更加方便，因此我們和教授進行討論，設計出將其合併後的評估指標，計算公式如下所示：

$$I(A, B) = \frac{J_i(A, B)}{1 + \alpha \times J_o(A, B)}$$

其中 $\alpha \in R^+$ ，為一個可調節的權重值，當 $\alpha < 1$ 時，表示評估指標較偏向受試者要將畫筆盡可能地落在圖形路徑內，當 $\alpha > 1$ 時，表示評估指標較偏向受試者要盡可能避免將畫筆落在圖形路徑外。該指標還有一個優點，即它是非線性的，就是隨著 $J_o(A, B)$ 的值越大， $I(A, B)$ 的值下降的速度會趨緩，這很符合臨床上的認知，當受試者可能有輕微的精細動作發展遲緩時， $J_o(A, B)$ 的值會從 0 開始逐漸上升，指標必須要對上升幅度足夠敏感，以反映出分數上的差異，然而當 $J_o(A, B)$ 大於某個閾值時，代表受試者可能是亂畫或真的有精細動作發展遲緩，才難以控制畫筆的範圍，此時指標便不需變化太大，畢竟已經足以反映出問題了。

至於文字謄寫的部分我們和早期療育中心尚未討論出最合適的評估分數方法，雖然現今 Optical Character Recognition (OCR) 已非常成熟，但我們希望能夠辨別受試者寫的字和標準字究竟有多少差異，而不是僅單單辨識寫的字為何，因此 OCR 並不完全適用於我們，然而 OCR 在一定程度上仍可提供最大概率文字，因此我們可能會在一開始將 OCR 辨識的字與標準字比對，給予初步的分數。CNN 模型雖然適合用於影像辨識，但我們並未討論出合適的方法，判斷受試者寫的字要和標準字差多少才不被模型所接受（如筆畫、鉤、撇、捺等...），由於缺少訓練集的答案部分，CNN 模型也無法用於此測驗，我們目前嘗試採用 Histograms of Oriented Gradients (HOG) 等...比較相似度的方式進行評估。可能之評分流程如下所示：

- Step 1. 利用 OCR 比對測驗字，若與標準字比對結果相同則給予一基本分數 C_1
- Step 2. 計算測驗字的筆畫數目，根據該筆畫數目與標準字的差值多寡給予一分數 C_2
- Step 3. 使用 HOG 等相似度算法，比對測驗字及標準字給予一分數 C_3
- Step 4. 設定適合的權重值 α 、 β 、 γ ，將上述基本分數進行加權相加以獲得最終分數 S

陸、 系統具體成果

將前述所提及的系統內容開發成實際的 Android App，會遭遇到許多的問題，以下將主要針對介面、系統、權限、相容性四個部分來闡述我們開發 App 的過程，以及在這之中所遭遇到的一些問題。

首先是介面及系統設計，Android 是基於 Linux Kernel 的 Open Source 行動作業系統，在現今 Android 是最被廣泛使用的智慧型手機作業系統，基於它的開放性，市面上有非常多基於 Android 系統的智慧型手機，其系統版本、螢幕尺寸、處理器、相機模組、非原生系統軟體等...，這些軟體及硬體規格上的差異性，使得在開發 Android App 時必須了解軟體的生命週期，以及要面向的目標客群，如此才能夠撰寫出符合功能需求且具備系統穩定性的應用程式。

因此我們在 build.gradle 中設定 minSdkVersion 21、targetSdkVersion 33，代表支援 Android 5~Android 13 的智慧型手機，Android 5 大約於 2014 年問世，我們認為以智慧型手機的生命週期，已經足以支援絕大多數人的設備了。我們的介面主要有註冊及登入頁面、主頁面、標題列、精細動作測驗頁面、粗大動作測驗頁面、個人資料設定頁面、系統設置頁面等...，如下圖所示：

