

中國文化大學
資訊工程學系
資訊系統專題

糖尿病視網膜病變辨識系統

學 生：俞 華 鈴

李 馥 彤

林 永 泰

指導教授：林 世 崧

中 華 民 國 114 年 5 月

糖尿病視網膜病變辨識系統

專題學生：俞華鈴、李馥彤、林永泰

指導教授：林世崧 博士

中國文化大學 資訊工程學系

摘要

隨著現代社會飲食習慣與生活型態改變，糖尿病患者人數快速上升，導致因糖尿病引起的視網膜病變成為全球成人失明的主要原因之一。早期發現與治療是避免視力喪失的關鍵，特別是在醫療資源較匱乏的偏鄉地區，診斷延遲問題尤為嚴重。本專題旨在建構一套基於深度學習技術的糖尿病視網膜病變自動辨識系統，協助醫師提升診斷效率，並縮短偏鄉醫療落差。系統採用眼底影像作為輸入，最終目標為提供一個可實際應用於初步篩檢的輔助診斷系統，降低病患視力喪失的風險，並減輕後續醫療負擔。

關鍵詞：深度學習、卷積神經網路、資料前處理與增強、影像辨識

指導教授_____ (簽名)

目 錄

摘 要.....	I
目 錄.....	II
表 目 錄.....	IV
圖 目 錄.....	V
第 1 章 研究動機與目的	1
1.1 研究動機	1
1.2 研究目的	1
1.3 研究流程	2
第 2 章 文獻探討	4
2.1 糖尿病	4
2.2 糖尿病型視網膜病變	5
2.3 糖尿病視網膜病變公開據集	8
2.4 遷移式學習	8
第 3 章 研究內容與方法	10
3.1 訓練架構	10
3.2 流程說明	12
3.3 軟硬體資源規劃	13
3.4 模型訓練	14
3.4.1 CNN	14
3.4.2 VGG16	17
3.4.3 ResNet50	19
第 4 章 研究成果	25
4.1 系統架構	25
4.1.1 系統流程設計	26
4.1.2 互動介面設計	27
4.2 辨識成果	30

4.2.1 CNN	30
4.2.2 VGG16	34
4.2.3 ResNet50	36
4.2.4 統整與比較	39
第 5 章 人力配置	41
第 6 章 預期成果	42
第 7 章 執行進度	43
參考文獻	44
附錄 A 開發工具介紹	45
A.1 軟體開發工具介紹	45
A.1.1 Visual Studio Code	45
A.1.2 TensorFlow	45
A.1.3 Colab	47
A.2 影像辨識電腦硬體	48

表 目 錄

表 2.1 視網膜病變眼底影像分級	7
表 2.2 糖尿病視網膜病變資料集	8
表 3.1 本研究實驗軟硬體環境	13
表 3.2 CNN 模型架構	15
表 3.3 VGG16 模型架構	17
表 3.4 ResNet50 模型架構	22
表 4.1 CNN 二分法指標	32
表 4.2 CNN 四分法指標	33
表 4.3 VGG16 指標	35
表 4.4 ResNet50 指標分析	39
表 4.5 加權平均指標數據	39
表 5.1 參與本項研究之人員及工作內容	41
表 A.1 影像辨識電腦	48

圖 目 錄

圖 1.1	研究流程圖	2
圖 2.1	Layer Transfer [7]	9
圖 3.1	系統架構圖	10
圖 3.2	糖尿病視網膜辨識系統架構圖	11
圖 3.3	研究流程	12
圖 3.4	高斯處理變化圖	14
圖 3.5	ResNet50 模型實作流程圖	20
圖 4.1	系統架構圖	25
圖 4.2	系統流程設計圖	26
圖 4.3	GUI 介面圖	27
圖 4.4	Stage1 影像分析.....	27
圖 4.5	No DR 辨識結果	28
圖 4.6	Stage1 與 Stage2 影像分析過程圖	28
圖 4.7	Moderate DR 辨識結果	29
圖 4.8	二分法準確率	30
圖 4.9	CNN 二分法 Accuracy 與 Loss 曲線圖	31
圖 4.10	CNN 混淆矩陣	31
圖 4.11	四分法準確率	32
圖 4.12	CNN 四分法 Accuracy 與 Loss 曲線圖	32
圖 4.13	CNN 四分法混淆矩	33
圖 4.14	VGG16 準確率	34
圖 4.15	VGG16 Accuracy 與 Loss 曲線圖	34
圖 4.16	VGG16 混淆矩陣	35
圖 4.17	ResNet50 Accuracy 與 Loss 曲線圖	36
圖 4.18	ResNet50 損失函數	37
圖 4.19	ResNet50 標準化混淆矩陣	37

圖 7.1	執行進度甘特圖	43
圖 A.1	Visual Studio Code	45
圖 A.2	TensorFlow	46
圖 A.3	Colab	47

第 1 章 研究動機與目的

1.1 研究動機

隨著全球糖尿病患者人數不斷攀升，糖尿病視網膜病變已成為成年人失明的主要原因之一。尤其在偏遠地區，受限於醫療資源不足與專業診斷人員匱乏，病患往往無法及時接受檢查與治療，導致視力不可逆損傷。

傳統眼底檢查需依賴專業醫師目視判讀，程序耗時且受限於設備與技術普及程度。隨著深度學習與醫療影像技術的進步，利用人工智慧協助糖尿病視網膜病變的早期偵測成為可行且迫切的需求。因此，本研究期望藉由科技手段縮短診斷流程，提升篩檢覆蓋率，減緩糖尿病引發的失明問題，改善偏鄉醫療可及性。

1.2 研究目的

本研究旨在開發一套基於深度學習技術的糖尿病視網膜病變自動辨識系統，使用眼底影像進行五級分級診斷，輔助醫療人員進行初步篩檢。透過卷積神經網路（CNN）模型訓練與優化，提高影像辨識的準確率與穩定性，並結合雲端平台，實現遠端即時診斷功能。最終目標為建立一個高效、準確且易於部署的醫療輔助系統，以提升早期診斷率，降低視力損失風險，並促進偏鄉地區醫療服務之普及與提升。

1.3 研究流程

本研究架構如圖 1.1，介紹糖尿病視網膜病變的研究背景及相關議題，延伸的研究動機及為找出糖尿病視網膜病變檢測需要解決及改善的問題。

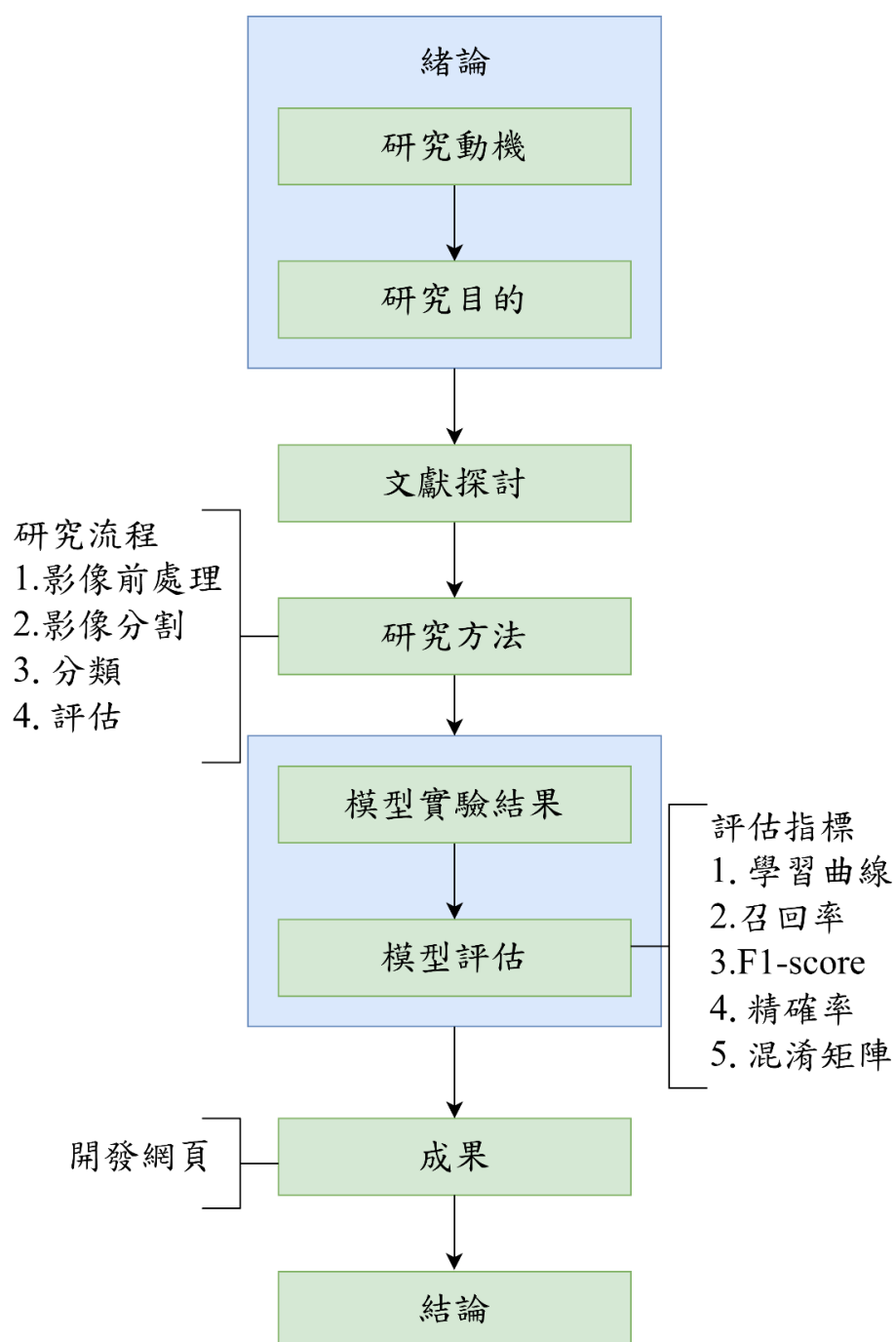


圖 1.1 研究流程圖

於文獻部分，將由現有的糖尿病視網膜病變自動化流程相關文獻整理，並探討過去學者所提出的技術，在整理的文獻範圍內篩選並明確研究主題。接著建立本研究架構。

實驗部分則針對模型進行訓練與測試，獲得模型實驗結果後，使用多項指標進行綜合性評估，模型評估使用「學習曲線」、「召回率」、「F1-score」、「精確率」及「混淆矩陣」，藉以全面觀察模型在不同面向的表現與穩定度。

開發網頁式糖尿病視網膜病變輔助診斷系統，提供視覺化介面，協助臨床醫師快速且準確判讀視網膜影像之病變程度，期能促進早期發現與即時治療。

第 2 章 文獻探討

2.1 糖尿病

根據國際糖尿病聯盟（International Diabetes Federation, 2025）[1] 的統計，到了 2025 年，全球糖尿病患者數量將突破 3 億人。糖尿病不僅造成血糖失控，更可能引發心血管、腎臟、神經系統及視覺系統等多重器官的併發症，從生活品質下降，到失明、洗腎甚至截肢，對患者個人、家庭乃至整個社會帶來沉重衝擊。

