

中華民國第 65 屆中小學科學展覽會

作品說明書

高級中等學校組 電腦與資訊學科
團隊合作獎

052503

AI' S A IS FOR APPLE 基於人工智慧影像辨識之水果品質辨識及分類

學校名稱： 國立新竹女子高級中學

作者：	指導老師：
高二 王婕霏	徐逸娟
高二 陳品妍	
高二 趙子涵	

關鍵詞： 影像識別、電腦視覺、人工智慧

摘要

本研究開發了一種基於影像辨識的水果品質分類系統，採用最適合同時處理局部及全局資訊的 Swin-Transformer 模型，透過分析水果外觀來判斷其品質，並多次實驗以調整參數、訓練多個模型以辨識不同水果種類與品質。

使用者上傳水果影像後，系統即能識別水果種類和品質，並提供新鮮度建議與食用方式。此系統相較於傳統的檢測方法，具備非破壞式檢測優勢，且降低成本，適用於農業生產與消費市場。

模型測試結果顯示，水果類型分類器準確率為99.0%，蘋果品質分類器為85.04%，橘子品質分類器為97.67%，顯示該系統在分類與品質評估上具有較高的準確性，對水果檢測及提升食品管理有重要意義。

壹、研究動機

隨著人工智能和機器學習技術的迅速發展，影像辨識技術在各個領域的應用越來越廣泛，特別是在農業和食品產業中。由於肉眼無法準確辨識水果品質，導致品質不佳，甚至已經過期或變質的水果，這不僅影響了食用體驗，還可能導致浪費。

基於此，本研究旨在開發一種基於影像辨識技術的水果品質分類系統，透過自動化的方式，幫助消費者準確分類水果的品質。研究將利用攝影機對水果進行影像捕捉，並結合機器學習演算法，對水果的外觀特徵進行分析，從而判斷其是否成熟、是否存在腐爛、病蟲害或其他品質問題。提供水果的新鮮度、品質等評估訊息，幫助消費者在購買時及食用前做出更準確的選擇，避免選擇不合格的水果，從而提升購物效率和食品安全，減少食品浪費。

現有的研究方法多需使用特殊儀器或破壞水果，因此本研究希望透過對影像辨識技術的深度應用，建立一個自動化、不破壞水果的水果標準化辨識系統，進而解決大眾在水果標準化過程中常遇到的難題，並推動食品產業的科技創新和數位轉型。

貳、研究目的

本研究的研究目的為以深度學習的技術，進行水果品質的分類，以達到下列目標：

- 一、使用影像辨識，進行非破壞式影像偵測。
- 二、將偵測到的不同水果依據種類進行分類。
- 三、將偵測到的水果依據品質優劣進行分類。
- 四、使系統根據水果的分類結果，提供適當的食用方式。

參、研究設備及器材

CPU:

型號：Intel Core i9-7900X

規格：10核心、20執行緒，時脈 4.50GHz

GPU:

型號：NVIDIA GeForce RTX 2080

規格：2944 CUDA核心，8GB GDDR6 記憶體

記憶體

型號：Kingston 9905701-020.A00G

規格：4條 16GB Kingston DDR4記憶體，頻率2667MHz，總容量64GB

硬碟：

SUS: WDC WD8004FRYZ-01VAEB0

規格：7.3TB SATA HDD

程式語言：

Python 3.12.3

函示庫套件：

PyTorch, TorchVision, CUDA Toolkit, Timm, OpenCV, Termcolor, YACS, PyYAML, Scipy

肆、文獻回顧

一、水果品質辨識相關產品與論文比較

研究者發覺現有的水果品質辨識研究中多使用紅外線光譜影像、高光譜成像技術等，設備昂貴，計算成本大。

表4-1 水果品質相關文獻探討（研究者自製）

產品或論文	待改良處
智慧型非破壞式水果品質分類系統（ 教育部，2021 ）	設備繁雜，裝置難以攜帶
食品原物料影像辨識技術（ 財團法人精密機械研究發展中心，2022 ）	需使用特殊設備如RGB-D相機
農業部國產優良品牌水果認證制度（ 陳世芳，2021 ）	需破壞水果並食用判定
Felix Produce Quality Meter 非破壞性果實品質測定儀（ Felix Instruments, 2014 ）	設備成本昂貴如近紅外光譜儀
Non-invasive hyperspectral imaging approach for fruit quality control application and classification: case study of apple, chikoo, guava fruits (Vetrekar et al., 2015)	需使用特殊影像如高光譜成像

為使此模型可被日常使用以幫助採購或食用與處理判斷，需要將設備限制在容易取得的範圍內，因此排除使用昂貴或計算成本大的特殊設備。

二、影像辨識模型比較

最終本研究決定透過「影像辨識」以偵測水果品質，以下為技術方法描述：

(一) 針對同時辨識水果種類與品質之模型表現比較

表4-2 影像辨識模型比較（研究者自製）

模型	CNN (維基百科貢獻者, 2025)	ViT (Shiau & Su, 2024)	Swin-Transformer (Liu et al., 2021)
描述	透過卷積核 (Kernels)滑動對圖像做訊息提取，並藉由步長 (Strides)與填充(Padding)控制圖像的長寬。	將影像劃分為小塊(patches)，並將這些小塊作為輸入，利用變換器的自注意力機制 (MHSA)來提取全局特徵。	將影像分為小塊後，利用滑動窗口機制(Shifted Window Attention) 將影像劃分為局部窗口(W-MSA)，減少計算複雜度；引入SW-MSA則可交互窗口間的特徵，讓訊息在窗口間傳遞。
特徵學習方式	局部特徵學習	全局特徵學習	局部特徵學習 + 全局特徵學習，透過層次化架構逐步擷取影像資訊，兼具局部與全局特徵。
處理變形與視角變化	受限於局部特徵學習，對變形適應差	能較好處理長程依賴，對視角變化適應性較強	適應性強，透過滑動窗口機制提升對不同視角與變形的適應能力。
水果種類辨識 (全局)	中等	強	強
品質瑕疪檢測 (局部)	強	中等	強

(二) 參考文獻

1. CNN : A review of external quality inspection for fruit grading using CNN models ([Chuquimarcá et al., 2024](#))
 - (1)研究目的：綜述卷積神經網絡 (CNN) 模型在水果外觀品質檢測與分級的應用現狀。
 - (2)資料集與影像類型：分析使用的資料集數量、影像類型包含可見光、紅外線、超光譜、多光譜之照片，以及資料來源為實際或合成影像。
 - (3)模型比較：評估各種 CNN 模型的分類效能，探討使用預訓練模型進行遷移學習的效果，以及不同優化器對模型準確率的影響。

(4)結果：透過表格整理各研究的檢測參數與模型效能，提供研究者在選擇適當模型與方法時的參考。

2. Vision Transformer (ViT) : Performance of Vision Transformer and Swin-Transformer models for lemon quality classification in fruit juice factories ([Dümen et al., 2024](#))

(1)研究目的：評估 Vision Transformer 與 Swin-Transformer 在檸檬品質分類上的效能，並與其他深度學習模型進行比較。

(2)模型比較：實作八種深度學習模型，如 Xception、ResNet-50、InceptionV3等，與ViT 及 Swin-Transformer 兩種transformer 模型進行分類任務。

(3)結果：ViT 模型在分類準確率、召回率與精確率上表現最佳，Swin-Transformer 次之，優於其他深度學習模型。

(三) 總結

在水果品質辨識應用中，第一個文獻針對以 CNN 模型進行水果品質檢測與分級回顧。該文獻指出，多數研究集中於單一果種或特定品質指標，並主要仰賴 CNN 模型進行特徵提取與分類判斷。雖然 CNN 在早期取得良好成果，但在處理高複雜度多樣性影像資料上逐漸顯現其侷限性。

另一方面，Transformer 架構開始展現出相對於傳統 CNN 模型的潛力與優勢。在第二篇研究中以 ViT 為主體，探討其在檸檬品質分類任務中的表現，結果顯示 ViT 明顯優於包括 ResNet、InceptionV3 等傳統卷積模型。該研究表明 ViT 在處理農產品外觀辨識時，能夠有效捕捉整體影像資訊與高階特徵，突破了卷積模型在局部感知上的限制。然而，ViT 在面對高解析度或多尺度影像時，其固定分割機制與全局自注意力計算成本可能造成效率與精度的瓶頸。本研究因而選擇 Swin-Transformer 作為替代，藉其層級化與滑動窗口設計，以解決 ViT 架構在實務應用中的延展性限制。