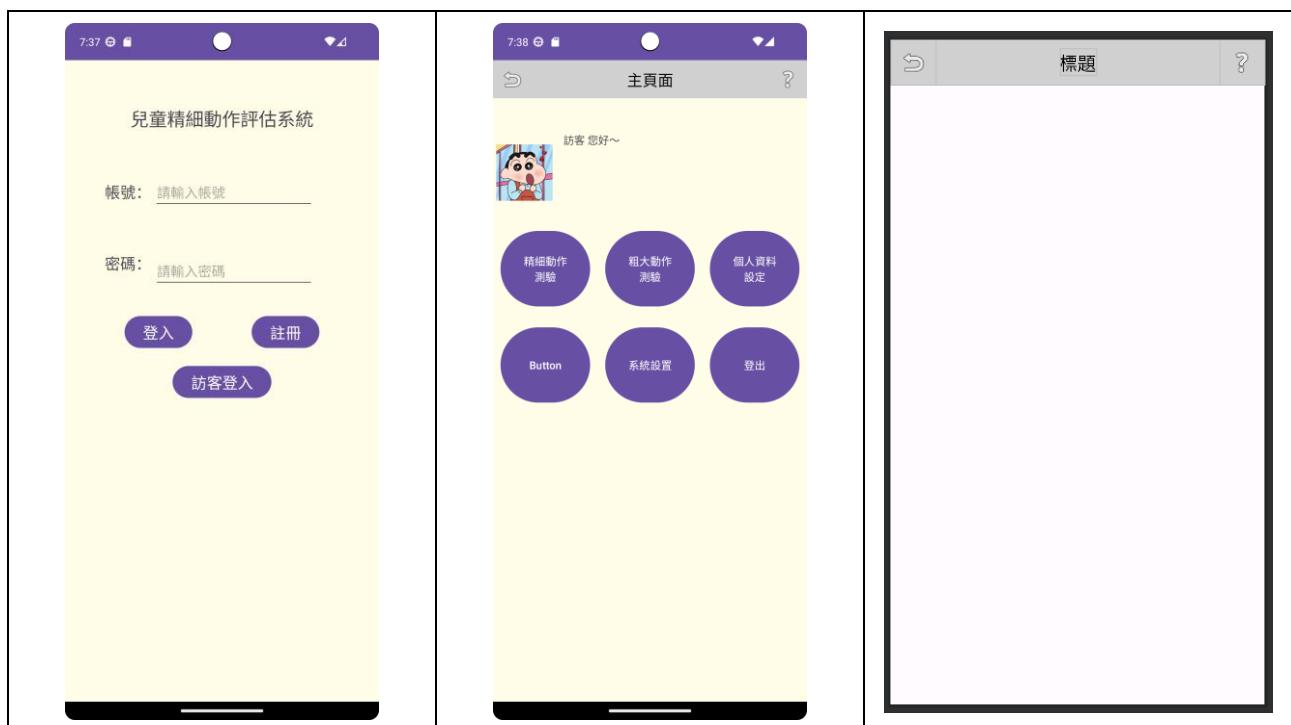


Figure 11. 註冊及登入頁面、主頁面、標題列

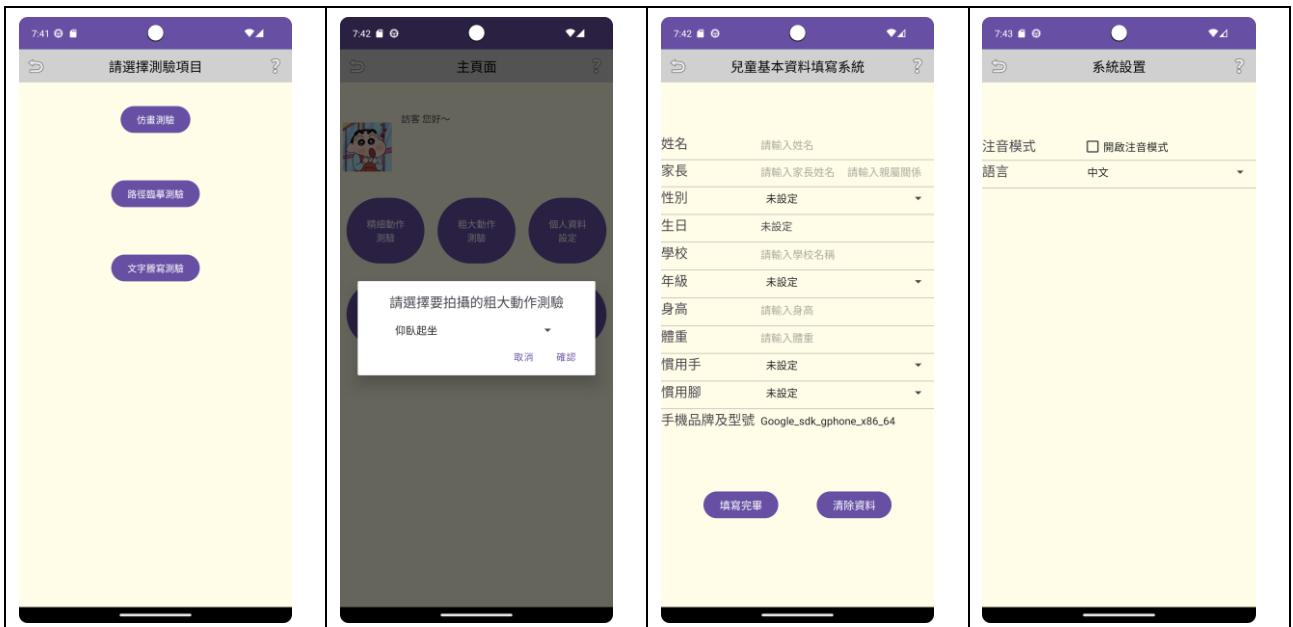


Figure 12. 精細動作測驗頁面、粗大動作測驗視窗、個人資料設定頁面、系統設置頁面

在精細動作測驗頁面中，個別測驗的頁面組成大致相同，在測驗開始時會顯示該測驗的提示詞，在測驗結束時則會顯示如分數等的相關資訊，如下圖所示：

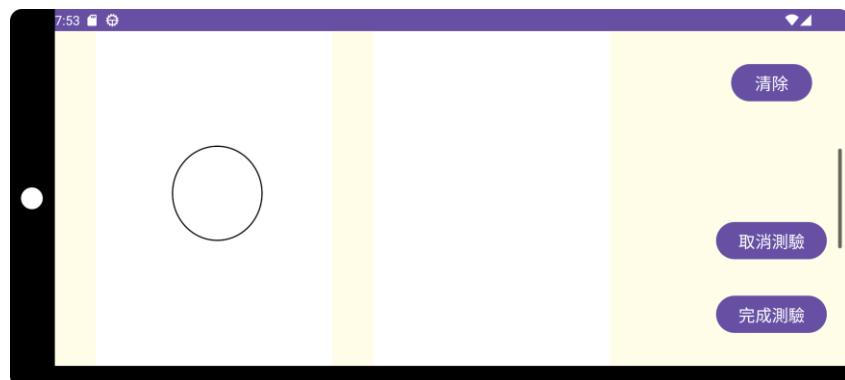


Figure 13. 圖形仿畫測驗頁面

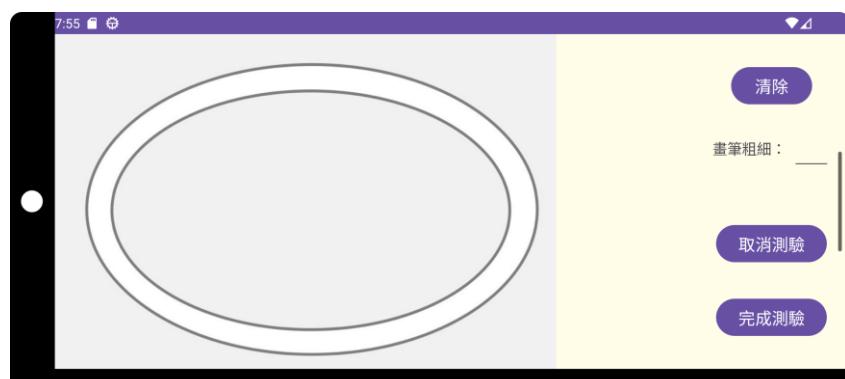


Figure 14. 路徑內臨摹測驗頁面

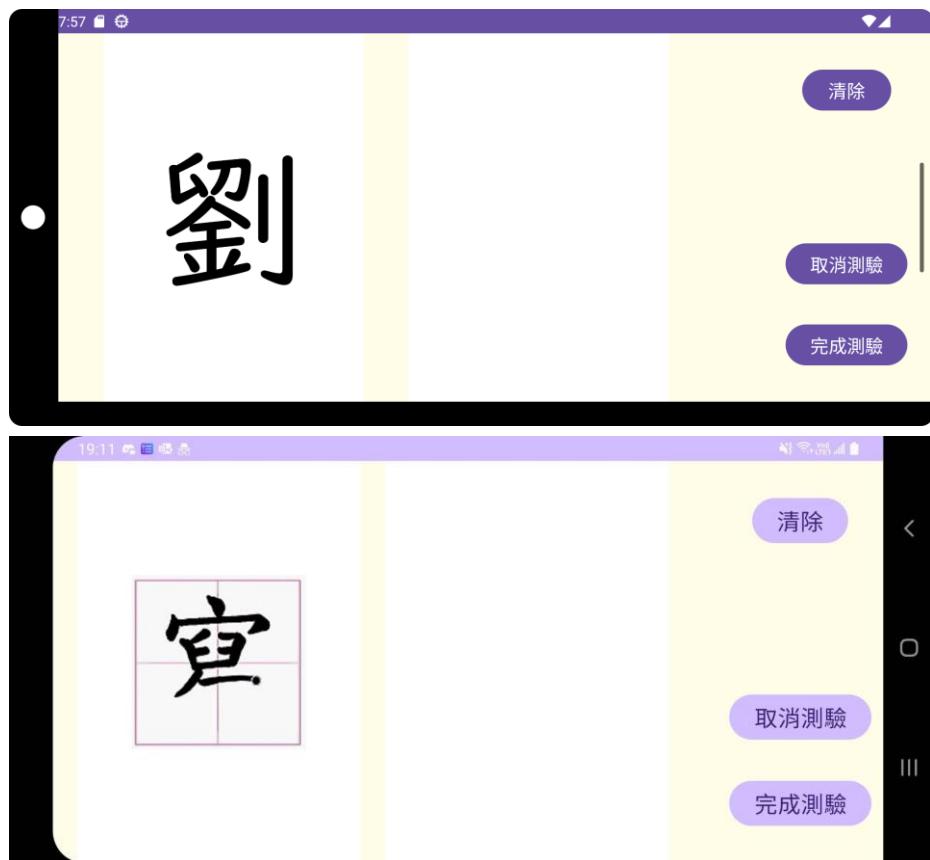


Figure 15. 文字謄寫測驗頁面(第二張為 gif 顯示)