糖尿病是一種因血糖濃度過高、胰島素分泌不足或作用異常所引發的代謝性疾病。依據類型不同，糖尿病的併發症可分為急性與慢性兩大類[2]。急性併發症中最常見的是血糖控制不穩，特別是低血糖症，患者一旦出現症狀，需即時補充含糖食物以緩解危機。慢性併發症則影響更為深遠，涵蓋眼睛、腎臟、心血管以及神經系統等，長期發展下可能導致不可逆的器官損傷。

其中，糖尿病視網膜病變（Diabetic Retinopathy, DR）不僅涉及血管異常，如血管通透性增加、微血管阻塞及新生血管形成，也包含視網膜神經細胞的功能障礙與退行性變化，且神經病變可能早於血管病變出現[3]。

2.2 糖尿病型視網膜病變

糖尿病型視網膜病變根據以下病變特徵進行嚴重等級分級，包含微動脈瘤、視網膜出血、硬性滲出物、軟性滲出物(Soft Exudates)、視網膜內微血管異常(IRMA)、視神經血管新生及任何地方血管新生等，視網膜病變主要分為以下兩類[4]：

(1) 非增值性 (Non-Proliferative Diabetic Retinopathy, NPDR)：

非增值性視網膜病變特徵包含微血管瘤、棉絮一樣的斑點、硬性滲出物和出血的情況，根據這些病徵分為輕度、中度和重度三個階段。

(2) 增值性 (Proliferative Diabetic Retinopathy, PDR)：

視網膜的血液供應不足導致視網膜內的微血管病變、滋生新的血管，過多的血管造成眼球出血、視網膜剝離的情形，嚴重會導致視力下降及失明的危險。

其詳細的特徵分類，根據病徵可以分為以下五個等級：

(1) 無視網膜病變（No DR）：

無明顯視網膜病變。

(2) 輕度非增殖性視網膜病變（Mild NPDR）：

僅有微動脈瘤。

(3) 中度非增殖性視網膜病變（Moderate NPDR）：

微動脈瘤和出血較多，以及棉絮狀斑、靜脈串珠。

(4) 重度非增殖性視網膜病變（Server NPDR）：

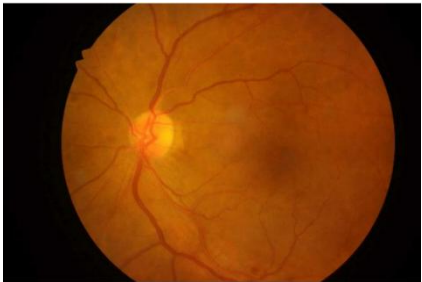
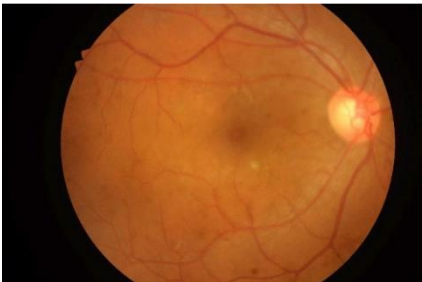

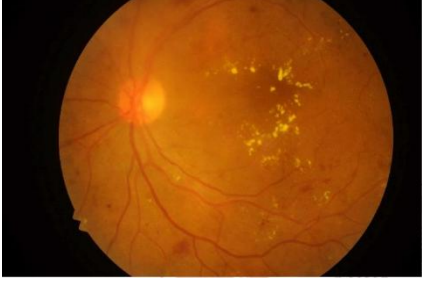

明顯的視網膜內微血管異常、靜脈串珠。

(5) 增值性視網膜病變（PDR）：

視網膜新生血管形成、玻璃體或視網膜前出血、纖維增生。

表 2.1 為各等級眼底影像示意圖。

表 2.1 視網膜病變眼底影像分級

DR 嚴重分級	DR 眼底影像
無視網膜病變 (No DR)	
輕度非增值性視網膜病變 (Mild NPDR)	
中度非增值性視網膜病變 (Moderate NPDR)	
重度增值性視網膜病變 (Sever NPDR)	
增值性視網膜病變 (PDR)	

2.3 糖尿病視網膜病變公開據集

在分級實驗中，這些數據將被切分為訓練集、測試集及驗證集，測試及和驗證及通常用於評估系統效能，進一步用來與其他研究系統效能上進行比較。其中資料集依據疾病嚴重程度分為五類，內容包含訓練及與測試集的張數、是否為多個專家標記等資訊。除了 IDRiD 資料集為固定的分辨率(4288*4288)，APTOS 2019 資料集則為不固定影像分辨率。APTOS 2019 資料集的總圖片數量為 5,590 張，其中包含標記 3,662 張。糖尿病視網膜病變資料集的資訊如表 2.2 所示。

表 2.2 糖尿病視網膜病變資料集

資料庫	訓練集(張)	測試集(張)	總數(張)	分類數量	是否標記
IDRiD[5]	413	103	513	5	是
APTOS 2019[6]	3662	1928	5590	5	是

2.4 遷移式學習

遷移式學習 (Transfer Learning) 是一種機器學習技術，它的核心概念是將一個任務上學到的知識，轉移到另一個相關任務上。不需要每次都從頭訓練模型，而是可以把已經在大量資料上訓練好的模型。

像是 ImageNet 上的 ResNet、VGG、EfficientNet 等，拿來「微調（fine-tune）」或「凍結部分層」後，應用在新任務上，模型首先於來源資料上訓練，取得具代表性的特徵表達，將前面幾層保留，代換掉後面幾層之後重新訓練模型的參數，接著將部分網路層（如卷積層）之參數遷移至目標任務中。

圖 2.1[7]為遷移式學習的基本流程，針對目標資料可選擇僅訓練輸出層以降低過擬合風險，或在資料充足情況下對整體網路進行微調（fine-tuning），以進一步提升預測準確率。

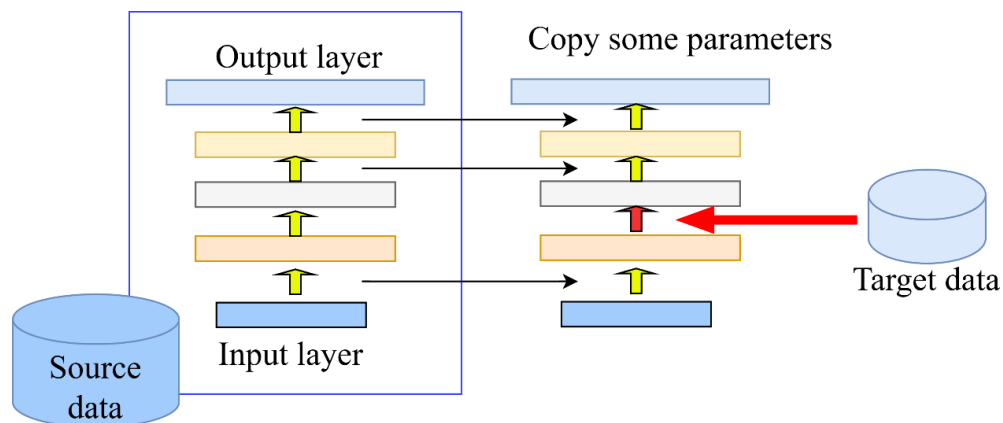


圖 2.1 Layer Transfer [7]

第 3 章 研究內容與方法

本研究以糖尿病視網膜辨識為核心，開發一套基於網頁平台之輔助診斷系統。該系統提供醫師便利的操作介面，使用者可透過網頁介面上傳視網膜影像，系統將影像傳送至已訓練完成之辨識模型進行分類分析，模型可辨識五種視網膜病變等級，包括無視網膜病變(No DR)、輕度非增殖性視網膜病變 (Mild NPDR)、中度非增殖性視網膜病變 (Moderate NPDR)、重度非增殖性視網膜病變 (Server NPDR) 及增殖性視網膜病變 (PDR)。分析完成後，系統將辨識結果回傳並於網頁上顯示對應之 DR 分級，協助臨床醫師快速判斷病變程度，作為診斷與後續治療決策之參考依據。整體架構如圖 3.1 所示。

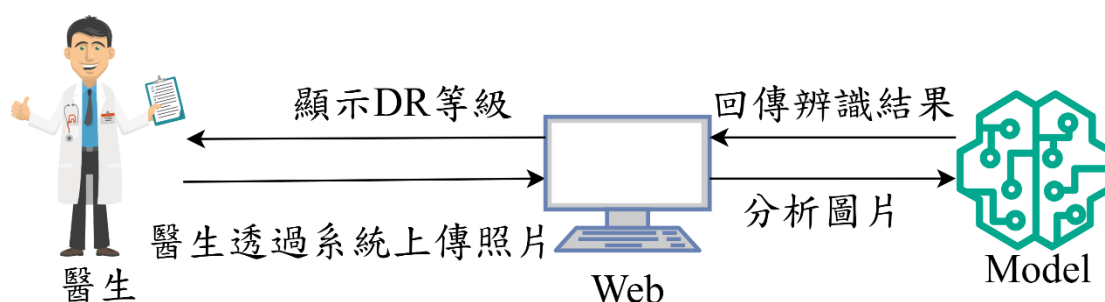


圖 3.1 系統架構圖

3.1 訓練架構

透過資料分割與擴增提升模型泛化能力，並以 CNN、VGG16、ResNet50 進行糖尿病視網膜病變五分類訓練。最後以準確率、損失函數與混淆矩陣等指標評估模型效能與實用性，如圖 3.2 所示。

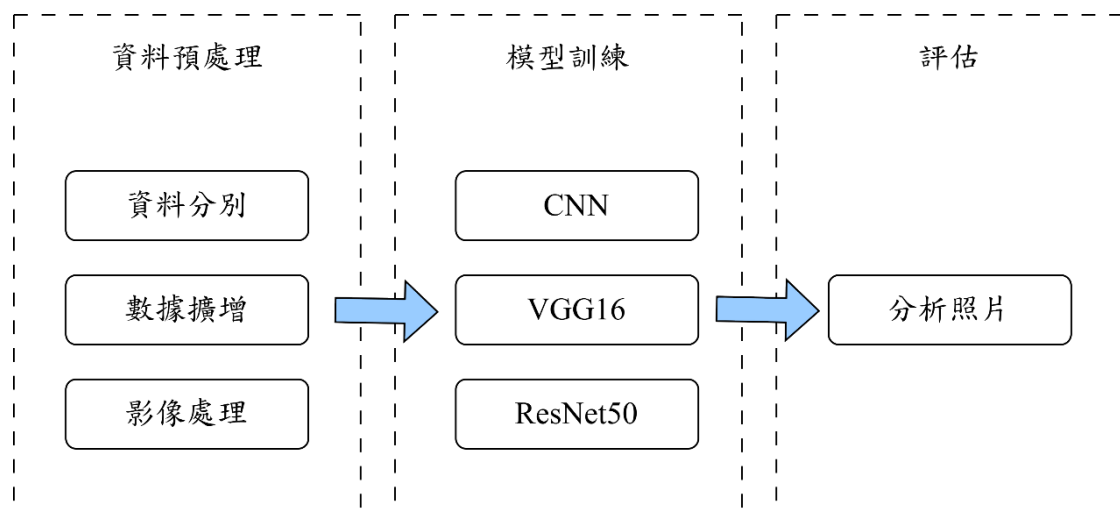


圖 3.2 糖尿病視網膜辨識系統架構

整理架構分為以下三大主軸：

(1) 資料預處理：

首先進行資料分割（如訓練集與驗證集的劃分），接著進行資料增強以平衡資料分布與提升模型泛化能力，最後針對影像進行處理（影像裁切、縮放、標準化等），以符合模型輸入需求。