對照之下，本研究採用 Swin-Transformer 模型，其多尺度窗口注意力機制能更有效捕捉水果外觀的細部變化，亦更適應跨品種間的分類任務。以下針對Swin-Transformer 做更詳細的解說。

三、Swin-Transformer 模型

目前已有許多使用CNN或ViT處理水果品質辨識的模型，但並未有同時辨識水果種類及品質的任務。Swin-Transformer作為較晚出現，較少被用於處理類似任務的研究，且可兼顧全局及局部特徵處理，更加符合我們的辨識目標需求。故選擇Swin-Transformer模型進行研究。

Swin-Transformer 通過將圖像切分為小的區域來進行處理，每個區域內部會進行自注意力計算，圖4-1。這樣的局部窗口化自注意力能夠更高效地捕捉到圖片中的細節特徵，如水果的顏色、形狀、表面質地等。此外，Swin-Transformer 使用移動窗口 (shifted windows)來擴展模型的感受野，使得它比傳統的卷積神經網絡更具靈活性，有助於捕捉更大範圍的上下文信息，如水果的整體外觀和可能的品質變化。最後，如圖4-2，Swin-Transformer 通過層次結構處理圖像及不同尺度的圖像特徵，從低層的細節特徵到高層的全局特徵，都能夠精確地反映水果的品質信息。

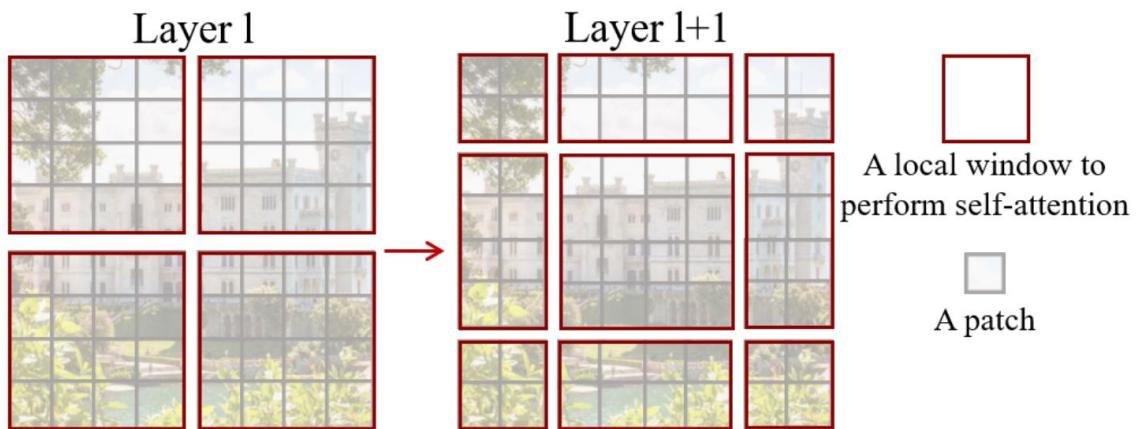


圖4-1 Swin-transformer局部窗口自注意力機制與圖像分塊處理 (Liu et al., 2021)

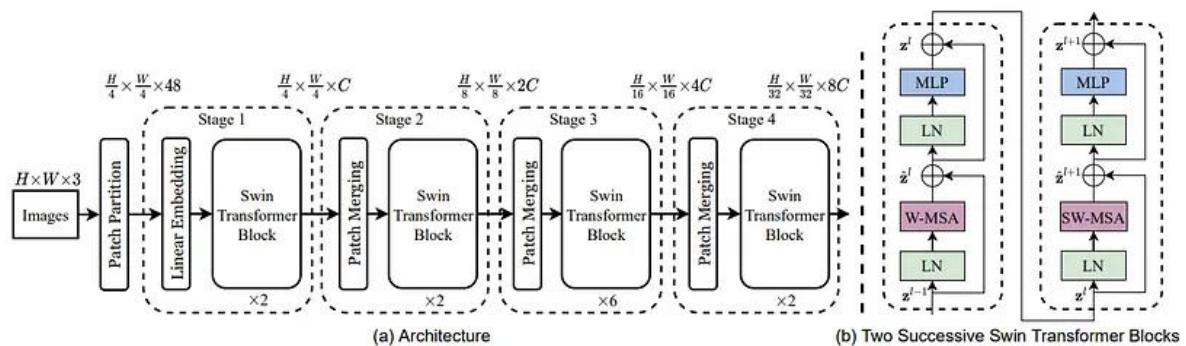


圖4-2 左側為整體架構，右側為 Swin-Transformer Blocks 詳細的架構 (Liu et al., 2021)

伍、研究過程與方法

一、流程規劃圖

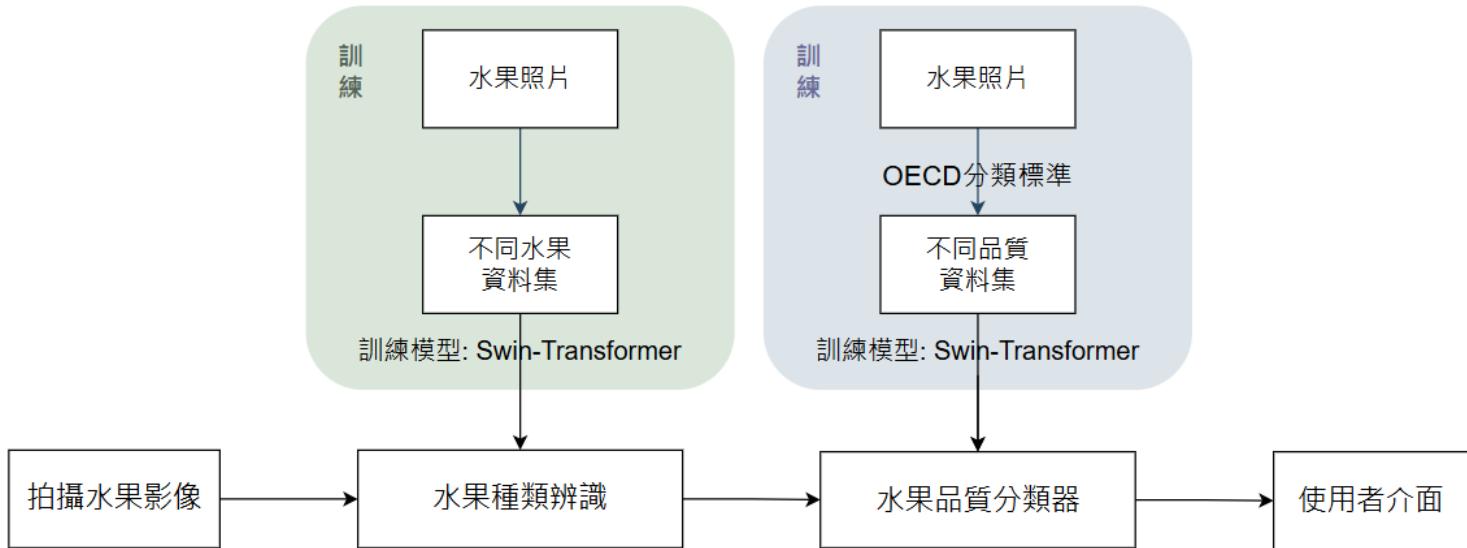


圖5-1 研究流程示意圖（研究者自製）

二、訓練資料的收集與處理

作者找到多種水果的照片後，先分出不同水果種類，接著再根據OECD的分類方法，將每種水果分為A~D由優到劣四類，最終提供多種資料集透過模型做訓練。

(一) 資料來源

大部分為網路上的開源資料集，例如kaggle網站中的dataset專區所提供的訓練資料集，以及部分作者自行拍攝。本實驗共下載了12組資料集 (Abdul Nabi, 2022; Bali, 2022; Bhatt, 2024; Hilton, 2023; Hossenabir, 2024; Huang & Chen, 2023; Maher, 2023; Marquis03, 2023; Patil & Meshram, 2022; Park, 2022; Roboflow, 2024; Subhan, 2024)，其中涵蓋了不同種類和狀態的水果。

(二) 品質分類標準

1.經濟合作暨發展組織 (Organization for Economic Cooperation and Development)

經濟合作暨發展組織，英文縮寫OECD，旨在幫助各成員國家的政府實現可持續性經濟增長和就業，保持金融穩定，從而為世界經濟發展作出貢獻。本實驗參考該組織出版的蔬果品質分類標準 (International Standards for Fruit and Vegetables) (OECD, 2024)