Figure 16. 蝴蝶圖路徑內臨摹測驗提示詞



Figure 17. 路徑內臨摹測驗評估結果

我們在設計精細動作測驗頁面時，有考量慣用手的問題，因此在受試者尚未設定基本資料時，由於右撇子的人口基數較大，將一律以右手為慣用手來設計介面，倘若受試者設定成左撇子，則測驗介面的標準圖與測驗圖（即可供仿畫 Canvas 空間）位置將會進行對調，如下圖所示：

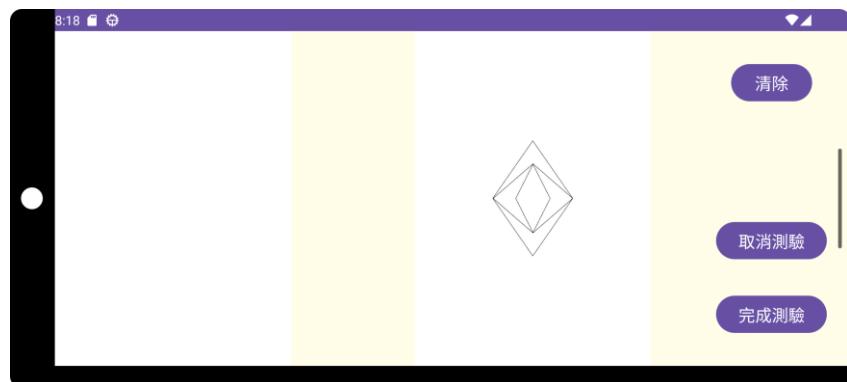


Figure 18. 圖形仿畫測驗頁面（左撇子）

在設計這些頁面時，我們盡可能採用 Android Studio 的導線（GuildLine）進行設置，該布局元件並不會顯示在頁面上，但基於目前主流的 Constraint Layout，讓我們頁面中的元件對齊 Parent 以及導線，能夠使畫面的布局元件始終保持在相對的位置上，如此便能支援不同螢幕尺寸的手機，且不會造成嚴重的跑版問題。在個人資料設定頁面中，因為顯示的資訊比較多，可能導致無法完全容納於頁面，我們採用 scrollView 以及 Linear Layout 來進行設計，讓受試者可以上下滑動並填寫各個欄位，以解決版面空間太小的問題。



Figure 19. 基於導線並採用 scrollView 來設計的個人資料設定頁面

接著便是精細動作測驗頁面中的圖檔大小問題了，由於每隻手機的螢幕尺寸和解析度均不相同，導致 dpi 的大小不一，因此我們會於程式中讀取當前手機的介面大小，來對原始圖片進行一定程度的壓縮，使其能夠正常顯示於 App 中，同時會記錄壓縮的比例，以供早期療育相關人員評估不同手機對於測驗上是否有一定程度的影響。

```

3 usages
public Bitmap ImageCompress(Bitmap oldBitmap, int Limit_w, int Limit_h) {
    int srcWidth = oldBitmap.getWidth();
    int srcHeight = oldBitmap.getHeight();
    float heightScale = ((float) Limit_h) / srcHeight;
    float widthScale = heightScale;
    Matrix matrix = new Matrix();
    matrix.postScale(widthScale, heightScale, px 0, py 0);
    Bitmap newBitmap = Bitmap.createBitmap(Math.round(srcWidth * widthScale), Math.round(srcHeight * heightScale), Bitmap.Config.ARGB_8888);
    Canvas canvas = new Canvas(newBitmap);
    Paint paint = new Paint();
    canvas.drawBitmap(oldBitmap, matrix, paint);
    return newBitmap;
}

@Override
protected void onSizeChanged(int w, int h, int oldw, int oldh) {
    super.onSizeChanged(w, h, oldw, oldh);

    // Initialize Empty Canvas
    myBitmap = ImageCompress(myBackground, getWidth(), getHeight());
    myCanvas = new Canvas(myBitmap);
}

```

Figure 20. 進行圖片壓縮的程式碼片段

由於在進行測驗時，根據受試者的手機解析度不同，會導致我們 Canvas 的畫筆粗細不同，因此我們會根據圖片壓縮的比例來調整畫筆的粗細（這在路徑內臨摹上會顯得特別重要），然而這個方法不一定會足夠準確，因此在路徑內臨摹測驗頁面中，我們允許受試者自行調整畫筆粗細，以適配測驗的畫筆及路徑寬度（兩者約相等）。

由於早期療育中心的同仁在進行收案時，同時也會進行粗大動作測驗，為了收案的方便

性，我們的 App 也允許收案人員錄製粗大動作測驗的內容，也會替他們將該內容在格式統一後保存至資料庫系統中，並限制錄製的秒數在 10 秒（最長時間限制）。

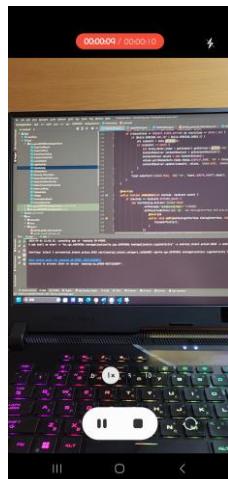


Figure 21. 實際用 Android 手機在 App 中錄製影片（基於系統內建的相機）

因為我們在進行測驗時會發生許多頁面跳轉的情形，在 Android App 程式設計中，每個頁面基本上會對應一個 Activity，這些 Activities 即是提供畫面的一個元件，然而一般進行頁面跳轉時，Android 系統會預設將跳轉前的 Activity 儲存至 Stack，如此一來當受試者按下返回鍵時便可回到跳轉前的 Activity，但對於我們的 App 來說，當受試者進行測驗時，這種行為是不被允許的，否則便可能重複進行相同測驗。

因此，我們採取的作法是禁用返回鍵，當受試者按下返回鍵時會提示受試者是否關閉 App，而不是回到跳轉前的頁面，這個行為是被定義在標題列左上角的返回按鍵，如此一來我們便能很好地掌握受試者的操作情形，避免非預期的頁面跳轉，同時我們也不需要利用 Stack 來儲存 Activities，而是當受試者按下標題列的返回按鍵時再跳轉至相對應的頁面即可，這種做法能夠有效的節省系統資源，也能夠使 App 的穩定性提高。

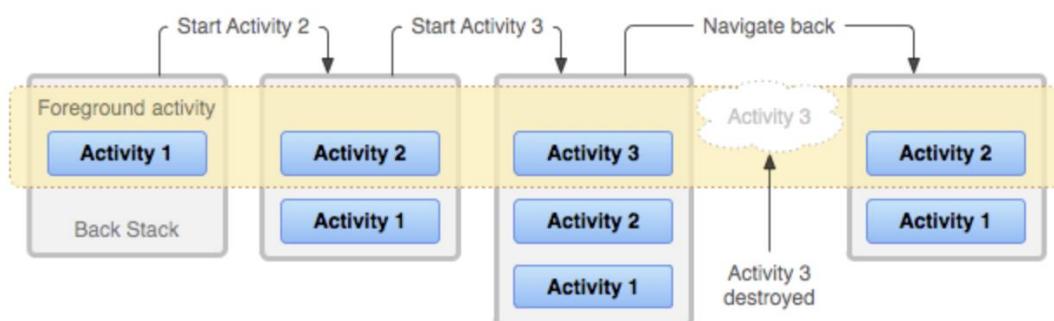


Figure 22. Android 系統預設的 Launch Mode (Standard)

```

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK) {
        new AlertDialog.Builder(context: this)
            .setMessage("是否要結束應用程式？") Builder
            .setPositiveButton( text: "是", (dialogInterface, i) -> finishAffinity())
            .setNegativeButton( text: "否", listener: null)
            .create() AlertDialog
            .show();
    }
    return super.onKeyDown(keyCode, event);
}