(2) 模型訓練：

本研究採用多種深度學習架構進行模型訓練與分類任務，包括基本的卷積神經網路(CNN)、VGG16 及 ResNet50 三種架構。糖尿病視網膜影像為輸入模型，輸出為五級糖尿病視網膜病變等級，包括 No DR、Mild NPDR、Moderate NPDR、Severe NPDR 與 PDR，完成病變等級之分類辨識。

(3) 模型評估：

模型訓練完成後，將預測結果與真實標籤進行比對，並採用多項指標進行效能評估，包括準確率（Accuracy）、損失函數值（Loss）與混淆矩陣（Confusion Matrix）等。藉由評估結果，檢視模型效能與實用性。

3.2 流程說明

內容為研究流程圖，展示了糖尿病視網膜病變辨識系統的開發過程，如圖 3.3 所示。

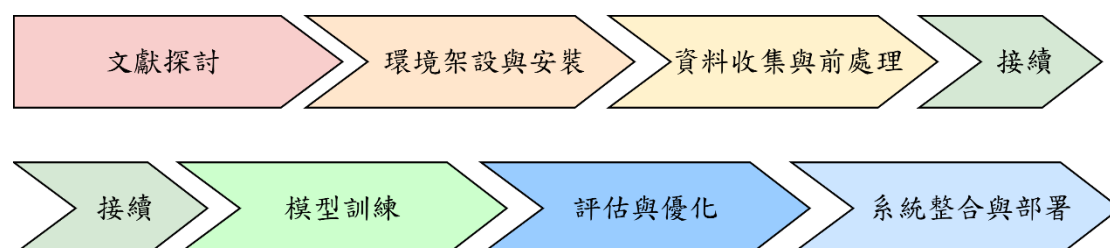


圖 3.3 研究流程

流程說明分為以下六個步驟：

(1) 文獻探討：

此階段主要探討糖尿病視網膜病變的背景、相關議題與自動化檢測的研究議題，並整理相關技術文獻，明確研究主題。

(2) 環境架設與安裝：

建立開發環境，包括軟體資源規劃 Google Colab、TensorFlow、Visual Studio Code。

(3) 資料收集與前處理：

使用公開資料集（IDRiD、APTOS 2019 等），進行影像標準化、資料增強等處理，以提升模型訓練效果。

(4) 模型訓練：

採用深度學習技術（CNN、VGG16、ResNet50）進行模型訓練，並透過遷移學習優化模型性能。

(5) 評估與優化：

使用準確率、混淆矩陣等指標評估模型效能，並進行超參數調整以提升辨識準確率。

(6) 系統整合與部署：

將訓練完成的模型整合至雲端平台，開發使用者介面，提供使用者便利的操作介面。

3.3 軟硬體資源規劃

電腦硬體方面，使用搭載 Intel Core i7-12700H 處理器、16GB 記憶體容量與 NVIDIA GeForce RTX 4050 GPU 的 Windows 11 筆記型電腦，作為本地測試與備援平台，提供穩定效能支援原型設計與開發。軟體配置方面：作業系統為 Windows 11 家用版、程式撰寫所使用的平台為 Visual Studio Code 1.100.1、所使用的程式語言為 Python 3.8.20、深度學習為 Tensorflow 2.3.0。如圖 3.1 所示。

表 3.1 本研究實驗軟硬體環境

實驗環境配置	規格
作業系統	Windows 11 家用版
處理器	Intel(R) Core(TM) i7-12700H
外部顯卡	NVIDIA GeForce RTX 4050 laptop GPU
標準記憶體	16GB
程式撰寫平台	Visual Studio Code 1.100.1
程式語言	Python 3.8.20
深度學習框架	Tensorflow 2.3.0

3.4 模型訓練

本研究針對糖尿病視網膜病變影像進行多分類任務，並以遷移學習(Transfer Learning)方法為核心，採用 CNN、VGG16、ResNet50 模型進行比較。

3.4.1 CNN

將原始眼底影像資料載入系統，並透過 ImageDataGenerator 工具進行前處理，包含影像正規化 (Normalization) 與批次大小設定 (Batch size = 32)。

進行資料增強處理 (Data Augmentation)，如水平翻轉 (horizontal flip)，藉此增加樣本多樣性與模型泛化能力。圖片將尺寸統一 (224×224) 後在 OpenCV 中進行高斯模糊處理 (Gaussian Blur) 如圖 3.4。圖片在進行銳化後會使病變狀態更明顯，增加訓練效果。

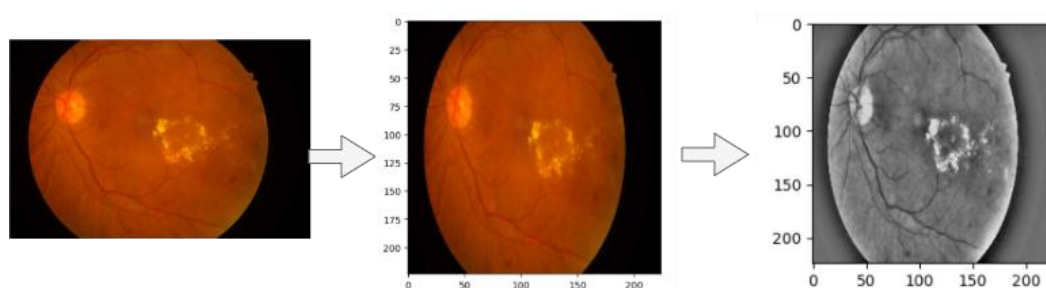


圖 3.4 高斯處理變化圖

處理後的資料集依比例劃分為訓練集 (70%)、驗證集 (15%) 及測試集 (15%)，作為模型的學習與調整依據。

訓練期間同時記錄訓練集與驗證集的損失 (Loss) 與準確率 (Accuracy) 變化，繪製學習曲線以觀察模型學習狀況；並對驗證資料計算並視覺化混淆矩陣 (Confusion Matrix)，分析各類別預測準確性與錯誤分布。

此外，測試集資料（Test set）並不包含標籤（No labels），因此不參與模型的訓練與驗證流程。測試集僅作為最終模型推論之輸入資料，並產生預測結果。該結果將輸出產生.csv 檔案，作為模型應用於真實未見資料時的性能評估依據。

(1) 模型架構

先將模型的輸入為大小為（224,224,3）的眼底圖像，採用 CNN 模型，三層卷積、雙層全連接結構，配合 Batch Normalization 和 Dropout，既能提取多層次特徵，又能控制過擬合。架構如表 3.2。

表 3.2 CNN 模型架構

層名稱 (Layer)	描述 (Description)
Conv2D (8, 3×3, valid, ReLU)	核心大小 3×3，提取初級特徵並輸出 8 張特徵圖，輸出尺寸從 224×224 變為 222×222，並使用 ReLU 激活函數，將負值置零，引入非線性。
MaxPooling2D(2×2)	取每個 2×2 區域的最大值將特徵圖空間尺寸縮成 111×111，減少參數量與計算量，強化空間不變性。
BatchNormalization()	正規化該層輸出，穩定分佈（均值≈0，方差≈1），用於加速收斂並減少對學習率的敏感度。
Conv2D (16, 3×3, valid, ReLU)	深度加倍到 16 張特徵圖尺寸、從 111×111 → 109×109。
MaxPooling2D(2×2)	109×109 → 54×54。
BatchNormalization()	再次穩定激活分佈。

(待續)

表 3.2 CNN 模型架構(續)

Conv2D (32, 4×4, valid, ReLU)	使用 4×4 卷積核以抓取更大範圍的特徵 輸出 32 張特徵圖，尺寸從 54×54 縮小 成 51×51。
MaxPooling2D(2×2)	尺寸從 51×51 縮小為 25×25。
BatchNormalization()	最終的特徵圖正規化。
Flatten()	全連接層前處理，將三維張量攤平成一 維向量。
Dense(32, ReLU)	32 個神經元，學習扁平化特徵之間的複 雜組合。
Dropout(0.15)	訓練期間隨機丟棄 15%的節點，防止過 擬合。
Dense (2,activation='softmax')	對應兩個分類，Softmax 輸出互斥的概率 分佈，總和為 1。

(2) 編譯模型：

使用 `model.compile()` 編譯模型，其中會設定最佳化器 (Optimizer)、損失函數 (Loss Function) 及和評估指標 (Metrics)，再進行模型訓練。

編譯整體步驟如下三點：

1) 優化器：

採用 Adam，決定模型如何依據損失函數去更新權重，學習率設為 $1e-5$ ，較小的步伐有助於穩定訓練。

2) 損失函數：

使用 `binary_crossentropy`，適用於二分類問題，計算預測值與真實值之間的差距。

3) 評估指標：

使用 accuracy，用來監控模型性能的數值，在每個 epoch 結束時計算二分類準確率。

(3) 訓練模型：

編譯完模型後，要開始訓練模型，使用 model.fit()，進行模型訓練，使用 history 物件會儲存每個 epoch 的 loss、accuracy、val_loss、val_accuracy，可用來後續繪圖與分析訓練曲線；將 epochs 設置 30，完整訓練 30 次資料；validation_data=val_batches 在每個 epoch 結束後，用驗證集評估模型性能，檢測是否有過擬合。

3.4.2 VGG16

資料預處理方式與 CNN 相同，資料集一樣依比例劃分為訓練集（70%）、驗證集（15%）及測試集（15%），作為模型的學習與調整依據。

(1) 模型架構：

表 3.3 為 VGG16 模型架構。

表 3.3 VGG16 模型架構

層名稱 (Layer)	描述 (Description)
model.add(Vgg16)	加入預訓練。
Dropout(0.25)	隨機丟棄 25%的節點，防止過擬合。
Flatten()	展平成一維向量，以便接在全連階層上。
Dense(5,activation='softmax')	對應五個分類。

載入 VGG16 預訓練，去掉頂層全連階層部分，只保留卷基層和池化層作為特徵提取氣，並凍結預訓練所有層，鎖定所有層的權重，讓它們在訓練過程中保持不變。

(2) 編譯模型：

使用 `model.compile()` 編譯模型，其中會設定最佳化器 (Optimizer)、損失函數 (Loss Function) 及和評估指標 (Metrics)，再進行模型訓練。