這些出版物針對不同水果提供插圖和評論，有助於針對水果不同特徵進行客觀分類。隨著人工智能和機器學習技術的迅速發展，影像辨識技術在各個領域的應用越來越廣泛，特別是在農業和食品產業中。由於肉眼無法準確辨識水果的內部品質，導致品質不佳，甚至已經過期或變質的水果，這不僅影響了食用體驗，還可能導致浪費。

2. 分類標準

OECD由優質到劣質將水果分成Extra Class, Class I, Class II, Not allowed 四類，在本作品中稱其為A,B,C,D：

Extra Class (A類)：外觀完整，且顏色均勻，可食用

Class I (B類)：輕微受損，或顏色較不均勻，可食用

Class II (C類)：受損較為嚴重，處理後可食用

Not allowed (D類)：嚴重受損，不可食用

以蘋果為例，OECD提供相關敘述及圖片，並將其分類(圖5-2)



41: Very slight superficial defect – Limit allowed in Extra Class

A類



42: Slight skin defects due to hail, total surface affected not exceeding 1 cm² – Limit allowed in Class I

B類



48: Skin defects due to scab (*Venturia inaequalis*) not exceeding 1 cm² – Limit allowed in Class II

C類



17: Superficial scald – Not allowed

D類

圖5-2 OECD蘋果分類判別敘述 (OECD, 2024)

參考OECD的分類標準，我們將同種水果圖片分成四類並將資料集命名為「水果_類」，例如：apple_A, apple_B, orange_C 等。接著取一定的數量匯入模型的訓練及驗證資料夾，並保留一部份作為測試集。

三、Swin-Transformer 模型訓練

(一) 超參數調整

本研究為提高模型表現，透過調整並比較不同batch size對於準確率及損失函數loss的影響，選擇最適合模型的Batch size 參數。總共測試了 Batch Size 32、64、128 三種不同的值。

1. 損失函數 (Loss)

在參數調整的時候研究者使用損失函數衡量模型表現。相較於準確率，損失函數能提供更細緻的誤差資訊，使模型能在訓練過程中有效學習並改善預測能力。

準確率僅判斷預測結果是否正確，而損失函數則能衡量預測值與真值之間的距離，使模型能逐步修正誤差。特別是在深度學習中，模型透過梯度下降 (Gradient Descent) 來更新模型參數以最小化損失，使訓練過程能有效收斂。

(二) 模型訓練

我們使用Swin-Transformer的Swin-B版本進行微調訓練，以下是模型架構。

1. Patch Embedding

在 Swin Transformer 中，圖片的處理過程從 Patch Embedding 開始。這一部分的主要目的是將輸入圖像分割成小區塊 (patches)，並將這些區塊映射到高維的特徵空間，成為可以進行自注意力運算的 tokens。

具體步驟如下：

(1) 圖片分割 (Image Partitioning)

首先，將輸入圖像 X 劃分為大小為 $P \times P$ 的小區塊 (Patch)。假設原始圖像的大小是 $H \times W$ ，則劃分後的 Patch 數量為：

$$N = \frac{H}{P} \times \frac{W}{P}$$

每個 Patch 是一個 $P \times P \times C$ 的子圖，其中 C 是圖像的通道數。在本次實驗中，因使用RGB通道，故 C 為3。

(2) 展開每個 Patch (Flattening the Patch)

每個 $P \times P \times C$ 的 Patch 會被展平為一個一維向量。這樣每個 Patch 都會變成一個長向量，所有 Patch 的展平向量將組成一個矩陣。

$$P = [p_1 \ p_2 \ \dots \ p_N] \in R^{N \times P^2C}$$

(3) 線性投影 (Linear Projection)

接下來，每個展平的 Patch P_i 會經過一個線性投影層（也就是一個全連接層），將每個 Patch 映射到一個 D -維的嵌入空間。這樣，每個 Patch 會被轉換為一個維度為 D 的向量，並組成一個矩陣 Z_0 ，這些向量就是最終的 token：

$$Z_0 = P \cdot W_p + b_p$$

其中：

$P \in R^{N \times P^2 C}$ 是之前展平的Patch。

$W_p \in R^{P^2 C \times D}$ 是線性投影矩陣，將每個Patch映射到D-維空間。

$b_p \in R^D$ 是偏置項。

$Z_0 \in R^{N \times D}$ 是每個Patch投影後的特徵表示。

這些嵌入向量 Z_0 就是 Transformer 模型後續計算的輸入 token，它們包含了圖像的局部信息，並將這些局部特徵轉換為高維特徵。

2. Feature Extraction

進入 Swin-Transformer 之後，這些 token 經過多層處理，其中主要包含以下幾個關鍵步驟：

(1) Swin-Transformer Block

如圖5-3，Swin-Transformer Block 是 Swin-Transformer 的核心運算單元，它負責處理圖像的特徵並提取局部與全局資訊。每個 Swin Transformer Block 主要包含兩個自注意力機制（W-MSA 和 SW-MSA）。

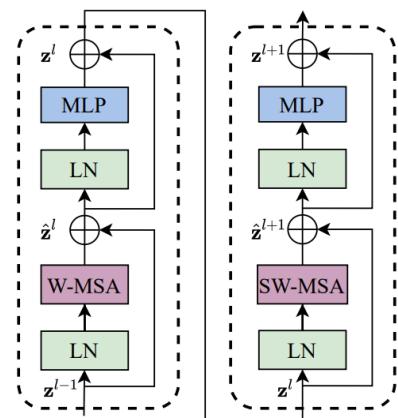


圖5-3 Swin Transformer Block
(Liu et al., 2021)

(2) 窗口自注意力(W-MSA)

W-MSA的主要目標是降低傳統自注意力（MSA）的計算量，同時保持局部區域內的資訊學習。

因此，Swin-Transformer 採用「局部窗口」來計算自注意力，每個窗口大小為 $M \times M$ 。對於窗口內的 token，計算標準的 Scaled Dot-Product Attention：

$$\text{Attention}(Q, K, V) = \text{Softmax} \left(\frac{QK^T}{\sqrt{d}} + B \right) V$$

其中：

- Q, K, V為Query, Key, Value 矩陣
- d是注意力維度
- B是相對位置偏差(Relative Position Bias)

假設一個圖片有 $h \times w$ 個patch，則使用MSA與W-MSA的複雜度分別為

$$\Omega(\text{MSA}) = 4hwC^2 + 2(hw)^2C,$$

$$\Omega(W - \text{MSA}) = 4hwC^2 + 2M^2hwC,$$

其中C為通道數。

(3) 滑動窗口自注意力(SW-MSA)

雖然窗口化自注意力大大減少了計算量，但它會限制窗口內的特徵學習，並且不同窗口之間的資訊無法交互。為了解決這個問題，Swin-Transformer 引入了 移動窗口的概念，其操作方式如圖5-4。

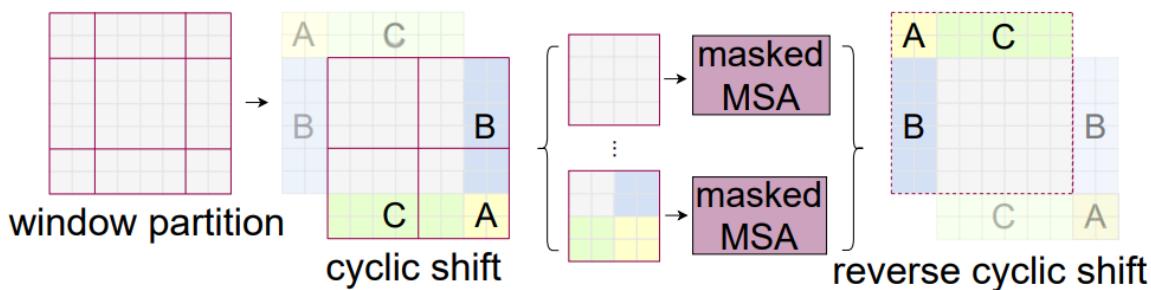


圖5-4 Swin-Transformer滑動窗口自注意力 (Liu et al., 2021)

(三) 模型種類

我們共有兩種模型，第一種是用來分辨不同的水果種類，第二種則是用來辨識水果的品質好壞。

1. 模型一：水果種類辨識

(1) 功能與目標

此階段專注於提取水果圖像的特徵，例如顏色、形狀、紋理等，並根據這些特徵進行分類。Swin-Transformer 的局部窗口化自注意力機制能有效捕捉水果的細節特徵，而移動窗口策略則有助於理解水果的整體外觀，從而實現種類辨識。此分類器可應用於多種水果，並輸出具體的水果種類名稱。