```

Figure 23. 禁用返回鍵的程式碼片段

```

public class TitleBar extends ConstraintLayout {
    1 usage
    ConstraintLayout Title_Bar;
    1 usage
    ImageButton BackBtn, HelpBtn;
    2 usages
    TextView TitleTxv;
    public TitleBar(@NonNull Context context, AttributeSet attrs) {
        super(context, attrs);
        LayoutInflater.from(context).inflate(R.layout.title_bar, root: this, attachToRoot: true);
        Title_Bar = (ConstraintLayout) findViewById(R.id.title_bar);
        BackBtn = (ImageButton) findViewById(R.id.Bar_BackButton);
        HelpBtn = (ImageButton) findViewById(R.id.Bar_HelpButton);
        TitleTxv = (TextView) findViewById(R.id.Bar_TitleTextView);
    }

    public void setTitle(String Title) { TitleTxv.setText(Title); }
}

```

Figure 24. 標題列元件的基本程式碼片段

接著是權限及相容性的問題，這部分主要體現在儲存受試者的測驗圖片、影片內容以及系統的相關資料，由於 Android 在各個版本針對系統權限及安全性都有些許變化，因此會導致不同版本之相容性問題，我們的 App 在系統權限的要求如下所示：

```

<uses-permission
    android:name="android.permission.READ_EXTERNAL_STORAGE"
    android:maxSdkVersion="32" />
<uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE"
    android:maxSdkVersion="32"
    tools:ignore="ScopedStorage" />
<uses-permission android:name="android.permission.READ_OWNER_DATA" />
<uses-permission android:name="android.permission.READ_OWNER_DATA" />
<uses-permission android:name="android.permission.READ_MEDIA_IMAGES" />
<uses-permission android:name="android.permission.ACCESS_MEDIA_LOCATION" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.INTERNET" />

```

Figure 25. AndroidManifest.xml 中關於系統權限請求的片段

我們在進行檔案儲存時，會根據受試者的 Android 系統版本，採用不同的檔案處理方式，來將測驗圖片或影片的內容保存到手機端上，並且同時刷新手機端的媒體庫，通知 Android 系統我們儲存了新的檔案，如此一來受試者便能成功在 Gallery 看到 App 中所保存的測驗內容檔案了。

```
// Refresh the Gallery
MediaScannerConnection.scanFile(context,
    new String[]{fileOutput.toString()}, mimeTypes: null,
    new MediaScannerConnection.OnScanCompletedListener() {
        no usages
        public void onScanCompleted(String path, Uri uri) {
            Log.i( tag: "ExternalStorage", msg: "Scanned " + path + ":" );
            Log.i( tag: "ExternalStorage", msg: "-> uri=" + uri );
        }
    });

```

Figure 26. 刷新媒體庫的程式碼片段

```
// Android Q Version
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.Q) {
    ActivityCompat.requestPermissions((Activity) context, new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE, Manifest.permission.READ_MEDIA_IMAGES},
    StorageManager storageManager = (StorageManager) context.getSystemService(Context.STORAGE_SERVICE);
    StorageVolume storageVolume = storageManager.getStorageVolumes().get(0); // Internal Storage
    Bitmap bitmapInputImage = myBitmap;

    ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
    bitmapInputImage.compress(Bitmap.CompressFormat.JPEG, quality: 100, byteArrayOutputStream);
    byte[] bytesArray = byteArrayOutputStream.toByteArray();
    File fileOutput = null;
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R) {
        fileOutput = new File( pathname: storageVolume.getDirectory().getPath() + "/Download/" + FilePrefix + formatter.format(curDate) + ".jpg");
    }
    FileOutputStream fileOutputStream = new FileOutputStream(fileOutput);
    fileOutputStream.write(bytesArray);

    // Refresh the Gallery
    MediaScannerConnection.scanFile(context,
        new String[]{fileOutput.toString()}, mimeTypes: null,
        new MediaScannerConnection.OnScanCompletedListener() {
            no usages
            public void onScanCompleted(String path, Uri uri) {
                Log.i( tag: "ExternalStorage", msg: "Scanned " + path + ":" );
                Log.i( tag: "ExternalStorage", msg: "-> uri=" + uri );
            }
        });
}

fileOutputStream.close();
}
```

Figure 27. Android Q (含) 以上的圖片儲存方法之部分程式碼片段

```
else // Old Version
{
    // 檢查權限是否已經擁有
    if (ContextCompat.checkSelfPermission(this.getContext(), Manifest.permission.WRITE_EXTERNAL_STORAGE) != PackageManager.PERMISSION_GRANTED) {
        // 權限未擁有，進行申請
        ActivityCompat.requestPermissions((Activity) context, new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE, Manifest.permission.READ_MEDIA_IMAGES});
    } else {
        try {
            fileName = FilePrefix + formatter.format(curDate) + ".jpg";

            ContentValues values = new ContentValues();
            values.put(MediaStore.Images.Media.DISPLAY_NAME, fileName);
            values.put(MediaStore.Images.Media.MIME_TYPE, "image/jpg");
            File directory = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DCIM);
            File file = new File(directory, fileName);
            values.put(MediaStore.MediaColumns.DATA, file.getAbsolutePath());

            Uri uri = context.getContentResolver().insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, values);
            try (OutputStream output = context.getContentResolver().openOutputStream(uri)) {
                myBitmap.compress(Bitmap.CompressFormat.PNG, quality: 100, output);
            }
            // Refresh the Gallery
            MediaScannerConnection.scanFile(context,
                new String[]{file.toString()}, mimeTypes: null,
                (path, uri) => {
                    Log.i( tag: "ExternalStorage", msg: "Scanned " + path + ":" );
                    Log.i( tag: "ExternalStorage", msg: "-> uri=" + uri );
                });
        } catch (Exception e) {
            Log.d( tag: "onBtnSavePng", e.toString()); // java.io.IOException: Operation not permitted
        }
    }
}
```

Figure 28. Android Q 以下的圖片儲存方法之部分程式碼片段

```

if (Start_Recording_Flag) {
    int Gross_Motor_Index = getIntent().getExtras().getInt("Index");
    Intent it = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
    Bundle bundle = new Bundle();
    bundle.putInt("Index", Gross_Motor_Index);
    it.putExtra(MediaStore.EXTRA_DURATION_LIMIT, value: 10);
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.Q) {
        ContentValues contentValues = new ContentValues();
        contentValues.put(MediaStore.Video.Media.DISPLAY_NAME, "GM" + (Gross_Motor_Index + 1) + "_" + getResources().getStringArray(R.array.Grade)[dataStorageIndex]);
        contentValues.put(MediaStore.Video.Media.MIME_TYPE, "video/mp4");
        Uri videoUri = getContentResolver().insert(MediaStore.Video.Media.EXTERNAL_CONTENT_URI, contentValues);
        it.putExtra(MediaStore.EXTRA_OUTPUT, videoUri);
    }
    startActivityForResult(it, REQUEST_VIDEO_CAPTURE);
}