編譯整體步驟依序分為五點：

1) 優化器：

採用 Adam，決定模型如何依據損失函數去更新權重，學習率設為 $1e-5$ ，較小的步伐有助於穩定訓練。

2) 損失函數：

使用 `binary_crossentropy`，適用於二分類問題，計算預測值與真實值之間的差距。

3) 評估指標：

使用 `accuracy`，用來監控模型性能的數值，在每個 epoch 結束時計算二分類準確率。

4) `class_weight` 參數：

計算類別權重以對應數據不平衡，自動計算權重，使少數類別獲得更高的權重，緩解不平衡的問題。

5) `EarlyStopping` 參數：

在訓練中，驗證準確率沒有提升的情況，會觸發早停，使在訓練結束後，呈現驗證率最高時的權重，確保模型最優狀態。

(3) 訓練模型：

編譯完模型後，要開始訓練模型，使用 `model.fit()`，進行模型訓練，使用 `history` 物件會儲存每個 `epoch` 的 `loss`、`accuracy`、`val_loss`、`val_accuracy`，可用來後續繪圖與分析訓練曲線；將 `epochs` 設置 50，完整訓練 50 次資料；`validation_data=val_batches` 在每個 `epoch` 結束後，用驗證集評估模型性能，檢測是否有過擬合。

3.4.3 ResNet50

本研究針對糖尿病視網膜病變影像進行多分類任務，並以遷移學習（Transfer Learning）方法為核心，採用 ResNet50 預訓練模型進行特徵提取與訓練，流程如圖 3.5 所示。

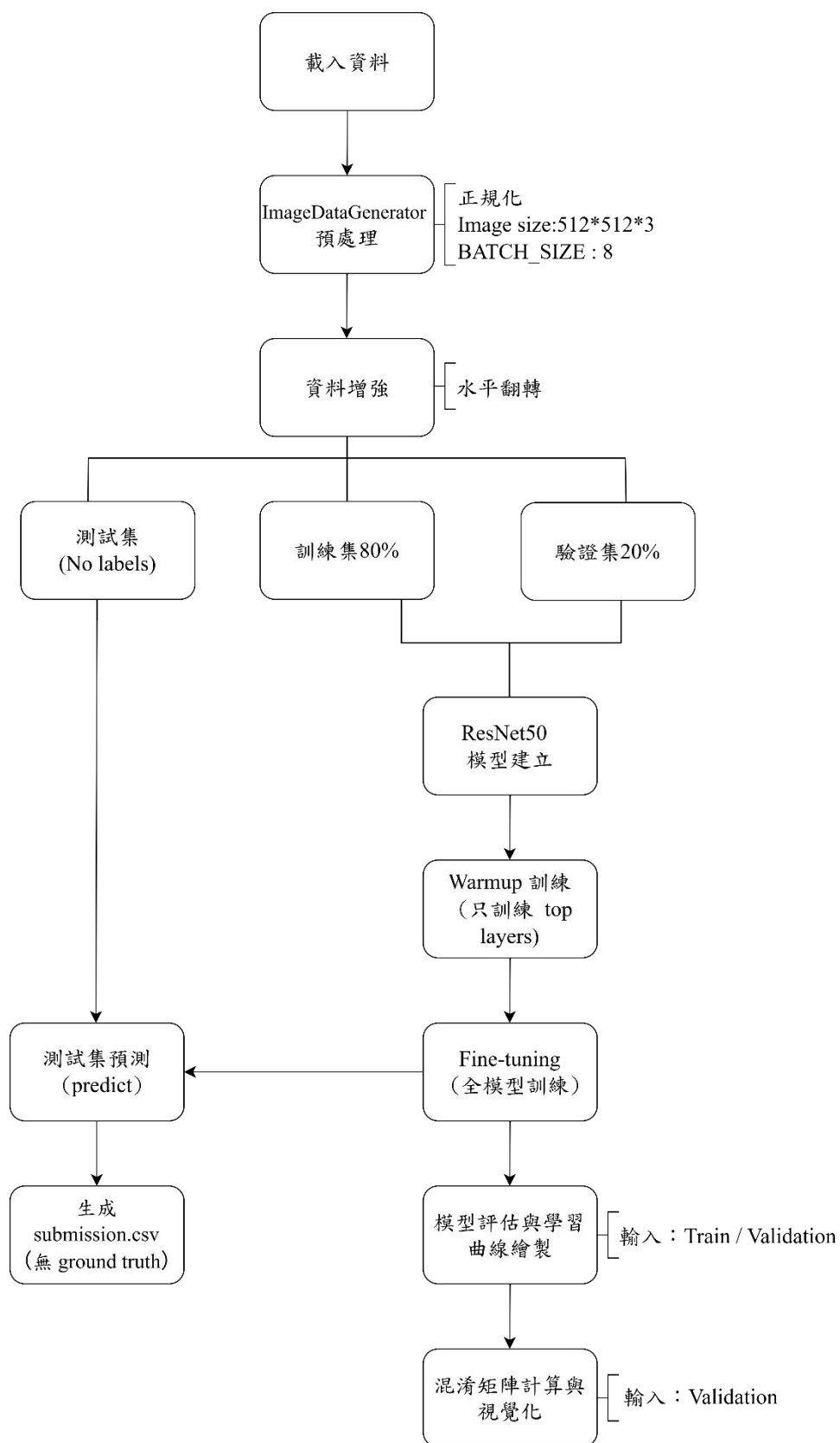


圖 3.5 ResNet50 模型實作流程圖

98+將原始眼底影像資料載入系統，並透過 ImageDataGenerator 工具進行前處理，包含影像正規化（Normalization）、尺寸統一（ $512 \times 512 \times 3$ ）與批次大小設定（Batch size = 8）。

進行資料增強處理（Data Augmentation），如水平翻轉（horizontal flip），藉此增加樣本多樣性與模型泛化能力。處理後的資料依 8:2 比例劃分為訓練集（80%）與驗證集（20%），作為模型的學習與調整依據。

在模型建構階段，採用已在 ImageNet 上預訓練（pre-trained）的 ResNet50 架構，並凍結底層特徵提取層，僅對頂層進行 Warm-up 訓練，以適應目標資料的特性。在 Warm-up 階段結束後，啟用全模型進行 Fine-tuning 微調訓練，進一步提升分類效能。此遷移學習策略不僅加速收斂速度，也可藉由保留通用特徵，提升在小樣本視網膜病變資料上的表現。

訓練期間同時記錄訓練集與驗證集的損失（Loss）與準確率（Accuracy）變化，繪製學習曲線以觀察模型學習狀況；並對驗證資料計算並視覺化混淆矩陣（Confusion Matrix），分析各類別預測準確性與錯誤分布。

此外，測試集資料（Test set）並不包含標籤（No labels），因此不參與模型的訓練與驗證流程。測試集僅作為最終模型推論之輸入資料，並產生預測結果。該結果將輸出為 submission.csv 檔案，作為模型應用於真實未見資料時的性能評估依據。

先將模型的輸入為大小為(512,512,3)的 RGB 彩色眼底圖像，採用 ResNet50 模型，並載入 ImageNet 預訓練權重。

在 ResNet50 後方接續一層 GlobalAveragePooling2D，該層負責將最後一層特徵圖轉換為一個一維向量，進行特徵壓縮，同時保留整體語意資訊。

接著加入第一層 Dropout(0.5)，隨機丟棄 50% 的神經元。緊接著為一層 Dense 全連接層，包含 2048 個神經元並使用 ReLU 激活函數，以強化模型的非線性特徵抽取能力，再次加入 Dropout(0.5)作為正規化措施。

最後，模型輸出層為一層 Dense(5)全連接層，對應糖尿病視網膜病變的五個等級，並採用 softmax 激活函數將模型輸出轉為機率分布，用以進行多分類任務的最終預測。ResNet50 模型架構，如表 3.4 所示。

表 3.4 ResNet50 模型架構

層名稱 (Layer)	描述 (Description)
Input	輸入尺寸為(512,512)的 RGB 眼底圖像。
GlobalAveragePooling2D	將特徵圖進行全局平均池化，壓縮為一維向量。
Dropout(0.5)	丟棄 50% 的神經元。
Dense(2048, ReLU)	全連接層，具 2048 個神經元，ReLU 激活函數。
Dropout(0.5)	丟棄 50% 的神經元。
Dense(5, softmax)	輸出層，對應 5 個 DR 分類等級，採用 softmax 機率分布輸出。

(1) Warm-Up：

模型初始化後，將所有層進行凍結，以保留透過 ImageNet 預訓練所學得之通用特徵。接著，選擇性地解凍模型的最後五層，允許其於 warm-up 階段進行微調，可在保有預訓練特徵的同時，降低訓練初期模型不穩定的風險，提升整體訓練效能。

本階段可分為以下四個步驟進行：

1) 凍結所有層與部分解凍：

模型初始化後，先將所有層凍結，以保留預訓練的 ImageNet 特徵。接著解凍最後五層，允許其在 warm-up 階段微調，以適應眼底圖像特徵。

2) 編譯模型：

使用 `model.compile()` 編譯模型，採用 Adam 優化器，學習率設為 $1e-3$ ，損失函數為 `categorical_crossentropy`，評估指標為 `accuracy`。

3) 訓練設定：

訓練批次 `batch_size` 大小設定為 8，並使用 `ImageDataGenerator` 分出訓練與驗證資料。warm-up 階段訓練輪數為 2 (`warmup_epochs`)，每輪使用完整資料集進行訓練與驗證。

4) 模型訓練：

執行 `model.fit()`，僅微調模型最後幾層分類頭，以穩定收斂並為後續 fine-tuning 做準備。

(2) Fine-Tuning :

完成 warm-up 階段後，進入模型的完整微調 (fine-tuning) 階段，進一步提升模型的辨識能力與泛化表現。將整個模型的所有層皆設為可訓練狀態，以允許參數全面調整，使模型更能適應視網膜影像的特徵差異。

為避免過擬合與控制學習率過高所帶來的不穩定性，此階段導入 EarlyStopping、ReduceLROnPlateau，透過完整的訓練資料集進行 fine-tuning，其訓練步數與 warm-up 階段相同，總訓練週期 epoch 設定為 20，確保模型充分學習資料中的細節，強化其分類效能。

分為以下三個步驟進行：

1) 訓練回調函數設定：

避免過擬合與學習率過高問題，使用 EarlyStopping 監控 val_loss，若連續數輪未改善即中止訓練。

使用 ReduceLROnPlateau，當 val_loss 停滯時自動降低學習率。

2) 模型編譯：

使用 model.compile() 編譯模型，使用 Adam 優化器，學習率設為 learning_rate (預設 1e-4)。損失函數設為 categorical_crossentropy。

3) 模型訓練：

使用完整訓練資料進行 fine-tuning，步數與 warm-up 相同，並設定總訓練 epoch 為 20。

第 4 章 研究成果

為了輔助眼科醫師更快速、準確地診斷患者，本研究開發了一套簡易的視網膜病變辨識系統。

4.1 系統架構

此系統的主要功能包括患者資料管理、診斷結果儲存、風險分析及回診提醒通知，透過以上功能，醫師不僅能集中管理所有患者的檢查與診斷紀錄，還可即時掌握需要回診的對象，提升臨床效率與醫療品質。系統架構如圖 4.1。