(2) 類別

此階段的資料集命名為水果名稱，資料集內包含該種水果的A到D類，目前包含兩種水果類別，分別是apple及orange。

(3) 模型輸出

水果名稱。此輸出內容將經過篩選器，來決定第二階段使用何種品質分類模型。

2. 模型二：水果品質辨識

(1) 功能與目標

此階段專注於評估水果的品質等級。Swin-Transformer 模型將分析水果圖像的外觀特徵，如表面瑕疵、顏色均勻度、成熟度等，並根據這些特徵判斷水果的品質。模型通過層次結構處理圖像，從局部細節到全局特徵，反映水果的品質信息。最終輸出水果的品質等級。

不同水果會對應到不同的品質辨識模型，我們將訓練多個針對不同水果的品質模型，當圖片經過第一個模型後，便可根據模型一的判斷結果，選擇對應到該水果的模型二。

(2) 類別

將每種水果根據OECD分類，以蘋果資料集為例，四個類別分別是apple_A, apple_B, apple_C, apple_D，其他水果以此類推。

(3) 輸出：水果品質。

表5-1 兩種訓練模型比較（研究者自製）

模型	模型一：水果種類辨識	模型二：水果品質辨識
功能	提取多種水果圖像的特徵，並辨識不同水果種類	分析水果圖像的外觀特徵，並評估水果的品質等級
類別	apple, orange	apple_A, apple_B, apple_C, apple_D, orange_A, orange_B, orange_C, orange_D
輸入	不同水果的照片	同水果不同品質的照片
輸出	具體的水果種類名稱	水果的品質等級

四、模型測試

為了提升模型準確率，我們想知道是哪些照片被電腦判斷錯誤，藉此來回顧調整我們提供的訓練資料集有那裡可以修正，並不斷重複。

(一) 測試程式碼與原理說明

eval.py程式碼是用來對已訓練好的模型進行測試，對圖片進行分類後計算出準確率，以下為程式解析。

1.匯入所需函式庫與模組

2. parse_option()

如圖5-5，此函式功能為設置命令行參數解析，在測試時傳入各種選項（例如模型配置、資料集路徑等）。

```
def parse_option():
    parser = argparse.ArgumentParser('Swin Transformer Test script', add_help=False)
    parser.add_argument('--cfg', default='configs/swin_tiny_patch4_window7_224.yaml', type=str, metavar="FILE",
                        help='path to config file')
    parser.add_argument(
        '--opts',
        help="Modify config options by adding 'KEY VALUE' pairs. ",
        default=None,
        nargs='+',
    )
```

圖5-5 測試程式碼節-parse_option()（研究者自製）

參數解釋：

--cfg：指定配置文件的路徑，包含模型架構和超參數。

--data-path：資料集的路徑。

--resume：指定要加載的模型檔案的路徑。從指定的訓練中斷點（checkpoint）恢復模型的權重。

最後返回解析的參數:args和:配置選項config(模型的架構、訓練參數、數據集路徑等)。

3. 圖像處理

如圖5-6，在引入函式庫並準備所需函式後，我們將先使用操作檔案系統(os)，列出資料夾中的檔案後，以PIL(Python Imaging Library)讀取和處理圖像。接著使用圖像處理模組(torchvision.transforms)進行圖像的預處理: (transforms.Resize)將圖像大小調整為 224x224 像素，(Swin Transformer 模型所需的輸入尺寸)、(transforms.ToTensor)轉換圖片格式成 Tensor格式，並進行標準化處理(transforms.Normalize)，將像素值縮放到[-1, 1]範圍。

```
// transform_test = transforms.Compose([
//     transforms.Resize((224, 224)),
//     transforms.ToTensor(),
//     transforms.Normalize([0.5, 0.5, 0.5], [0.5, 0.5, 0.5])
//])
```

圖5-6 測試程式碼節錄-圖像處理（研究者自製）

4. 建立模型並載入權重

如圖5-7，解析配置選項(config)然後建立模型，並載入已經訓練好的模型權重。設置模型為 eval() 模式，這樣它就不會進行梯度計算，可以進行測試。

```
DEVICE = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
args, config = parse_option()
model = build_model(config)
checkpoint = torch.load('/media/ibm/F/pinyan/Swin-Transformer-main/outp
model.load_state_dict(checkpoint['model'], strict=False)
model.eval()
model.to(DEVICE)
```

圖5-7 測試程式碼節錄-載入權重（研究者自製）

因為在訓練過程中，模型會根據輸入數據和目標標籤進行前向傳播，然後計算損失 (loss)。接著，模型會進行反向傳播 (backpropagation)，計算梯度，並根據這些梯度更新模型的參數，以最小化損失。

然而在測試過程中，我們只需要模型對新數據的預測結果。不需進行反向傳播或更新模型的參數。所以測試時不需要計算梯度。

且計算梯度的過程需要額外的內存和計算資源。因此我們選擇去除梯度計算以節省內存，並加快測試速度。

5. 測試圖像並計算準確率

如圖5-8，先讀取指定資料夾中的測試圖像 (testList)，接著對每張圖片進行預處理 (transform_test) 並將其送入模型進行測試。

torch.max() 函數的作用是返回每張圖像輸出的最大值及其對應的索引（這裡的索引即為預測類別）。

其中`_`, `pred = torch.max(out.data, 1)` 會取出模型的預測結果，而`out.data`為模型的輸出。將輸出對應至索引後，逐張顯示圖片的名稱及預測類別，同時統計各類別的預測數量並計算準確度。

```
for file in testList:  
    img = Image.open(path + file)  
    img = transform_test(img)  
    img.unsqueeze_(0)  
    img = Variable(img).to(DEVICE)  
    out = model(img)  
    # Predict  
    _, pred = torch.max(out.data, 1)  
    print('Image Name: {}, predict: {}'.format(file, classes[pred.data.item()]))  
  
    total += 1  
    if classes[pred.data.item()] == args.classes:  
        acc += 1  
  
    result[pred.data.item()] += 1
```

圖5-8 測試程式碼節錄-測試圖像並計算準確率（研究者自製）

6. 輸出結果

最後，如圖5-9，程式會顯示測試集的準確度，以及每個類別的預測數量，輸出結果如圖5-10。

```
print()  
print('Accuracy:', acc / total * 100)  
print('照片總數:', total)  
print(f"預測為{', '.join(classes)}的照片數量: {result}")
```

圖5-9 測試程式碼節錄-輸出（研究者自製）

TERMINAL

```
Image Name: orange_A(326).jpg, predict: orange_A  
Image Name: orange_A(189).jpg, predict: orange_A  
Image Name: orange_A(494).jpg, predict: orange_A  
Image Name: orange_A(477).jpg, predict: orange_A  
Image Name: orange_A(438).jpg, predict: orange_A  
Image Name: orange_A(149).jpg, predict: orange_A  
Image Name: orange_A(103).jpg, predict: orange_A  
Image Name: orange_A(509).jpg, predict: orange_A  
Image Name: orange_A(309).jpg, predict: orange_A  
  
Accuracy: 97.36842105263158  
照片總數: 532  
預測為orange_A orange_B orange_C的照片數量: [518, 14, 0]
```

圖5-10 輸出結果（研究者自製）

五、模型使用

訓練完模型後利用程式 final_test.py 導入模型權重並進行水果辨識。final_test.py 分成兩階段，分別執行種類與品質的辨識，並使用變數將第二次模型辨識程式中需改變的參數及路徑依據首次模型測試結果選擇並改變，以下為實用流程：

(一) 輸入圖片與指令

需先將圖片上傳至指定位置並輸入指令

(二) 第一階段分類：水果種類辨識 (class)

第一階段為判斷水果類別，程式會測試指定路徑中的照片並輸出其對應的水果種類。

```
# **第一階段分類** (辨識水果種類)
classes = ('apple', 'orange')
test_path = 'dataset/test/single-test/' # 測試資料夾
test_list = os.listdir(test_path)

for file in test_list:
    img = Image.open(os.path.join(test_path, file))
    img = transform_test(img).unsqueeze(0).to(DEVICE)
    out = base_model(img)

    # 預測結果
    _, pred = torch.max(out.data, 1)
    fruit_class = classes[pred.item()]
    print(f'Image Name: {file}, predict: {fruit_class}')
```