```

Figure 29. Android Q (含) 以上的影片儲存方法之部分程式碼片段

除此之外我們還會保存受試者在 App 的系統資料至本機端，防止連不上資料庫造成的測驗資料遺失，我們使用了 DataStore 來進行資料儲存，該方法較為輕便且易於使用，很適合我們用於儲存較少量的資料。

```

public class MyDataStore {
    // 定義DataStore的文件名稱
    1 usage
    private static final String FILE_NAME = "MY_APP";
    // 聲明MyDataStore存儲數據的KEY鍵
    3 usages
    public final static Preferences.Key<Boolean> NEED_INIT_PERSONAL_INFO = PreferencesKeys.booleanKey( name: "need_init_personal_info");
    14 usages
    public final static Preferences.Key<String> NAME_KEY = PreferencesKeys.stringKey( name: "name");
    4 usages
    public final static Preferences.Key<String> PARENT_NAME_KEY = PreferencesKeys.stringKey( name: "parent_name");
    4 usages
    public final static Preferences.Key<String> PARENT_RELATION_KEY = PreferencesKeys.stringKey( name: "parent_relation");
    4 usages
    public final static Preferences.Key<Integer> SEX_KEY = PreferencesKeys.intKey( name: "sex");
    4 usages
    public final static Preferences.Key<String> BIRTHDAY_KEY = PreferencesKeys.stringKey( name: "birthday");
    4 usages
    public final static Preferences.Key<String> SCHOOL_KEY = PreferencesKeys.stringKey( name: "school");
    12 usages
    public final static Preferences.Key<Integer> GRADE_KEY = PreferencesKeys.intKey( name: "grade");
    4 usages
    public final static Preferences.Key<String> HEIGHT_KEY = PreferencesKeys.stringKey( name: "height");
    4 usages
    public final static Preferences.Key<String> WEIGHT_KEY = PreferencesKeys.stringKey( name: "weight");
    6 usages
    public final static Preferences.Key<Integer> HANDEDNESS_KEY = PreferencesKeys.intKey( name: "handedness");
    4 usages
    public final static Preferences.Key<Integer> FOOTEDNESS_KEY = PreferencesKeys.intKey( name: "footedness");
    4 usages
    public final static Preferences.Key<String> PHONEMODEL_KEY = PreferencesKeys.stringKey( name: "phonemodel");
    4 usages
    private static RxDataStore<Preferences> dataStore;
}

```

Figure 30. DataStore 用來儲存資料的 Preferences Key

```

// 寫入數據
38 usages
public <T> void putValue(Preferences.Key<T> key, T value) {
    dataStore.updateDataAsync(preferences -> {
        MutablePreferences mutablePreferences = preferences.toMutablePreferences();
        mutablePreferences.set(key, value);
        Log.e(tag: "DataStore", msg: "寫入數據--" + key + "->" + value);
        return Single.just(mutablePreferences);
    });
}

// 獲取數據
33 usages
public <T> T getValue(Preferences.Key<T> key) {
    Flowable<T> value = dataStore.data().map(preferences -> preferences.get(key));
    return value.blockingFirst();
}

```

Figure 31. DataStore 用來存取資料的函式

接著則是系統設置，這部分由於早期療育中心並沒有特別要求，因此我們只有針對部分功能進行設置，包括注音模式以及語言設置，注音模式是為了某些歲數較小的兒童，在家長沒有協助操作下會難以進行文字閱讀，因此才會想到可以添加這個功能。我們採取的作法基本上就是更改 App 的字體，不過由於注音字體會比一般的字體還要寬，在版面配置上會顯得稍微擁擠，會需要額外花時間進行調整，所以目前我們尚未將其完整實作於 App 上。另一個功能則是語言設置，在比較正式的 App 開發上，會針對不同國家進行語言設置，我們也粗淺地將部分 strings.xml 的內容翻譯成英文以及日文，不過目前該系統主要的受眾群體仍是以台灣人為主，因此我們尚未將其完整實作於 App 上，但未來若要增加多國語系，只需於 strings.xml 上進行擴充即可。



Figure 32. 套用注音字體的個人資料設定頁面



Figure 33. 不同語言的 strings.xml

最後稍微補充一下我們在精細動作測驗頁面的畫布（Canvas）設計，我們主要是利用 Android Studio 的 onTouchEvent（ACTION_DOWN、MOVE、UP）來判斷受試者對於螢幕上的操作行為，因此無論受試者是使用手指或是觸控筆皆能夠進行測驗，而針對在畫布上繪製的線條，我們是採取貝茲曲線（Bézier Curve）來進行擬合及繪製：

$$B(t) = (1 - t)^2 P_0 + 2t(1 - t)P_1 + t^2 P_2$$

其中 $t \in [0,1]$ ，如此便能較好的擬合受試者進行曲線的繪製行為。

```
// While Move Finger, Keep Refreshing Moving Path
1 usage
private void touch_move(float x, float y) {
    // Calculate Previous Point and Current Point Difference
    float dx = Math.abs(x - myX), dy = Math.abs(y - myY);
    // Judging the Difference is Tolerance or Not, if Bigger than Tolerance Value then Drawing
    if (dx >= TOUCH_TOLERANCE || dy >= TOUCH_TOLERANCE) {
        // Drawing Bezier curve
        myPath.quadTo(myX, myY, x2: (x + myX) / 2, y2: (y + myY) / 2);
        // Refreshing Previous (X, Y) Coordinate
        myX = x;
        myY = y;
        // Destroy Previous Circle
        myCirclePath.reset();
        // Refreshing Current Circle's Position
        myCirclePath.addCircle(myX, myY, radius: 30, Path.Direction.CW);
    }
}
```

Figure 34. Canvas 中利用貝茲曲線進行擬合及繪製的程式碼片段

為了能將受試者的基本資料以及測驗的結果記錄下來，經由與早療療育中心的同仁討論，我們決定要使用 Firebase 來串接到 Android Studio 上進行使用，因為 Firebase 的母公司是 Google，對於 Android 系統的相容性會比一般的資料庫系統來得更加優秀，而且在 Android Studio 上也能很方便地呼叫 API 來進行使用。

因為 Firebase 的資料儲存方式是採取樹狀結構進行儲存，屬於 BaaS（Backend as a Service），其 Realtime Database 可透過 JSON 格式存取來提供即時更新的 NoSQL 資料庫。因此我們決定先畫 Entity–Relationship Model，以利於我們進行資料庫系統開發，由於早期療育中心希望能儲存個人資料、測驗圖片、影片，其中圖片格式為.jpg，影片格式為.mp4，因此

檔名便顯得相當重要，我們會在測驗圖片或影片的檔名後面添加時間戳記，前綴包含早期療育中心所需要的一些基礎個人資訊（例如年級、姓名等...），如此一來便比較不會發生檔名重複的問題，才能夠完整地將測驗的內容儲存到資料庫系統中。