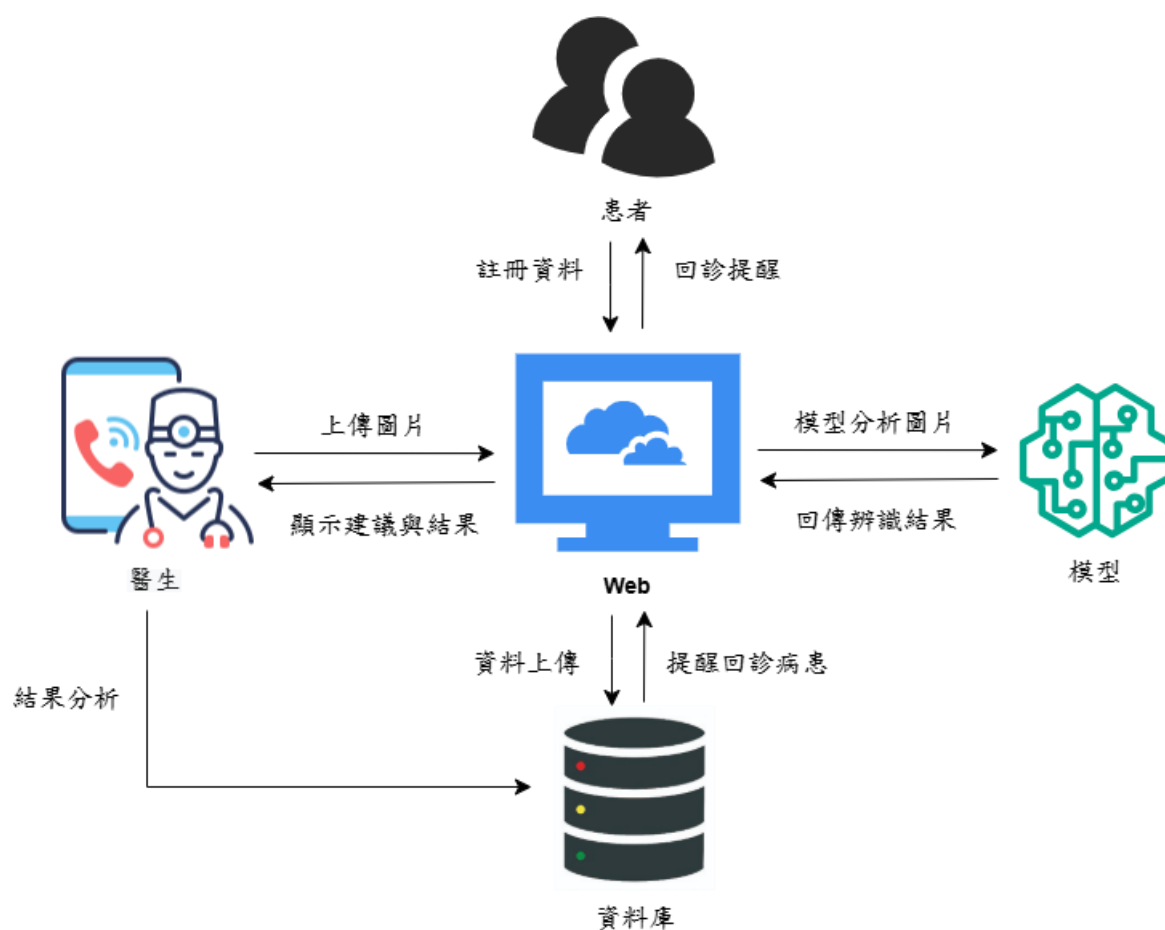


圖 4.1 系統架構圖

4.1.1 系統流程設計

當患者就醫時，詢問病情後由醫師為患者拍攝視網膜照片，將眼底影像上傳至辨識系統，醫生將系統分析輸出的結果結合自身經驗給出最後診斷報告，將診斷內容及患者資料存入資料庫中，回診時間到時會提醒繪者回診。流程如圖 4.2。

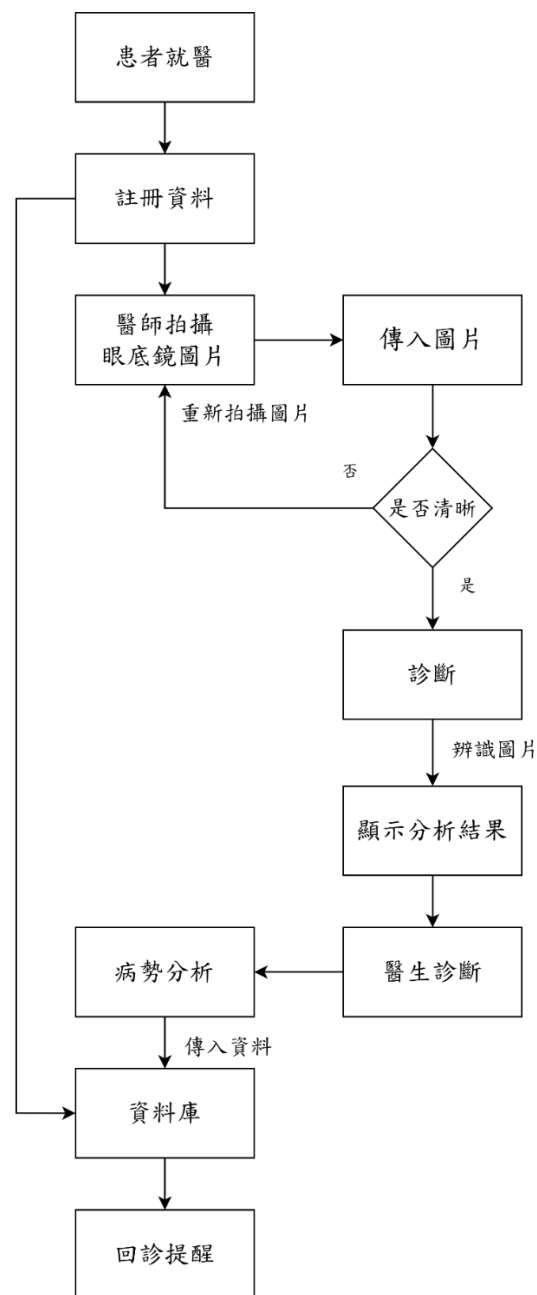


圖 4.2 系統流程設計圖

4.1.2 互動介面設計

目前僅展示辨識功能，實際介面為二期內容。介面如下圖 4.3 所示。



圖 4.3 GUI 介面圖

點選 Browse 鍵，選擇並傳入要進行辨識之視網膜圖片後點選 Detect 按鈕開始辨識分析影像，先進行 Stage1 有無 DR 分析，如圖 4.4。

```
1/1 ————— 0s 182ms/step  
[DEBUG] Stage 1 prediction (DR / No DR): [[0. 1.]]  
[DEBUG] Stage 1 result: 1
```

圖 4.4 Stage1 影像分析

如分析結果為 No DR，介面即出現等級與分析結果，如圖 4.5。

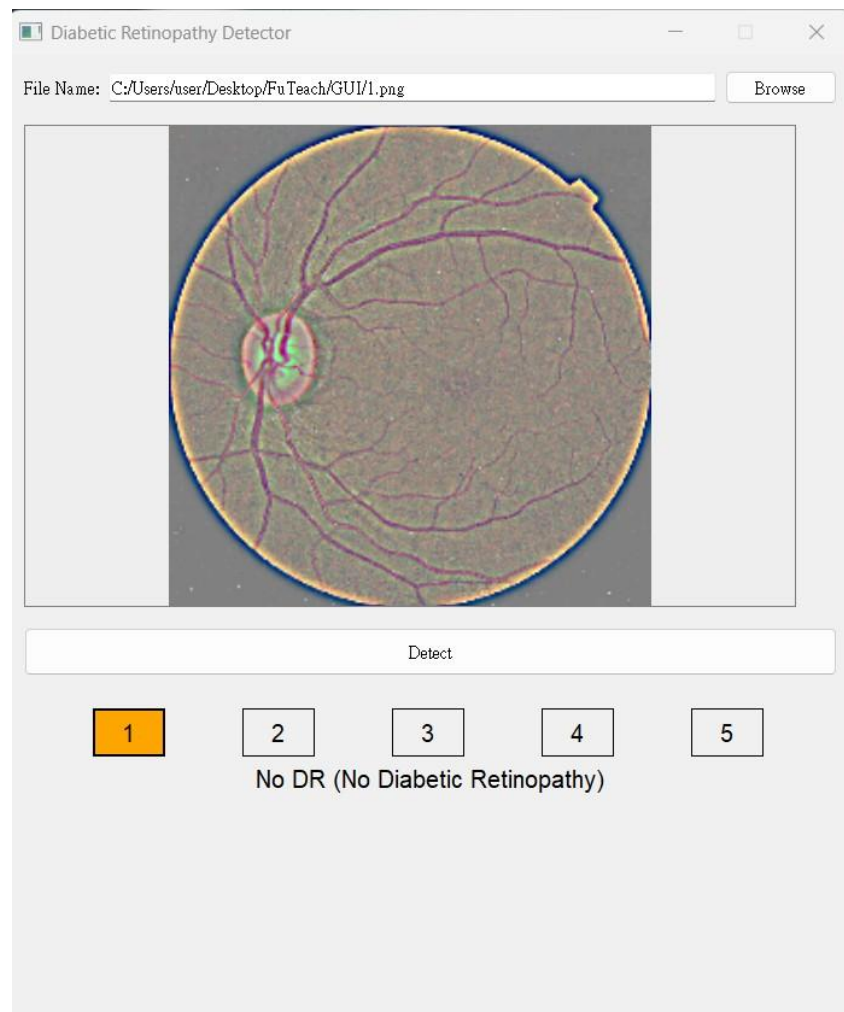


圖 4.5 No DR 辨識結果

當點選 Detect 按鈕進行影像辨識，Stage1 分析為 DR，就會啟動 Stage2 進行辨識分析，如圖 4.6。

```
1/1 ————— 0s 28ms/step
[DEBUG] Stage 1 prediction (DR / No DR): [[1. 0.]]
[DEBUG] Stage 1 result: 0
1/1 ————— 0s 97ms/step
[DEBUG] Stage 2 prediction (DR severity): [[0. 1. 0. 0.]]
[DEBUG] Stage 2 result (mapped): 2
```

圖 4.6 Stage1 與 Stage2 影像分析過程圖

分析完，介面即出現等級與分析結果，如圖 4.7。



圖 4.7 Moderate DR 辨識結果

4.2 辨識成果

展示 CNN、VGG16 及 ResNet50 模型訓練後的準確率與為每個模型分別計算加權平均 Precision、Recall 及 F1-score，以綜合考量多數與少數類別的預測表現。

4.2.1 CNN

考慮到 No DR 的訓練及數量已經相當於四個 DR 類別的訓練及總和，因此我們將辨識分為兩個階段，第一階段為二分法，確認有無 DR，如果有 DR 則進入下一階段四分法，進行更深層的 DR 等級分類。

二分法的訓練結果，模型在最後表現訓練準確率高達 0.98，說明模型在訓練數據上幾乎收斂，如圖 4.8。

```
Epoch 28/30  
81/81 ————— 7s 80ms/step - accuracy: 0.9844 - loss: 0.0602 - val_accuracy: 0.9418 - val_loss: 0.2124  
Epoch 29/30  
81/81 ————— 7s 80ms/step - accuracy: 0.9883 - loss: 0.0567 - val_accuracy: 0.9382 - val_loss: 0.2124  
Epoch 30/30  
81/81 ————— 7s 81ms/step - accuracy: 0.9887 - loss: 0.0530 - val_accuracy: 0.9400 - val_loss: 0.2123
```