圖5-11 實用測試第一階段程式節錄（研究者自製）

(三) 切換第二階段模型

在辨認完水果種類後，需切換到第二種模型以辨認水果品質，透過 if/else 條件式，將第二階段的變數及模型設定成該種水果的模型權重路徑與類別，如圖5-12，例如當第一階段預測結果為apple時，程式將把模型路徑切換至第二階段的蘋果品質分類模型。

```
# **根據第一階段分類結果，載入對應的模型**
if fruit_class == 'apple':
    trained_model_path = "/media/ibm/F/pinyan/Swin-Transformer-main/output/swin_base_finetuned_apple"
    fine_classes = ('apple_A', 'apple_B', 'apple_C', 'apple_D')
else:
    trained_model_path = "/media/ibm/F/pinyan/Swin-Transformer-main/output/swin_base_finetuned_orange"
    fine_classes = ('orange_A', 'orange_B', 'orange_C')

# 載入對應模型
fine_model = load_model(config, trained_model_path)
```

圖5-12 切換第二階段模型程式節錄（研究者自製）

(四) 第二階段分類：水果品質辨識 (fruit)

第二階段為判斷水果品質類別，程式會辨識指定路徑中的照片並輸出其對應的水果品質，如圖5-13。

```
# **第二階段分類** (辨識品質類別)
out = fine_model(img)
_, fine_pred = torch.max(out.data, 1)
final_class = fine_classes[fine_pred.item()]
print(f'Final classification: {final_class}')
```

圖5-13 實用測試第二階段程式節錄（研究者自製）

(五) 輸出結果與反饋

根據上述兩個模型的預測結果，利用descriptions輸出建議之食用方法，如圖5-14、5-15。

```
# **結果對應文字**
descriptions = {
    "apple_A": "它長得超完美啦！適合食用以及送禮！",
    "apple_B": "這顆蘋果可能長得不是那麼好看，但也適合食用喔！",
    "apple_C": "喔不！這顆蘋果必須切除受傷部分，才可以食用喔！",
    "apple_D": "它已經爛掉了！建議不要吃！",
    "orange_A": "它長得超完美啦！適合食用以及送禮！",
    "orange_B": "這顆橘子可能長得不是那麼好看，但也可以食用喔！",
    "orange_C": "它已經爛掉了！建議不要吃！",
}
print(descriptions[final_class])
```

圖5-14 輸出文字反饋程式節錄（研究者自製）

```
Image Name: apple_A(3).jpg, predict: apple
Final classification: apple_A
它長得超完美啦！適合食用以及送禮！
```

圖5-15 輸出結果（研究者自製）

六、研究結果與討論

一、超參數調整結果

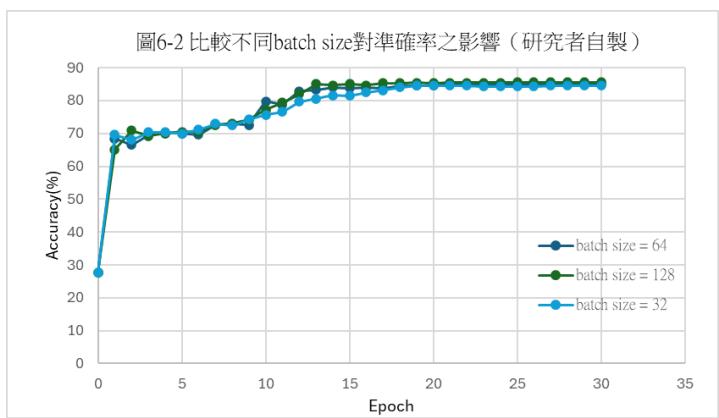
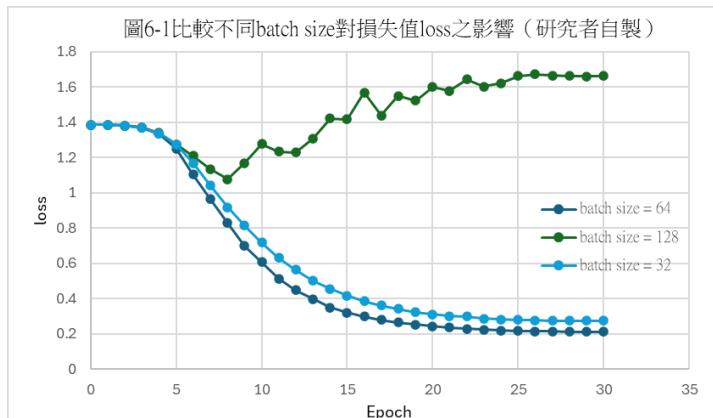
在本研究中，研究者使用相同的蘋果訓練資料集，並分別設定不同大小的Batch Size進行模型訓練，測試其對損失值（Loss）及準確率的影響。

如圖 6-1 所示，當 Batch Size 設定為 64 時，模型的 loss 值達到最小，這顯示在 Batch Size 為 64 的情況下，模型的訓練過程更加穩定，具備較好的收斂性，有助於提升其整體表現與泛化能力。

如圖6-2，當Batch Size 設定為 128 時，雖然模型的準確率達到了最高值 85.596%，但其 loss 值顯著高於 Batch Size 為 32 和 64 的情況。因此，僅以準確率作為衡量模型表現的指標可能存在偏差，無法全面反映模型的穩定性與泛化能力。這顯示在模型評估時，需綜合考慮 loss 值及其他指標，以獲得更可靠的性能評估。

表6-1 不同batch size之比較（研究者自製）

Batch Size	Batch Size = 32	Batch Size = 64	Batch Size = 128
最終損失值 loss(圖6-1)	0.2744	0.2133	1.6621
模型準確率 acc(圖6-2)	84.488%	85.042%	85.596%
說明	模型在每次更新時僅能利用較少的樣本，可能導致梯度更新穩定，影響模型收斂效果。	在此研究環境中提供了良好的平衡，既能夠有效學習關鍵特徵，又能維持梯度更新的穩定性，從而達到最優的損失值。	雖然可以提供較完整的樣本進行學習，但也可能因為更新頻率下降，使得模型的特徵學習不足或過度平滑，影響最終效能。



二、藉由錯誤影像提升蘋果品質分類模型準確率之分析與討論

嘗試提高蘋果品質分類模型準確率時，我們在測試時特意使程式回傳每張照片的分類結果，以便我們逐各查看分類錯誤的影像並分析原因。

透過分析錯誤辨識結果的原因，我們可以找出模型無法正確辨別的特徵，以下為幾個例子討論。

(一) 分類錯誤影像之原因分析與討論

1. 將蒂頭誤判成黑斑缺陷

如圖，這些照片原屬於A類(圖6-3)與B類(圖6-4)，但在測試時卻被模型分去D類，推測原因是由於中間蒂頭部分和黑斑相近，因而讓模型誤會，而被歸類於D類。

⇒ 解決方法：將圖6-3這種照片進行資料增強，將其隨機旋轉、平移、上下或左右顛倒，增加訓練照片數，使模型學習到此特徵。



圖6-3 A類被誤判為D類之蘋果照片
(Roboflow, 2024)

圖6-4 B類被誤判為D類之蘋果照片
(Roboflow, 2024)

2. 輕微疤痕判斷

我們觀察D類被分錯的照片，發現部分照片有一個共同點，這些被分錯的蘋果都是有部分黑斑或是爛掉，但紅色部分還是佔有一定面積，我們推測是因紅色部份過多而讓模型判斷錯誤。

例如表6-2中的圖6-5，因為此蘋果有小部分受損，理論上應該被分在切除後可食用的C類，但可能因為整顆蘋果大部分是完整的紅色，而被誤判為B類；而圖6-8則推測是蘋果除了右上角爛掉，其餘部分還是完整的紅色，所以被模型誤判成A類。

表6-2 小部分受損大部分完整造成模型誤判之例子（研究者自製）

圖				
圖名	圖6-5 (Subhan, 2024)	圖6-6 (OECD, 2024)	圖6-7 (OECD, 2024)	圖6-8 (Subhan, 2024)
實際類別	apple_C	apple_D	apple_D	apple_D
模型判別	apple_B	apple_C	apple_C	apple_A