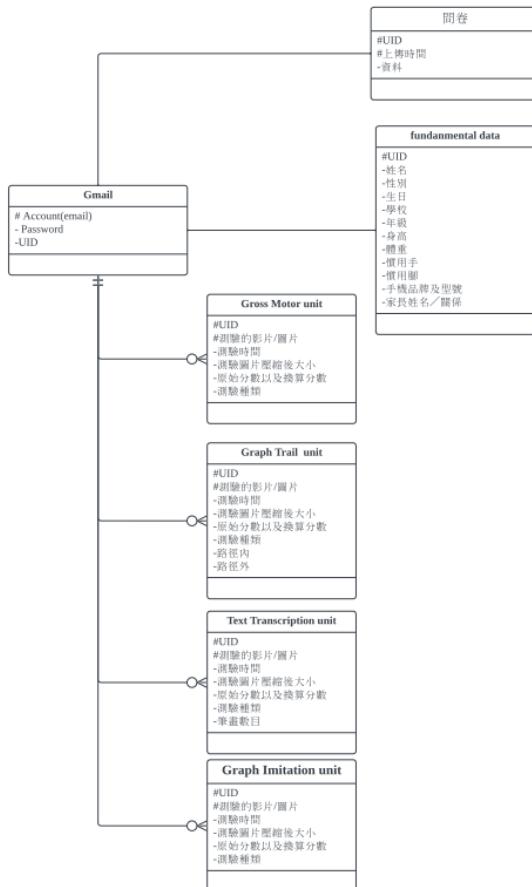


Figure 35. 資料庫系統中初步規劃的 Entity–Relationship Model

我們原先的規劃是使用 Gmail 來進行註冊以及登入，不過早期療育中心會將這個 App 進行收案使用，但一個國小班上可能會有數十個人，且不是大家都有 Gmail，因此我們增加了訪客登入來解決此問題，但如果是使用訪客登入，便不會有 Gmail 來當作 Primary Key，所以我們決定在訪客登入時給予受試者一組唯一的 UID (若使用 Gmail 進行註冊時也會獲得一組唯一的 UID)，這個 UID 將採取流水編號遞增的方式來當作 Primary Key 以區分不同的受試者，當受試者登入後，倘若有填寫基本資料，如學校、年級、性別、慣用手等...，我們便會將其保存至本機端並上傳至資料庫系統，這些資訊能夠方便早期療育中心分類出這位受試者應當被分類到的位置。

接下來是測驗資料的部分，分別有精細動作測驗的圖形仿畫、路徑內臨摹、文字謄寫以及粗大動作測驗等...。圖形仿畫有很多種標準圖形，我們主要是著重在多菱形的部分，主要記錄受試者仿畫的圖形以及經由 CNN 模型所計算出來的分數。測驗的第一開始會給受試者一張練習圖來練習，這將有助於使用者熟悉整個測驗的操作模式，接著就進入正式圖的部分，每個測驗都會有一個完成測驗按鈕，當按下確認鍵的同時會將這筆測驗的圖形、分數以及其他相關資訊紀錄下來，並將其儲存到資料庫系統。

路徑內臨摹與前面的圖形仿畫大致相同，我們會記錄受試者臨摹的圖形以及經由 IOU 及前述的數學公式所計算出來的分數，但是會多紀錄路徑內跟路徑外這兩筆原始資料，而非只是轉換後的分數，測驗的第一開始一樣會給受試者一張練習圖來練習，正式圖也是按下確認鍵的同時將該筆資料儲存下來。文字謄寫的部分基本上與前兩者相同，但需要額外儲存筆畫數目，因為使用者需要按照我們給的標準字來寫出一模一樣的文字，我們能夠藉由筆畫數目來判斷文字的相似與否，正式圖也是按下確認鍵的同時將該筆資料儲存下來。至於粗大動作測驗，分數的評判會由其它與早期療育中心合作的同仁來負責，我們主要是先幫他們建立好收案所需的資料庫系統，以及讓他們能夠直接使用 App 進行影片錄製，並產生相對應的檔名及格式，以方便他們進行後續的規劃與擴充。

最後是家長問卷的部分，早期療育中心提供家長（或受試者）反饋意見的地方，這部分他們還在設計問卷內容，等到設計完成後我們會在替他們加入至 App 中，以利於早期療育中心統整大家的意見（這時候便是以個人操作為主，而不是採用收案的方式進行），這將有助於未來系統開發的參考依據以及後續修正。

You can throw the Exception returned by `task.getException` inside a try block and catch each type of Exception that may be thrown by the method you are using.

Here is an example from the `OnCompleteListener` for the `createUserWithEmailAndPassword` method.

```
if(!task.isSuccessful()) {
    try {
        throw task.getException();
    } catch(FirebaseAuthWeakPasswordException e) {
        mTxtPassword.setError(getString(R.string.error_weak_password));
        mTxtPassword.requestFocus();
    } catch(FirebaseAuthInvalidCredentialsException e) {
        mTxtEmail.setError(getString(R.string.error_invalid_email));
        mTxtEmail.requestFocus();
    } catch(FirebaseAuthUserCollisionException e) {
        mTxtEmail.setError(getString(R.string.error_user_exists));
        mTxtEmail.requestFocus();
    } catch(Exception e) {
        Log.e(TAG, e.getMessage());
    }
}
```

Figure 36. Firebase API 的常見 Exception

```
reference.child(NameString).child(pathString: "Name").setValue(NameString);
reference.child(NameString).child(pathString: "Sex").setValue(SexArrayIndex);
reference.child(NameString).child(pathString: "Birthday").setValue(BirthdayString);
reference.child(NameString).child(pathString: "School").setValue(SchoolString);
reference.child(NameString).child(pathString: "Grade").setValue(GradeArrayIndex);
reference.child(NameString).child(pathString: "Height").setValue(HeightString);
reference.child(NameString).child(pathString: "Weight").setValue(WeightString);
reference.child(NameString).child(pathString: "Handedness").setValue(HandednessArrayIndex);
reference.child(NameString).child(pathString: "Footedness").setValue(FootednessArrayIndex);
reference.child(NameString).child(pathString: "PhoneModel").setValue(PhoneModelString);
```

Figure 37. Firebase API 以 child 的樹狀結構方式儲存基本資料



Figure 38. Firebase 中以樹狀結構方式查看 UID 為 4578 的基本資料

```

if (dataStore.getValue(MyDataStore.IS_CASE_COLLECTION_MODE)) { // 收案模式
    String Subpath = dataStore.getValue(MyDataStore.NAME_KEY) + "_" + dataStore.getValue(MyDataStore.CASE_NUMBER_KEY);
    db.collection( collectionPath: "Cases" ).document(Subpath).DocumentReference
        .collection( collectionPath: "Graph Trail" ).document(currentTime)
        .set(scores).Task<Void>
        .addOnSuccessListener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                Log.d(TAG, msg: "DocumentSnapshot successfully written!");
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Log.w(TAG, msg: "Error writing document", e);
            }
        });
} else {
}

```

Figure 39. 同步測驗結果到 Firebase 之程式碼片段

```

Bitmap gray = CanvasView.Rec(); // Preprocessed img
TensorImage tensorImage = new TensorImage(DataType.FLOAT32);
tensorImage.load(gray);

// 使用 ImageProcessor 將圖像轉換為單通道灰度 & 標準化
ImageProcessor imageProcessor = new ImageProcessor.Builder()
    .add(new TransformToGrayscaleOp()) // 將圖像轉為灰度
    .add(new NormalizeOp( mean: 0.0f, stddev: 255.0f))
    .build();

tensorImage = imageProcessor.process(tensorImage);