圖 4.8 二分法準確率

驗證準確率為 0.94，驗證損失率高於訓練損失率，模型可能出現過擬合之情況，而對於新樣本的泛化能力還有提升空間，結果曲線如圖 4.9。

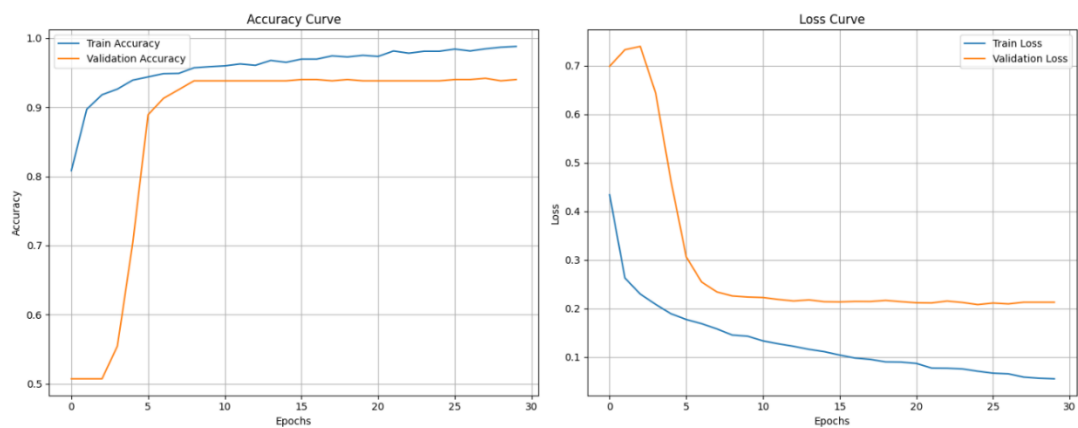


圖 4.9 CNN 二分法 Accuracy 與 Loss 曲線圖

由二分法混淆矩陣圖圖 4.10 中可觀察到，模型在分類「No DR」與「DR」的表現非常平均，準確度高。

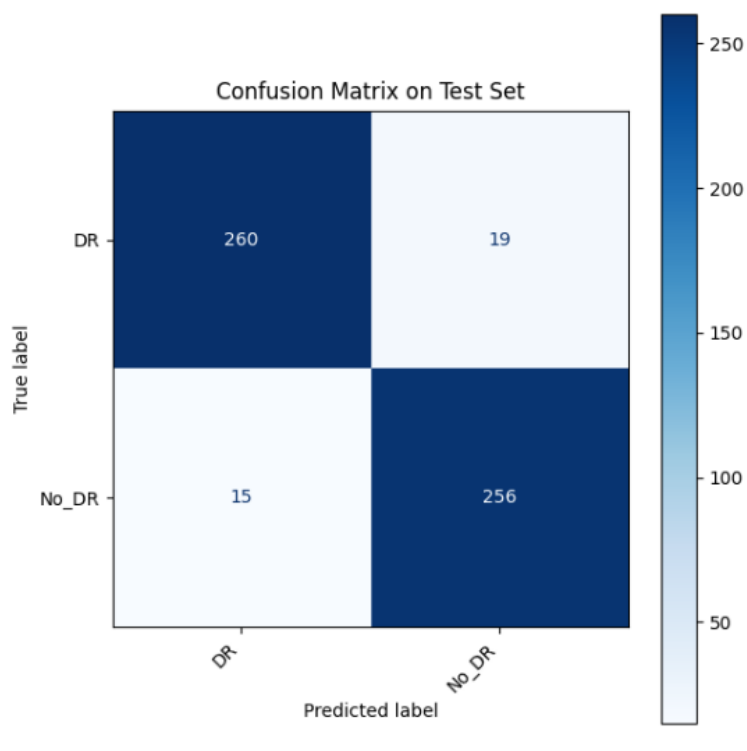


圖 4.10 CNN 混淆矩陣

表 4.1 為二分法常見評價指標。

表 4.1 CNN 二分法指標

類別	Precision	Recall	F1-score	support
DR	0.9455	0.9319	0.9386	279
No DR	0.9309	0.9446	0.9377	271

模型在最後表現訓練準確率高達 0.98，說明模型在訓練數據上幾乎收斂，四分法的訓練結果為圖 4.11。

```

Epoch 33/35
41/41 ----- 5s 112ms/step - accuracy: 0.9798 - loss: 0.1045 - val_accuracy: 0.5771 - val_loss: 1.2618
Epoch 34/35
41/41 ----- 5s 113ms/step - accuracy: 0.9826 - loss: 0.0961 - val_accuracy: 0.5591 - val_loss: 1.2874
Epoch 35/35
41/41 ----- 5s 119ms/step - accuracy: 0.9849 - loss: 0.1113 - val_accuracy: 0.5663 - val_loss: 1.2978

```

圖 4.11 四分法準確率

驗證準確率為 0.56，驗證損失率高達 1.2 多，並隨著訓練損失率還有上升趨勢，表示模型在新數據上的泛化能力很弱，而驗證損失率不降反升如圖 4.12，是過擬合典型特徵。

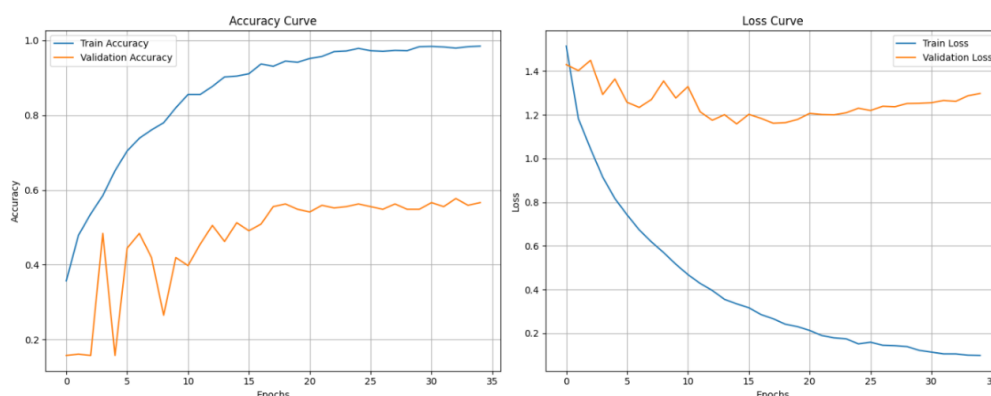


圖 4.12 CNN 四分法 Accuracy 與 Loss 曲線圖

四分法混淆矩陣圖如圖 4.13。

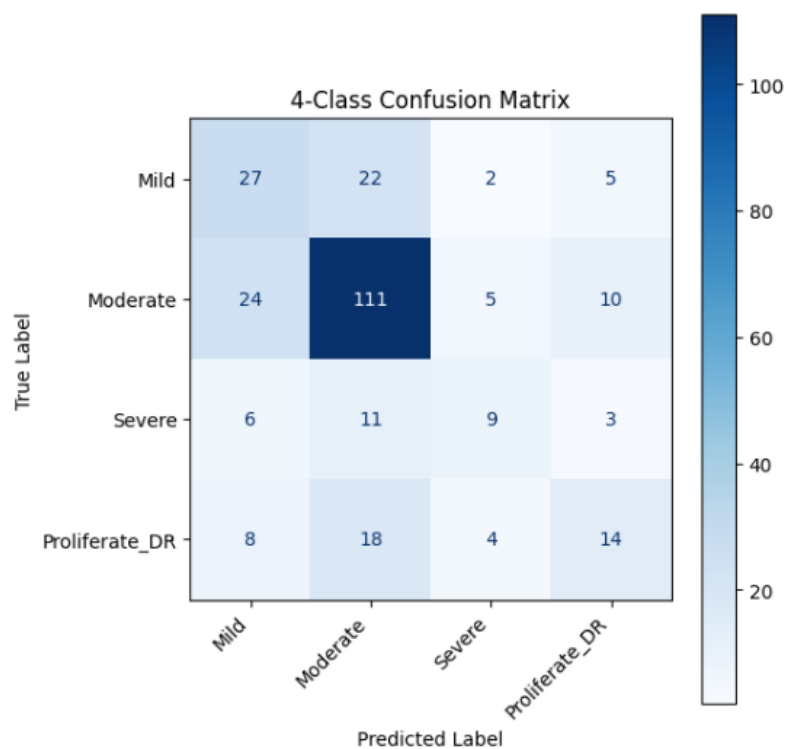


圖 4.13 CNN 四分法混淆矩

四分法評價指標中，召回率得知模型識別 Moderate 類別為 0.74 最為準確，其他類別的召回率較低，容易混淆，如下表 4.2。

表 4.2 CNN 四分法指標

類別	Precision	Recall	F1-score	support
Mild	0.4154	0.4821	0.4463	56
Moderate	0.6852	0.7400	0.7115	150
Severe	0.4500	0.3103	0.3673	29
Proliferate_DR	0.4375	0.3182	0.3684	44

4.2.2 VGG16

因設定 EarlyStopping 參數，在訓練過程中連續 10 個 epoch 驗證準確率都沒有超過 0.90 停止訓練，第 36 個 epoch 為最佳模型，驗證準確率達到最高點 0.90，驗證損失最低為 0.23。圖 4.14 所示。

```
Epoch 36/50  
37/37 [=====] - 12s 329ms/step - loss: 0.3199 - binary_accuracy: 0.8883 - val_loss: 0.2356 - val_binary_accuracy: 0.9082  
Epoch 37/50  
37/37 [=====] - 12s 329ms/step - loss: 0.3219 - binary_accuracy: 0.8895 - val_loss: 0.2789 - val_binary_accuracy: 0.8778  
Epoch 38/50  
37/37 [=====] - 12s 329ms/step - loss: 0.3349 - binary_accuracy: 0.8873 - val_loss: 0.2417 - val_binary_accuracy: 0.8918
```