3.不同品種蘋果

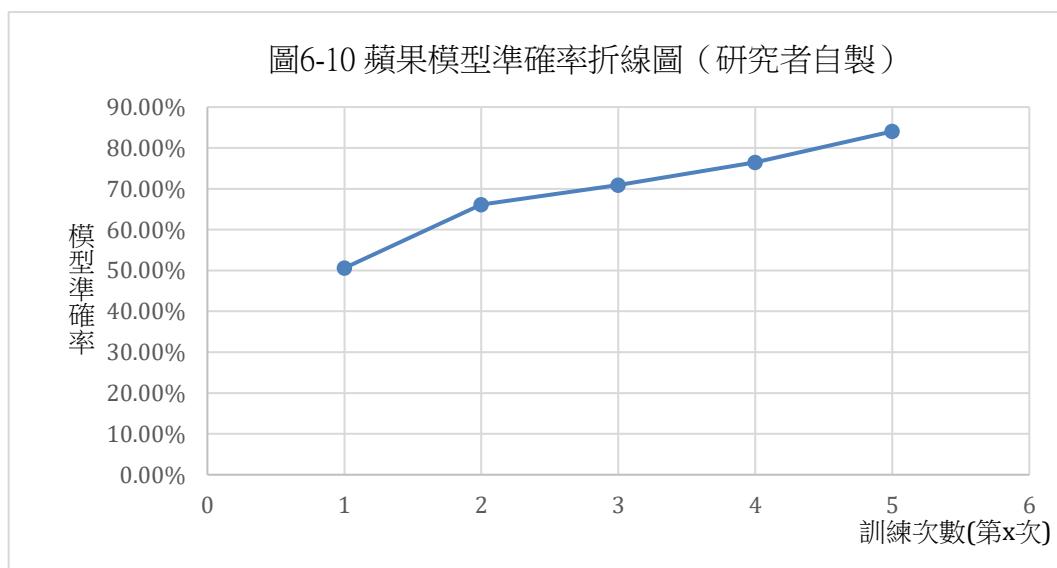
如圖6-9，這顆蘋果由於品種問題，是整顆黃色的，但無其他損傷，因此我們將其放在A類，但模型卻將其誤判為C類，為了避免出現這種狀況使判斷標準變模糊，我們決定在沒有將足夠多的黃蘋果照片納入訓練之前，先在資料集中刪除這種照片。



圖6-9 黃蘋果
(Roboflow, 2024)

(二) 錯誤影像分析之改良結果

將分類標準不斷調整後，我們又以更明確的分類標準重新將資料集二次檢查，防止模型學習到分類不清的特徵。如圖6-10，經過不斷重複測試訓練、找出問題並改善後，我們將準確率從初次的 50.6% 逐步提升至最終 85.04%。



三、橘子品質辨識模型準確率之分析與討論

(一) 橘子模型趨於二分法之原因討論

1. 初次測試模型表現

在剛開始訓練模型時，研究者只將現有的各類橘子照片分類後輸入模型做訓練，發現準確率可達82.9%，但當我們拿各類照片去測試時，如表6-3，發現模型可明確分出orange_A與orange_D的橘子，但在orange_B和orange_C的預測上則表現很差，模型會將這兩類的橘子分去A或D。

表6-3 橘子品質辨識模型表現（研究者自製）

	orange_A	orange_B	orange_C	orange_D
訓練照片數	200	50	60	200
驗證照片數	100	11	29	100
測試準確率	100.0	0.0	0.0	89.27

2. 推測原因

由上表結果，我們推測由於在訓練時給模型的照片數過於不平均，BC類照片過少，導致模型趨於二分法，只能分出A類與D類橘子，雖然看起來整個模型的準確率是相對較高的，但在單張測試時，就無法正確預測orange_B及orange_C。

3. 解決方法

為了改善上述問題，我們需要增加BC類的照片，使其訓練資料集與驗證資料集的數量與AD類平衡，使模型抓取BC類的特徵，以提高各類偵測的準確率。

(1) 增加orange_B、orange_C資料量

我們首先嘗試增加orange_B、orange_C資料量，除了在網路上下載資料集與自行拍攝外，我們使用了幾種常見的資料增強技術，例如對圖片進行隨機旋轉、平移、上下或左右顛倒圖片。資料增強的目的是為了通過對現有的訓練數據進行變換來創建更多樣化的訓練樣本，從而讓模型學到更多的特徵。

(2) 將orange_B,orange_C合併

不過由於網路上可使用的橘子影像資料集中品質極端，大多為新鮮、完美的與極端腐爛的類別，分別對應到orange_A與orange_D。於是orange_B、orange_C的資料量最終還是不足以平衡差距，所以我們在討論後將orange_B、orange_C合併，將其簡化為orange_A、orange_B、orange_C三類進行接下來的訓練及測試。

(二) 類別簡化改良結果

1. 橘子B類測試準確率

將橘子改為三類以後，新的orange_B類型中模型表現由0.00%提升至95.54%。

為進一步提升模型表現，接下來我們將對被誤判的影像進行歸納與原因討論。

表 6-4 橘子類別簡化後結果（研究者自製）

	orange_A	orange_B	orange_C
訓練照片數	600	600	550
驗證照片數	200	200	200
測試準確率	96.99	95.54	90.46

2. 錯誤影像分析

如圖6-11，模型將這顆原屬於C類的橘子分去A類，推測原因是其發霉範圍較小，且發霉部分與蒂頭相似，而使模型判斷錯誤。我們可透過資料增強加強模型對此類型影像的判別能力。

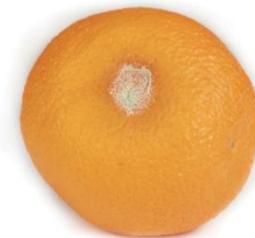


圖6-11 C類被誤判為A類之橘子照片
(Subhan, 2024)

(三) 最終改良結果

簡化分類結果及增加資料量後，我們使用重新篩選過的資料集進行訓練，以確保模型學習到不同的特徵。經過不斷重複測試訓練、找出問題並改善後，我們將橘子品質辨識模型的準確率提升至97.67%。

柒、結論

一、總結

(一) 水果訓練資料集

目前多數開源資料集僅將水果依種類進行標註，而針對品質的分類則較為少見。因此，本研究蒐集多種水果照片，先依據種類分類，再參考 OECD 的分類方法，根據不同水果的標準進一步細分品質等級。例如：橘子分為A、B、C三類，蘋果則分為A、B、C、D四類。最終，本研究提供多組標註完善的資料集，以利模型進行訓練與應用。

(二) 模型訓練

1. 經過實驗後，研究者找出batch size = 64時，模型學習效果最佳。
2. 模型訓練結果

表7 模型訓練結果（研究者自製）

模型	水果種類分類器	蘋果品質分類器	橘子品質分類器
準確率	99.0%	85.04%	97.67%

(三) 誤判情形討論

1. 蘋果誤判情形

本研究發現，當蘋果品質分類模型遇到蒂頭較深色的蘋果時，系統易將其誤判為黑斑缺陷，研究者透過分析錯誤分類影像，發現蒂頭與黑斑的顏色和形狀相似，導致模型混淆。因此，我們透過資料增強（如旋轉、平移、翻轉）來提升模型對蒂頭特徵的辨識能力，以修正此問題。

此外，當蘋果表面有小範圍的受損區域但大部分仍保持完整時，模型容易將其誤判為健康蘋果。因此，我們針對這類受損特徵補充訓練數據，以強化模型對輕微缺陷的識別能力。

2.橘子誤判情形

針對橘子品質分類模型，我們發現因訓練時不同類別的樣本數量不均，導致模型傾向於將橘子分類為 A 類（完好）或 D 類（嚴重腐爛），而忽略 B 類（輕微受損）與 C 類（中度受損）。為改善此問題，我們增加 B、C 類的訓練樣本，並透過資料增強技術擴充數據，使模型學習更多中間品質橘子的特徵。此外，由於 B 類與 C 類的影像特徵相似，且在實驗中模型難以精確區分兩者，因此我們將兩類合併為新的 B 類，簡化分類標準，以提高模型判斷的穩定性。

經過這些改進措施，蘋果品質分類模型的準確率從初期的 50.6% 提升至 85.04%，而橘子品質分類模型在類別合併後，準確率提升至 97.67%，顯示改進策略有效提升了模型的分類能力。

（四）本研究與現有研究

相較於現有的水果品質檢測技術，本研究提供更輕便、低成本且高效的解決方案。傳統方法往往依賴大型設備或特殊相機，難以廣泛應用，而本研究透過影像辨識與深度學習技術，達成非破壞式檢測，無需額外購置高價設備，即可進行水果品質分類。與需破壞水果才能進行檢測的方法相比，本研究保留了水果完整性，適合即時與大量檢測需求。此外，本研究能夠根據水果的品質與種類進行分類，並提供適當的食用方式，使其在農業與食品產業中具備更高的應用價值。