```

```

        break;
    case "The_3DSquare":
        // Load model
        AutoModel3dsquare02122024 model_3d = AutoModel3dsquare02122024.newInstance(context);
        // Runs model inference and gets result.
        AutoModel3dsquare02122024.Outputs outputs_3d = model_3d.process(inputFeature0);
        TensorBuffer outputFeature0_3d = outputs_3d.getOutputFeature0AsTensorBuffer();
        // Get result
        resultArray = outputFeature0_3d.getFloatArray();
        // Releases model resources if no longer used.
        model_3d.close();
        break;
    default:
        break;
}

//Log result
if(resultArray != null){
    Log.i(TestCur, msg: "float_score" + Arrays.toString(resultArray));
    // 初始化 roundedArray 的大小與 resultArray 相同
    roundedArray = new int[resultArray.length];

    for (int i = 0; i < resultArray.length; i++) {
        roundedArray[i] = Math.round(resultArray[i]);
    }
}

```

Figure 40. 串接模型之程式碼片段

```

CanvasView = findViewById(R.id.TranscriptionCanvasView);
//xieView = findViewById(R.id.xieid); //New
mGifImageView = findViewById(R.id.activity_gif_giv); //New

TestName = getIntent().getExtras().getString(key: "Next_TestName");
String Guidance_Title;

ImageView Standard_Diagram = (ImageView) findViewById(R.id.Transcription_STD_Diagram);

//GifImageView xieView = findViewById(R.id.xieid);

switch (Objects.requireNonNull(TestName)) {
    case "Xie":
        Guidance_Title = "正式題（一）：寫字謄寫測驗";
        Standard_Diagram.setImageResource(R.drawable.xie);
        mGifImageView.setImageResource(R.drawable.xiegif);
        break;
}

```

Figure 40. gif 顯示謄寫文字之程式碼片段

結論與未來展望

我們的系統核心目標是為早期療育提供一客觀、有效的工具，協助評估兒童在精細動作能力上的表現，我們透過深入的合作，結合醫學和資訊技術，確保系統在技術面結實，更能貼近實際的臨床需求，我們也盡可能使操作簡化，並保持良好的擴充性，如分數評估方法能適用於所有相同類型的測驗，因此即使新增或修改題目，系統也不需要進行太大規模的變更。經由這次系統的開發，我們確立了精細動作能力的電子化測驗評估方式，並實作於 App 上供早期療育中心收案使用，然而系統的常模及效度均需仰賴長期的收案來建立，因此目前較難以評估哪個年齡段的兒童應該要介於什麼分數段才是常態，但評估分數的高低確實能反應精細動作能力。

根據目前的測驗情形來看，路徑內臨摹測驗的評估最準確的，畢竟該測驗本身是基於像素層面的 IOU 計算，搭配我們的數學公式轉換（目前設 $\alpha = 1$ ），可以達到近乎無偏誤的評分。再者是多菱形仿畫測驗，CNN 模型的訓練準確率為 90.21%，預測準確率為 89.02%，因受限於訓練集資料量的影響，在我們的測試下，發現模型對於部分較特殊的資料表現不佳（如測驗圖與多菱形圖相差甚多），這部分需仰賴後續收案增加資料來提升準確率，但整體而言模型對於相似度的比較還是很準確的。最後則是文字謄寫，由於我們是基於 OCR 比對，OCR 會受到信心程度的影響來決定最相近的文字，導致分數呈不平均的分布，因此該測驗可能還需要考量更多的評分方式來進行改善。

以下是我對這次專題合作經歷的反思：

- (1) **跨領域合作的價值**：我們的團隊中有教授、醫師、技術教師、研究生、工程師等…，每個人都能提供不同的專業知識，以不同的視角來切入問題，這種合作方式有助於將醫學知識與資訊技術相結合，未來我將更加重視這點，以解決更多複雜的需求。
- (2) **技術對社會的貢獻**：我們所開發的系統，旨在幫助早期療育人員評估兒童在精細動作能力上是否有發育遲緩的現象，系統開發過程所討論的技術及成果，將有助於改善兒童的生活品質，我更深刻地體會到技術可以對社會產生積極的影響。
- (3) **專業知識及技能的培養**：在系統開發的過程，時常會牽涉到不同領域的內容，而且在實際開發的過程中，會遭遇許多無法預期的問題，因此平時就要多培養相關的能力，並檢視彼此之間的相關性，才能在遇到問題時，快速地找出有效的解決方式。
- (4) **在理想與現實取得平衡**：在進行系統設計時，時常會感受到理論與實務上不容易達成完全一致，且礙於時間與成本的考量，我們需要在設計的過程採用好的折衷方式，盡可能地達成雙方的共識，來設計符合實際需求的應用程式。

本系統作為早期療育的評估工具，具備十分廣泛的發展潛力，未來我們將致力於以下方向的拓展，以進一步提升系統的功能性及實用性：

- (1) **結合穿戴式裝置**：我們將研究如何結合穿戴式裝置，以擴展評估的範疇至更多日常活動，如孩童堆疊積木、使用餐具等...精細動作。這不啻可以提供更多元、真實場景下的活動數據，還可結合 AI 模型進行實時動態評估，找尋兒童在各種活動中的表現特徵，以利於後續臨床上之分析，提供最佳的醫療建議、擬定治療計畫等...，協助兒童動作能力的恢復。
- (2) **多感知資訊評估**：引入錄影機或其他感知設備來觀察兒童在測驗過程中的行為，同時結合帶有時序濾波器的 CNN 模型進行評估，更深入且準確地分析兒童的動作特徵。這不啻提供更全面、詳細的評估，還有助於建立動作能力之時序模型，能更好地反映兒童的發展軌跡。
- (3) **維持系統易用性**：隨著輔助設備的增加（如前述提到的穿戴式裝置、錄影機、其他感知設備等...），會顯著的拉高大眾使用的門檻，我們將特別關注系統的易用性。藉由深入研究使用者體驗，透過人機介面的優化，致力於開發直觀且簡便的操作介面，讓孩童也能毫無負擔的使用，並透過使用者的反饋，不斷調整系統設計，確保系統操作的簡易性，同時兼顧測驗的準確性和效度。
- (4) **效度及準確性提升**：為提高系統對不同受試者的評估準確性，我們將持續擴充數據集，針對特殊情況（甚至包括 Adversarial Attack）來進行模型的優化。隨著收案的次數增加，我們將建立更完善的常模，確保系統的效度和準確性，並在未來引入持續學習的概念，使系統能夠不斷適應新的資訊和使用情境，能不斷地帶來醫療層面的價值。
- (5) **跨領域合作**：我們將積極尋求與其他領域的合作，包括醫學、教育學、心理學等...，這種跨領域的合作能夠為系統發展提供更多元的視角和跨領域的專業知識。在將來也有機會舉辦工作坊和研討會，促進不同領域之間的交流，以共同推進早期療育評估的研究與實踐。

透過這些未來展望的目標，我們期望本系統能夠更全面，為提升兒童動作能力發展的評估準確性提供強而有力的支持。我們期盼這些研究能使系統成為該領域的有效工具，為早期療育領域，為台灣人工智能及醫療領域盡一份心力！

柒、 成員心得

陳昱佑：

在這次的專題開發中，我有幸參與了開發一款專門用於評估兒童精細動作的 APP，這是一段既挑戰又收穫豐富的經歷。從需求分析到系統設計，再到實際開發與測試，每一步都讓我深刻體會到跨領域合作的重要性，以及技術在醫療與教育應用中的無限可能性。