圖 4.14 VGG16 準確率

訓練和驗證損失都在穩定下降，準確率也持續上升趨於平穩，模型收斂情況良好，由圖 4.15 可見。

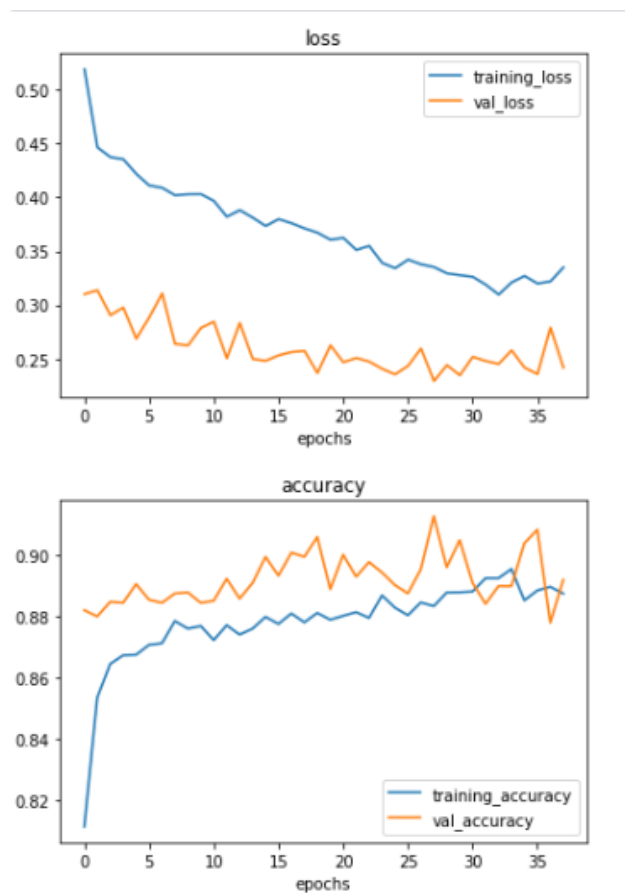


圖 4.15 VGG16 Accuracy 與 Loss 曲線圖

VGG16 之混淆矩陣圖，圖片顯示数据不平衡、Severe 與 Proliferate_DR 易被 Moderate 或 Mild 混淆，如圖 4.16 所示。

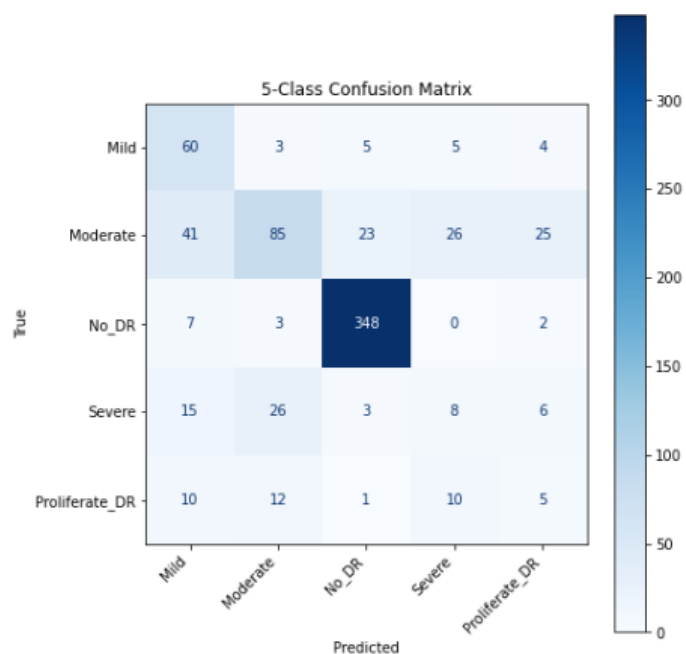


圖 4.16 VGG16 混淆矩陣

評價指標數據顯示模型對 No_DR 樣本辨識準確且漏檢少，而其他類別辨識能力較低，可再重點增強區分能力，如下表 4.3。

表 4.3 VGG16 指標

類別	Precision	Recall	F1-score	support
No_DR	0.9158	0.9667	0.9405	360
Mild	0.4511	0.7792	0.5714	77
Moderate	0.6589	0.4250	0.5167	200
Severe	0.1633	0.1379	0.1495	58
Proliferate_DR	0.1190	0.1316	0.1250	38

4.2.3 ResNet50

本研究於模型微調(Fine-tuning)階段共訓練 20 回合(Epoch)，透過 EarlyStopping 與 ReduceLROnPlateau 機制，自動調整訓練長度與學習率，有效提升模型穩定性與泛化能力。

整體訓練過程中，模型在第 1 回合的驗證準確率為 0.4588，隨著訓練次數增加，準確率逐步上升，至第 5 回合突破 0.72，第 10 回合達到 0.8283，並於第 15 回合達到最高驗證準確率為 0.8393，對應驗證損失最低值 0.1580。此後雖有小幅波動，但整體驗證準確率皆維持在 82%~84% 之間，如圖 4.17 所示。

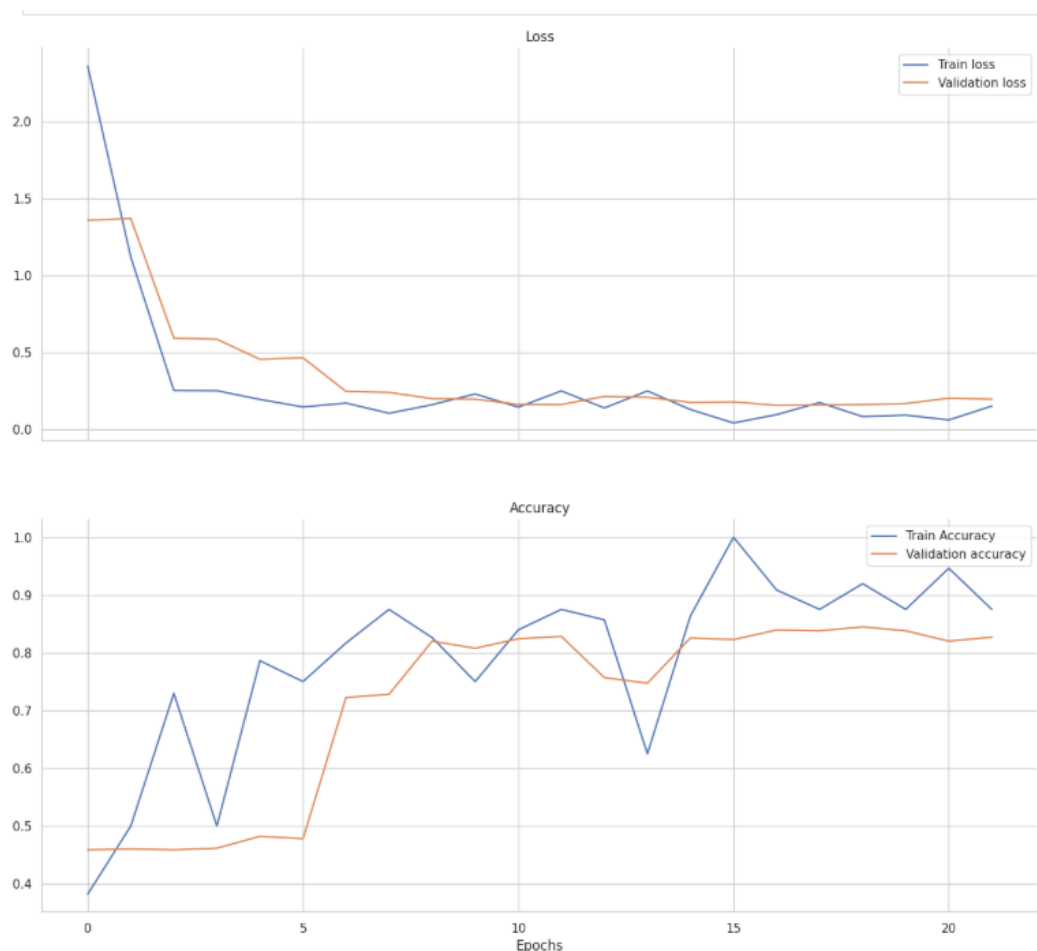


圖 4.17 ResNet50 Accuracy 與 Loss 曲線圖

同時，在第 13 與第 18 回合模型偵測到驗證損失停滯後，自動降低學習率（由 $1e-4$ 降為 $5e-5$ 再降至 $2.5e-5$ ），避免過度調整造成的不穩定。最終模型自動恢復至第 15 回合最佳權重，做為後續測試與部署使用，如圖 4.18 所示。

```
Epoch 13/20
366/366 — 0s 758ms/step - accuracy: 0.8716 - loss: 0.1267
Epoch 13: ReduceLROnPlateau reducing learning rate to 4.999999873689376e-05.
366/366 — 353s 965ms/step - accuracy: 0.8716 - loss: 0.1267 - val_accuracy: 0.8255 - val_loss: 0.1771 -
learning_rate: 1.0000e-04
Epoch 14/20
366/366 — 75s 205ms/step - accuracy: 1.0000 - loss: 0.0436 - val_accuracy: 0.8228 - val_loss: 0.1801 -
learning_rate: 5.0000e-05
Epoch 15/20
366/366 — 359s 981ms/step - accuracy: 0.9069 - loss: 0.1020 - val_accuracy: 0.8393 - val_loss: 0.1580 -
learning_rate: 5.0000e-05
Epoch 16/20
366/366 — 76s 207ms/step - accuracy: 0.8750 - loss: 0.1761 - val_accuracy: 0.8379 - val_loss: 0.1598 -
learning_rate: 5.0000e-05
Epoch 17/20
366/366 — 352s 961ms/step - accuracy: 0.9247 - loss: 0.0824 - val_accuracy: 0.8448 - val_loss: 0.1631 -
learning_rate: 5.0000e-05
Epoch 18/20
1/366 — 1:11 197ms/step - accuracy: 0.8750 - loss: 0.0947
Epoch 18: ReduceLROnPlateau reducing learning rate to 2.499999936844688e-05.
366/366 — 76s 207ms/step - accuracy: 0.8750 - loss: 0.0947 - val_accuracy: 0.8379 - val_loss: 0.1694 -
learning_rate: 5.0000e-05
```

圖 4.18 ResNet50 損失函數

模型在分類「0 - No DR」與「2 - Moderate」的表現非常準確，其中 No DR 的準確率高達 99%，Moderate 亦達到 93%。對於「1 - Mild」與「3 - Severe」雖有一定的辨識能力，但部分樣本容易被混淆為相鄰等級，如 Mild 有 17% 被判為 Moderate，而 Severe 則有 16% 誤判為 Moderate、10% 為 Proliferative DR，如圖 4.19 所示。

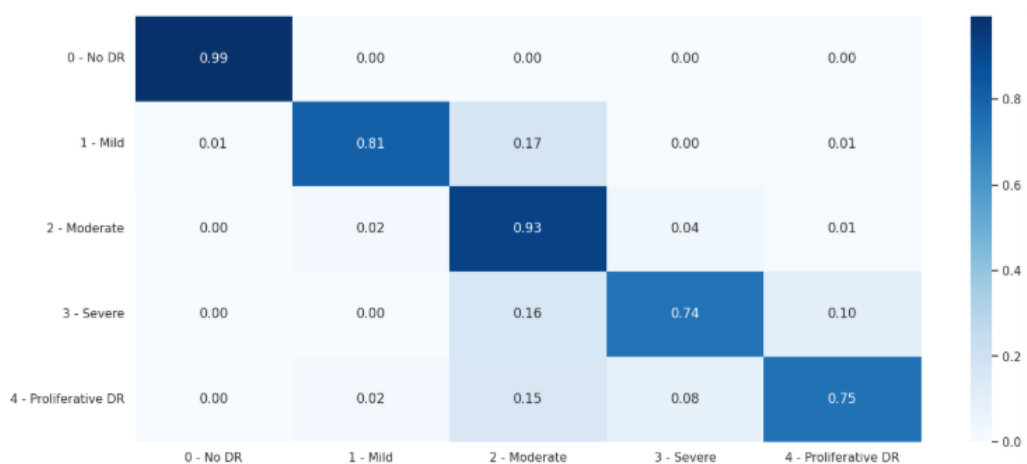


圖 4.19 ResNet50 標準化混淆矩陣

從 train.csv 資料集中採用分層抽樣 (stratified sampling) 方法，抽取 20% 資料作為測試集。該測試集包含 733 張眼底影像，涵蓋糖尿病視網膜病變 (DR) 五個等級。根據 support 欄位統計，測試集類別分布不均，類別 0 (無病變 No DR, 361 張) 與類別 2 (中度病變 Moderate, 200 張) 為主要類別，類別 1 (輕度病變 Mild, 74 張)、類別 3 (重度病變 Severe, 39 張) 及類別 4 (增殖型病變 Proliferative DR, 59 張) 則屬於少數類別，其中類別 3 樣本最為稀少。