二、未來展望

本研究的成果為水果品質檢測系統奠定了基礎，未來可進一步擴展。

（一）增加水果種類

本研究目前主要針對蘋果與橘子兩類水果進行品質分類。原先亦嘗試納入香蕉、芒果等其他品項，但受限於時間與資料量不足等因素，未能完成完整的訓練與驗證。未來，隨著研究的深入與數據的累積，可進一步擴展至更多種類的水果，以提升模型的適用性與準確性，使其在農產品檢測、品質評估等實務應用中發揮更廣泛的價值。

(二) 使用者介面

在未來的水果品質辨識系統中，使用者介面將直接影響用戶的操作體驗與系統的普及度。以下是我們對使用者介面設計的未來展望：

1.直觀化設計：

本系統將致力於提供直觀且易於操作的使用者介面，使不同年齡層與技術背景的用戶皆能輕鬆上手。透過簡潔明瞭的圖示、清晰的文字提示與流暢的操作流程，確保用戶能迅速理解並高效使用，進一步提升整體操作體驗與效率。

2. 多平台支援：

未來，將嘗試把水果品質辨識系統將擴展至多平台應用，包括行動應用程式與網頁版介面，以確保用戶能夠依據需求，在各種裝置上隨時隨地進行水果品質檢測。此舉將大幅提升系統的便利性與靈活性，滿足不同情境下的使用需求。

3. 個人化設定：

讓用戶能根據自身需求調整介面佈局、顯示內容與操作偏好。例如，用戶可以選擇顯示更多詳細數據，或是簡化介面以專注於特定的關鍵資訊。這將使系統更具彈性，滿足不同用戶的個別需求。

(三) 擴大應用範圍

系統可整合更多種類的水果檢測功能，擴大應用範圍至不同農產品，以滿足多元化的市場需求。

1.結合物聯網

將系統應用於智慧農業中，讓農場或工廠即時收到水果品質數據，實現遠程即時監控與品質管理，進一步推動農業數位轉型。

2.提升準確性於穩定性

透過持續改良機器學習模型與擴增影像辨識技術與資料量，提升系統的準確性與穩定性，使其能更精細地辨識水果的細微品質差異。

捌、參考資料

- 教育部（2021）。智慧創新競賽—12組成果專訪。教育部智慧創新專案網站。<https://ws-proj.moe.edu.tw/001/Upload/567/refile/13092/1472/c27fa84b-4abd-4ab5-acf1-1dd3b2cb916a.pdf>
- 財團法人精密機械研究發展中心（PMC）。（2022年10月13日）。食品原物料影像辨識技術，0.1秒幫你看出水果的甜。<https://www.pmc.org.tw/tw/service/show.aspx?num=125>
- 陳世芳（2021）。認識國產優良品牌水果認證制度。臺中區農業改良場。<https://www.tcdares.gov.tw/ws.php?id=1791>
- Felix Instruments. (2014). *F-750 Produce Quality Meter*. <https://felixinstruments.com/food-science-instruments/nir-spectroscopy/f-750-produce-quality-meter/>
- Vetrekar, N. T., Gad, R. S., Fernandes, I., Parab, J. S., Desai, A. R., Pawar, J. D., Naik, G. M., & Umapathy, S. (2015). Non-invasive hyperspectral imaging approach for fruit quality control application and classification: Case study of apple, chikoo, guava fruits. *Journal of Food Science and Technology*, 52(10), 6978 – 6989. <https://doi.org/10.1007/s13197-015-1838-8>
- 維基百科貢獻者。（2024年10月28日）。卷積神經網路。維基百科。<https://zh.wikipedia.org/zh-tw/%E5%8D%B7%E7%A7%AF%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C>
- Shiau, J.-Y., & Su, Y.-F.。（2024年5月11日）。ViT (Vision Transformer) 概述與優勢：對比 CNN 與 Swin 等 hierarchical 方法。Medium。 <https://medium.com/ai-blog-tw/vit-vision-transformer-%E6%A6%82%E8%BF%B0%E8%88%87%E5%84%AA%E5%8B%A2-%E5%B0%8D%E6%AF%94cnn%E8%88%87swin%E7%AD%89hierarchical%E6%96%B9%E6%B3%95-2e32c248255>
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. arXiv. <https://arxiv.org/abs/2103.14030>
- Chuquimarca, L. E., Vintimilla, B. X., & Velastin, S. A. (2024). A review of external quality inspection for fruit grading using CNN models. *Artificial Intelligence in Agriculture*, 14, 1 – 132. <https://doi.org/10.1016/j.aima.2024.05.001>
- Dümen, S., Yılmaz, E. K., Adem, K., & Avaroglu, E. (2024). Performance of vision transformer and swin transformer models for lemon quality classification in fruit juice factories. *European Food Research and Technology*, 250(6), 2291 – 2302. <https://doi.org/10.1007/s00217-024-04537-5>
- Organisation for Economic Co-operation and Development (OECD). (2024). *Apples*. https://www.oecd-ilibrary.org/agriculture-and-food/apples_421fcacc-en-de

[Data set]

- Abdul Nabi, K. (2022). *Fruit Classification (10 Class)*. [Data set] Kaggle. <https://www.kaggle.com/datasets/karimabdulnabi/fruit-classification10-class>
- Bali, P. (2022). *Apple*. [Data set] Kaggle. <https://www.kaggle.com/datasets/prabjotbali/apple>
- Bhatt, A. (2024). *Apple Dataset*. [Data set] Kaggle. <https://www.kaggle.com/datasets/ashishbhatt1234/apple-dataset>
- Hilton, D. (2023). *Apple Ripeness Levels Image Dataset*. [Data set] Kaggle. <https://www.kaggle.com/datasets/davidhilton/apple-ripeness-levels-image-dataset>
- Hossenabir, M. (2024). *Orange and Tangerine Images for Object Detection*. [Data set] Kaggle. <https://www.kaggle.com/datasets/mdismielhossenabir/orange-and-tangerine-images-for-object-detection>
- Huang, M.-L., & Chen, Y.-A. (2023). *A dataset of citrus fruit images* (Version 6) [Data set]. Mendeley Data. <https://doi.org/10.17632/bxfgvsn9kw.6>
- Maher, S. P. (2023). *Fruits Dataset (Images)*. [Data set] Kaggle. <https://www.kaggle.com/datasets/shreyapm/her/fruits-dataset-images>
- Marquis03. (2023). *Plants Classification*. [Data set] Kaggle. <https://www.kaggle.com/datasets/marquis03/plants-classification>
- Patil, K., & Meshram, V. (2022). *FruitNet: Indian fruits dataset with quality (Good, Bad & Mixed quality)* (Version 3) [Data set]. Mendeley Data. <https://doi.org/10.17632/b6fftwbr2v.3>
- Park, R. (2022). *Fruit Quality Classification*. [Data set] Kaggle. <https://www.kaggle.com/datasets/ryandpark/fruit-quality-classification>
- Roboflow. (2024). *Anomaly detection fruits dataset* [Data set]. Roboflow. <https://universe.roboflow.com/data-7my88/anomaly-detection-fruits/dataset/1>
- Subhan, M. (2024). *Fruit and Vegetable Disease (Healthy vs Rotten)*. [Data set] Kaggle. <https://www.kaggle.com/datasets/muhammad0subhan/fruit-and-vegetable-disease-healthy-vs-rotten>

【評語】052503

本作品基於人工智慧影像辨識之水果品質辨識及分類，運用 Swin-Transformer 架構，發展一套可辨識水果種類與品質等級的非破壞式檢測系統，嘗試解決現有水果檢測需破壞性或高成本設備問題。

未來精進發展建議：

1. 目前僅支援兩種水果（蘋果、橘子），可考慮增加多水果判斷訓練。
2. 品質分類精準度仍受限於資料不均與細部瑕疵混淆，建議提升中間等級樣本數。
3. 預測誤差主要來自蒂頭、品種色差與資料標註標準模糊，可增強資料多樣性。