在技術層面，我主要負責使用 Android Studio 建立應用程式框架，並整合 TensorFlow Lite Library 開發出一套基於機器學習的動作偵測模型。該模型能夠即時分析使用者的動作數據，提供精確的評估結果。此外，我運用了 JDBC API 與後端資料庫進行連結，確保使用者數據能夠穩定儲存與調用；透過設計 Web API，我成功實現了前後端的順暢通訊。同時，整個應用程式的使用者認證與數據備份功能，則依賴於 Firebase 平台，這讓開發效率與安全性大幅提升。

除了技術挑戰外，這次專題實作也讓我學習到如何與跨領域專家合作。魏志達教授提供了寶貴的技術指導，幫助我克服許多技術難點。他對程式結構化設計的要求與建議，讓我受益匪淺。陳嘉玲醫師則分享了許多與兒童動作發展相關的醫學知識，使我對應用場景有更深的理解，進一步優化了 APP 的功能設計。而陳姍妤學姊在實務經驗上的傳授，不僅提升了我的專案管理能力，也教會我如何更有效地與團隊成員溝通與合作。

這次的經歷不僅讓我技術能力更加精進，也讓我體會到科技如何結合醫療與教育，為社會帶來實質幫助。未來，我希望能繼續朝著這個方向努力，開發更多具有實際價值的應用程式，再次感謝魏志達教授、陳嘉玲醫師與陳姍妤學姊對我的指導與幫助。

蕭友翔：

參與兒童精細動作檢測系統的開發，讓我深刻體會到科技與醫療結合的價值與挑戰。在這段時間裡，從系統設計、深度學習模型優化到應用程式開發與資料庫管理，參與了專案的某些環節，每一步都為我帶來全新的成長與啟發。

在模型開發階段，學會了如何使用卷積神經網絡（CNN）分析精細動作，並將模型移植到 Google Colab 與 Git 平台，以促進團隊協作與版本管理。同時，根據發展遲緩兒童的特殊需求調整模型，大幅提升了分析的準確性與穩定性。這段經歷能讓我們更加理解，如何將複雜的技術應用於實際問題，尤其是透過 AI 技術來增強醫療工具的效能。

在應用程式的設計與優化過程中，我深刻感受到使用者經驗（UX）的重要性。我為不同的使用者群體設計了直觀友好的介面，加入了注音功能和趣味性插圖，使測驗更具吸引力。此外，我開發了多樣化的測驗模式與結果回饋功能，確保應用不僅實用，還能提供即時洞察，進一步提升使用者滿意度。

資料庫設計方面，透過 Firebase 構建了一個高效、安全的資料結構，並優化了資料同步

與備份功能，確保系統在多種情況下都能穩定運行。這些經驗不僅豐富了我們的技術技能，也更能從全局的角度看待專案的開發與管理。

這次專題還讓我對早期療育有了更深刻的理解。我了解到早療的多元性，不同的兒童可能需要針對語言發展、運動技能或社交能力等不同領域的介入。我們的專題正是希望能為家長和孩童提供一款實用且精準的評估工具，助力早療的推進。我也深刻感受到早期介入對兒童發展的重要性，它能幫助孩子在成長中獲得更好的支持，發揮最大的潛能。這段經歷不僅提升了我的技術能力，更讓我明白團隊合作與跨領域專案的重要性。未來，我希望繼續投入科技與醫療的融合，為促進人們生活品質的提升貢獻自己的力量。

區迅祁：

在這次的專題中，我學到了許多寶貴的經驗與技能。這個專題讓我深入了解了軟體開發的流程，從問題定義到解決方案的設計和實現，一步步地體驗了整個過程。透過與團隊合作，我學會了如何有效溝通和協作，以及如何將各種想法和專業知識整合到一個項目中。在技術方面的收穫外，我也學到了許多軟技能，例如時間管理和解決問題的能力。在專題中，我們經常遇到各種挑戰和障礙，需要快速找到解決方案。這讓我懂得了如何靈活應對困難，並從失敗中學習，不斷改進我們的工作。這個專題也讓我更了解了資訊工程領域的深度和廣度。從機器學習到軟體開發，我接觸到了各種不同的技術和工具，這擴展了我的視野，讓我對未來的職業發展有了更清晰的想法。最後通過這個專題，我不僅增加了專業知識和技能，更培養了團隊合作和問題解決的能力，這將對我的未來職業生涯產生深遠的影響。

葉祐齊：

藉由此次專題的機會，我學到了許多專業知識與技術，首先是應用程式的相關技術，熟悉了 Java 與 Kotlin 的語法，並在 Android Studio 上進行應用程式的開發與維護。在這段過程中，我還掌握了多種實作方法，包括應用程式元件的使用、UI 設計及與資料庫溝通的方式等。

在兒童精細動作評估系統的開發中，我負責整合模型、資料庫與 APP 的串接使用，同時進行 APP 功能的開發。在參與整個系統開發的過程中，我對軟體系統的整體架構有了更清晰的認識，學會了如何設計高效、易於管理的系統。我也開始意識到溝通的重要性，通過與魏志達教授、陳嘉玲醫師與陳姍妤學姊等人的交流，不僅能更有效率地解決問題，還能獲取許多實務上的寶貴經驗，這讓我學會了如何平衡用戶需求與技術實現。

在專題過程中，我也遇到了一些挑戰。由於應用程式需要實作多種功能，經常需要進行測試以找出程式中的 bug，並不斷完善介面與功能。這段過程充滿挑戰，尤其是在面對開發環境更新時遇到的相關問題，還有一些複雜功能的實現，讓我感受到了技術的深奧與複雜，並認識到自己在技術細節和工具運用上還有許多進步空間。

從大三到現在歷時約一年半，專題終於告了一個段落，系統在收案上還算順利。我們的專題與早期療育中心合作，做這個專題的過程中，我們碰到很多問題，像是模型的串接、

App 端各種功能的開發等等，都是此前未接觸過的領域，幸好網路上有許多 Android Studio 開發相關的學習資料，給了我不小的幫助。我也負責資料庫調整的部分，這個 firebase 跟之前學的資料庫有很大的不同，花了不少時間去研究，好在經過一陣子的努力，終於能較熟練地運用 Android Studio 跟 Firebase 兩個工具。最後我要感謝專題的同組同學，我們在各個方面互相協助，沒有他們的話，很難順利完成這次的專題，非常感謝他們。

捌、 參考文獻

- Gerber, R. J., Wilks, T., & Erdie-Lalena, C. (2010). Developmental milestones: motor development. *Pediatrics in review*, 31(7), 267–277. <https://doi.org/10.1542/pir.31-7-267>
- Li, R., Fu, H., Zheng, Y., Lo, W. L., Yu, J. J., Sit, C. H. P., Chi, Z., Song, Z., & Wen, D. (2019). Automated Fine Motor Evaluation for Developmental Coordination Disorder. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 27(5), 963–973. <https://doi.org/10.1109/TNSRE.2019.2911303>
- Polsley, S., Powell, L., Kim, H., Thomas, X., Liew, J., & Hammond, T.A. (2021). Detecting Children’s Fine Motor Skill Development using Machine Learning. *International Journal of Artificial Intelligence in Education*, 32, 991 - 1024.
- Patel, A., & Polsley, S., & Hammond, T. A. (2022, March), Using Neural Networks to Distinguish Children’s Age with Visual Features of Sketches Paper presented at 2022 ASEE Gulf Southwest Annual Conference, Prairie View, Texas. <https://peer.asee.org/39220>