經模型訓練完成後，於該測試集進行分類預測，整體準確率達 91%，顯示模型具備優異的整體預測能力。類別 0 (無病變) 之分類效果顯著，召回率高達 99%，F1 分數達 0.99，幾乎完全準確識別健康樣本。

對於類別 2 (中度病變)，模型呈現良好識別能力，準確率為 81%，召回率亦高達 99%，F1 分數為 0.89，惟精確率略低於召回率，表示模型存在部分其他類別樣本被誤判為中度病變的情況。

相較之下，類別 1 (輕度病變) 及類別 3 (重度病變) 之分類表現較為不足。類別 1 的 F1 分數為 0.77，召回率僅 69%，顯示仍有一定比例的輕度病變樣本被誤判為其他等級；類別 3 召回率僅 51%，反映模型在重度病變判別上仍存在挑戰，F1 分數為 0.62。此一現象可能與臨床病徵邊界模糊及該類別樣本量較少有關。

此外，類別 4 (增殖型病變) 之識別表現令人肯定，召回率達 71%，F1 分數為 0.80，表明模型對於臨床高度警戒之嚴重病變具備一定的識別能力。如表 4.4 所示。

表 4.4 ResNet50 指標分析

類別	Precision	Recall	F1-score	support
0-No DR	1.00	0.99	0.99	361
1-Mild	0.88	0.69	0.77	74
2-Moderate	0.81	0.99	0.89	200
3-Severe	0.77	0.51	0.62	39
4-Proliferative DR	0.91	0.71	0.80	59

4.2.4 統整與比較

因類別的樣本數差異比較大，使用加權平均指標 (weighted avg) 做為比較依據。參考三種模型的加權平均數，將各項數值乘上樣本數再除以總樣本數後，數值以百分比表示，如表 4.5。

表 4.5 加權平均指標數據

	CNN	VGG16	ResNet50
Accuracy	98.87%	90.82%	83.93%
Precision	93%	69%	92%
Recall	94%	69%	91%
F1-score	93%	68%	91%

模型加權指標之說明如以下三點：

(1) CNN：

在數據上顯示 CNN 的 weighted avg 與 Accuracy 相比略低，表示仍有少量類別被錯分，但整體表現依然非常優異。

(2) VGG16

VGG16 的 Precision 和 Recall 約為 69%、F1 約為 68%，兩項數值遠低於 Accuracy，代表在樣本量大的主要類別上準確度高，但在少數類別的預測效果不佳，導致加權平均指標下滑。

(3) ResNet50

ResNet50 的 weighted avg 相較於 83.93% 的 Accuracy 反而更高，表示在少數類別的表現優於 CNN，使得整體預測分布更均衡。

最後，統整對訓練的三種模型之比較，CNN 整體準確度較高，最後使用 CNN 作為系統訓練模型。

第 5 章 人力配置

參與本研究之工作人員及預定之工作內容如表 5.1 所示。

表 5.1 參與本項研究之人員及工作內容

姓名	工作內容
俞華鈴	資料蒐集、模型訓練與優化、資料整理與報告統整
李馥彤	資料蒐集、模型訓練與實驗分析、美術設計與報告編排
林永泰	資料蒐集、系統架構設計、網頁前端開發、美術設計

第 6 章 預期成果

我們以簡約直觀的網頁介面作為主要視覺呈現，透過清晰的圖示與按鈕，讓醫師能輕鬆操作並集中管理所有患者的檢查與診斷紀錄。整體功能包含患者資料管理、診斷結果儲存、風險分析以及回診提醒通知，協助醫師即時掌握需回診的高風險族群，進一步提升臨床效率與醫療品質。

在操作流程上分為三點：

(1) 上傳影像：

點擊「上傳」按鈕，選擇要分析的視網膜圖片。

(2) 清除影像：

若想重選或取消，點擊「清除」按鈕，即可移除當前圖片並重新上傳。

(3) 執行分析：

上傳完成後，點擊「分析」按鈕，系統將在畫面上以視覺化方式顯示處理後的影像結果，並同時回傳 DR 分級、分析結果摘要與對應的治療建議。

整合以上功能後，網頁不僅能讓醫師快速完成影像診斷作業，也能將結果與患者資料同步儲存，並透過風險分析與排程通知主動提醒回診，達到輔診診斷之目標。

第 7 章 執行進度

糖尿病視網膜病變辨識系統專題執行進度規劃表，如圖 7.1。

進度 \ 月份	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
題目選定											
資料收集分析											
網頁											
模型測試											
研究方法											
論文撰寫											

圖 7.1 執行進度甘特圖

參考文獻

- [1] International Diabetes Federation. (2025). IDF Diabetes Atlas (11th ed., pp. 41–42). Brussels, Belgium: International Diabetes Federation. Retrieved from <https://diabetesatlas.org/resources/idf-diabetes-atlas-2025/>
- [2] C. Steele, D. Steel, and C. Waine, Editors (2008), “5 - Acute And Chronic Complications Of Diabetes Mellitus And Living With The Disease,” *Diabetes and the Eye*, pp. 51-58, DOI: 10.1016/B978-0-08-045307-1.50010-1.
- [3] D. S. W. Ting, G. C. M. Cheung, and T. Y. Wong (2016), “Diabetic Retinopathy: Global Prevalence, Major Risk Factors, Screening Practices And Public Health Challenges: A Review,” *Clinical & Experimental Ophthalmology*, Vol. 44, No. 4, pp. 260-277, DOI: 10.1111/ceo.12696.
- [4] K. Viswanath, and D. D. M. McGavin (2003). “Diabetic Retinopathy: Clinical Findings And Management,” *Community eye health*. Vol. 16, No. 46, pp. 21-24.
- [5] Indian Diabetic Retinopathy Image Dataset (IDRiD) <https://ieee-dataport.org/open-access/indian-diabetic-retinopathy-image-dataset-idrid>
- [6] APTOS 2019 Blindness Detection – Kaggle: <https://www.kaggle.com/c/aptos2019-blindness-detection>
- [7] Yeh, Y.-H. (2018, July 13). Machine Learning — Transfer Learning. Medium. Retrieved May 19, 2025, from <https://medium.com/@yuhsienyeh/machine-learning-transfer-learning-遷移學習-5095f8a14367>

附錄 A 開發工具介紹

A.1 軟體開發工具介紹

本專題使用 python 架構撰寫，並在 colab 平台上使用 TensorFlow 類神經網路架構來訓練模型，並在 Visual Studio Code 上設計主要程式架構。

A.1.1 Visual Studio Code

Visual Studio Code (VS Code) 是微軟於 2015 年推出的免費、開源且跨平台編輯器，基於 Electron 架構，兼具輕量啟動與高效能。內建語法高亮、智慧提示、程式片段、整合終端與強大除錯功能，並透過 Language Server Protocol 支援多種語言的自動補全與語意分析。使用者可從擴充市場安裝數千款外掛，如 Python、JavaScript、C#、Docker 與 Kubernetes 支援，以及 ESLint、Prettier、GitLens 等工具，實現代碼格式化與版本控制。VS Code 原生整合 Git，提供側邊欄差異檢視、提交與分支管理。VS Code 已成為全端工程師、資料科學家與 DevOps 團隊的首選開發利器。如圖 A.1。

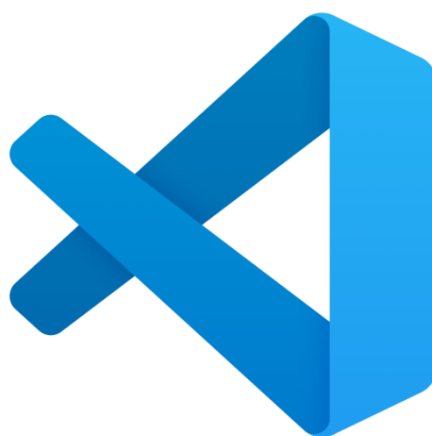


圖 A.1 Visual Studio Code

A.1.2 TensorFlow

TensorFlow 是 Google Brain 團隊於 2015 年開源推出的機器學習平台，採用計算圖機制與優化執行時引擎，可高效執行張量運算。它整合 Keras 高階 API，提供模型定義、訓練與評估一體化接口，同時支援 Eager Execution 增強開發靈活性。借助 `tf.data` 可構建高效數據管道，結合混合精度和 `tf.distribute` 分散式策略，實現多 GPU 與 TPU 集群擴展訓練。內建 TensorBoard 用於可視化指標、網絡結構及超參數調優；訓練完成後，可導出 SavedModel，並透過 TensorFlow Lite、TensorFlow.js 與 TensorFlow Serving 在行動端、瀏覽器與服務端實現一站式部署。豐富生態與跨平臺支持，使其成為從原型開發到大規模生產部署的理想選擇，如圖 A.2。



圖 A.2 TensorFlow

A.1.3 Colab

Colab，全名 Google Colaboratory，是由 Google 提供的一款免費且免安裝配置的雲端 Python 開發環境，專為數據科學和機器學習任務而設計。它基於 Jupyter Notebook，允許用戶在瀏覽器中直接撰寫並執行 Python 代碼，並預裝了 NumPy、Pandas、Matplotlib、Scikit-learn、TensorFlow、PyTorch 等主流科學計算與深度學習庫。使用者只需擁有 Google 帳戶，即可一鍵啟動運算環境，並在「執行階段設定」中選擇 NVIDIA GPU 或 Google TPU 加速器，以滿足各種高性能訓練需求。Colab 無縫整合了 Google Drive，讓用戶能方便地讀寫和備份資料集，並支援與 GitHub 的協同操作，以便於版本控制與團隊協作。此外，多人可同時在線編輯同一份 Notebook，並透過分享連結或導出為 .ipynb、HTML、PDF 等格式，輕鬆分發研究成果或教學內容。由於免除了本地環境配置的複雜性並提供了強大的雲端計算資源，Colab 成為初學者快速入門以及專業人士進行原型驗證和模型部署的首選平臺，如圖 A.3。



圖 A.3 Colab

A.2 影像辨識電腦硬體

影像辨識電腦硬體說明如表 A.1。

表 A.1 影像辨識電腦

作業系統	Windows 11 家用版
處理器	Intel(R) Core(TM) i7-12700H
晶片組型號	Intel HM670 PCH
標準記憶體	16GB
記憶體	DDR5-4800
外部顯卡	NVIDIA GeForce RTX 4050 laptop GPU