作品海報

AI'S A IS FOR APPLE 基於人工智慧



影像辨識之水果品質辨識及分類



壹、摘要

本研究開發了一種基於影像辨識的水果品質分類系統，採用適合同時處理局部及全局資訊的Swin-Transformer模型，透過分析水果外觀來判斷其品質，並多次實驗以調整參數、訓練多個模型以辨識不同水果種類與品質。使用者上傳水果影像後，系統即能識別水果種類和品質，並提供新鮮度建議與食用方式。此系統相較於傳統的檢測方法，具備非破壞式檢測優勢，且降低成本，適用於農業生產與消費市場。模型測試結果顯示，水果類型分類器準確率為99.0%，蘋果品質分類器為85.0%，橘子品質分類器為97.7%，顯示該系統在分類與品質評估上具有較高的準確性，對水果檢測及提升食品管理有重要意義。

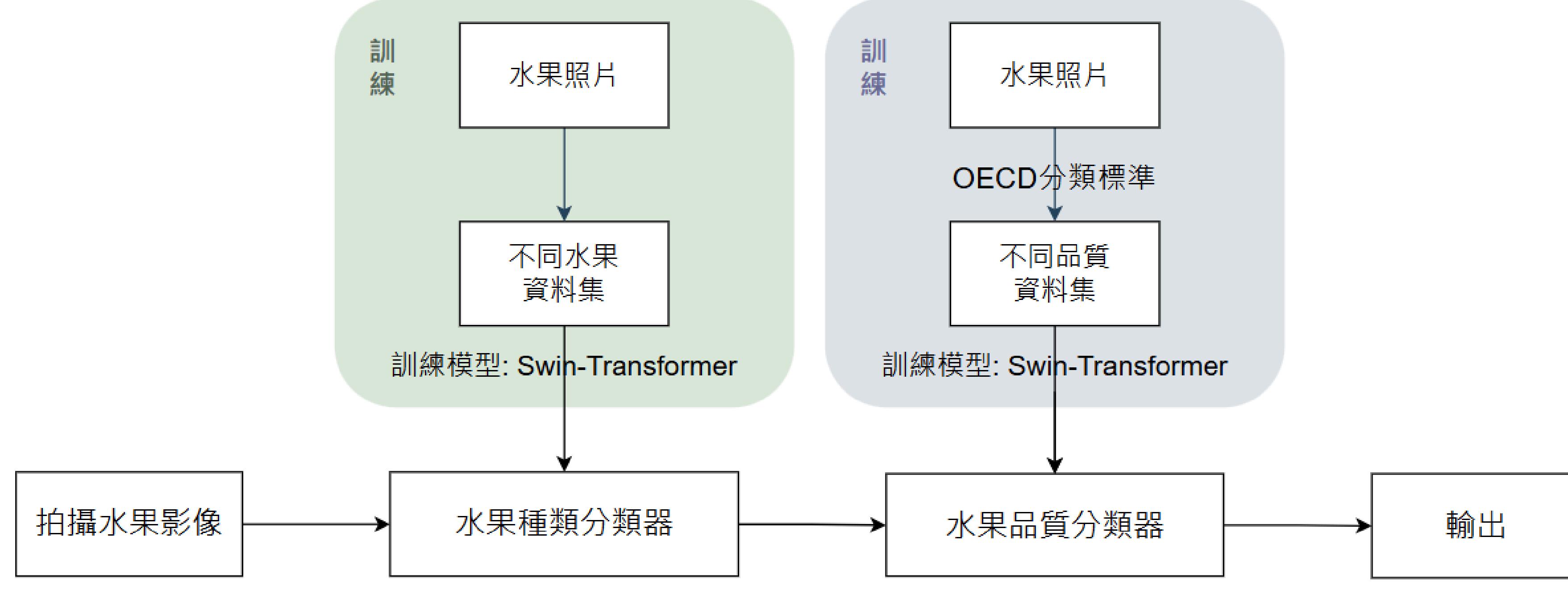
貳、研究目的

參、文獻回顧

- 一、使用影像辨識，進行非破壞式影像偵測。
- 二、將偵測到的不同水果依據種類進行分類。
- 三、將偵測到的水果依據品質優劣進行分類。
- 四、使系統根據水果的分類結果，提供適當的食用方式。

模型	CNN	ViT	Swin-Transformer
特徵學習方式	局部特徵學習	全局特徵學習	兼具局部與全局特徵
處理變形與視角變化	受限於局部特徵學習，對變形適應差	能較好處理長程依賴，對視角變化適應性較強	適應性強，透過滑動窗口機制提升對不同視角與變形的適應能力
水果種類辨識 (全局)	中等	強	強
品質瑕疵檢測 (局部)	強	中等	強

肆、研究流程圖



伍、研究過程與方法

一、訓練資料的收集與處理

OECD由優質到劣質將水果分成A,B,C,D四類：

A類：外觀完整，且顏色均勻，可食用

B類：輕微受損，或顏色較不均勻，可食用

C類：受損較為嚴重，處理後可食用

D類：嚴重受損，不可食用

這裡以蘋果為例：



二、Swin-Transformer 模型訓練

我們使用Swin Transformer的Swin-B版本進行微調訓練，以下是模型架構。

(一)Patch Embedding

這一部分的主要目的是將輸入圖像分割成小區塊，並將這些區塊映射到高維的特徵空間，成為可以進行自注意力運算的tokens。

1. 圖片分割 (Image Partitioning)

將輸入圖像 X 劃分為大小為 $P \times P$ 的小區塊

Patch 數量為： $N = \frac{H}{P} \times \frac{W}{P}$

每個 Patch 是一個 $P \times P \times C$ 的子圖，其中 C 是圖像的通道數。在本次實驗中，因使用 RGB 通道，故 C 為 3。

(二)特徵提取 (Feature Extraction)

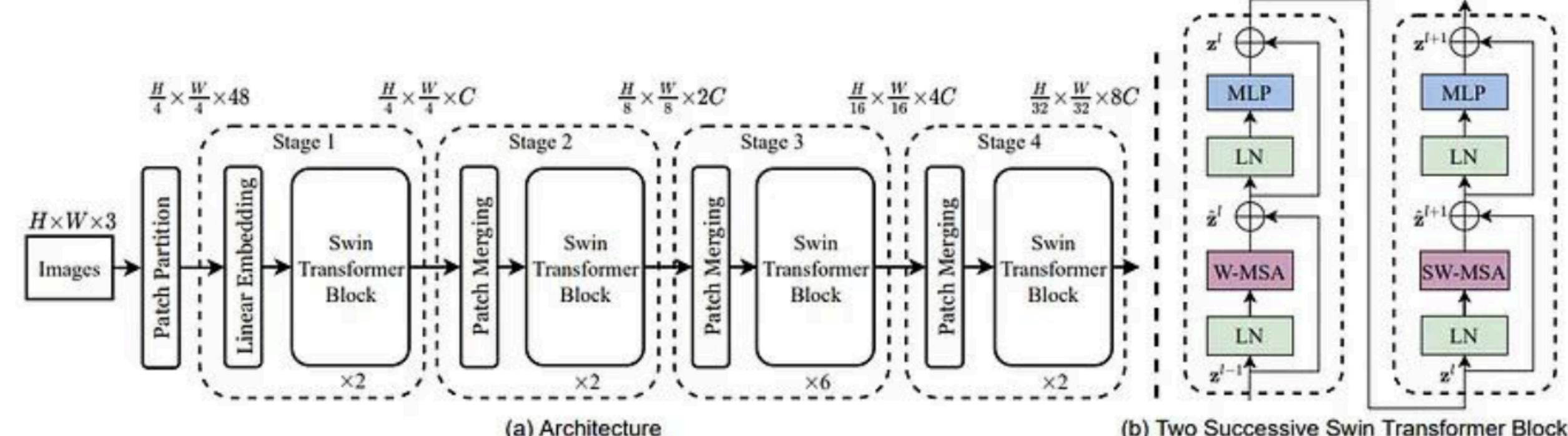
進入 Swin Transformer 之後，這些 token 經過多層處理，其中主要包含以下幾個關鍵步驟：

2. 展開每個 Patch(Flattening the Patch)

每個 $P \times P \times C$ 的 Patch 會被展平為一個一維向量。這樣每個 Patch 都會變成一個長向量，所有 Patch 的展平向量將組成一個矩陣 P 。 $P = [p_1 \ p_2 \ \dots \ p_N] \in \mathbb{R}^{N \times P^2 C}$

3. 線性投影(Linear Projection)

每個展平的 Patch p_i 會經過一個線性投影層將每個 Patch 映射到一個 D -維的嵌入空間。這樣，每個 Patch 會被轉換為一個維度為 D 的向量，並組成一個矩陣 Z_0 ，這些向量就是最終的 token。 $Z_0 = P \cdot W_p + b_p$



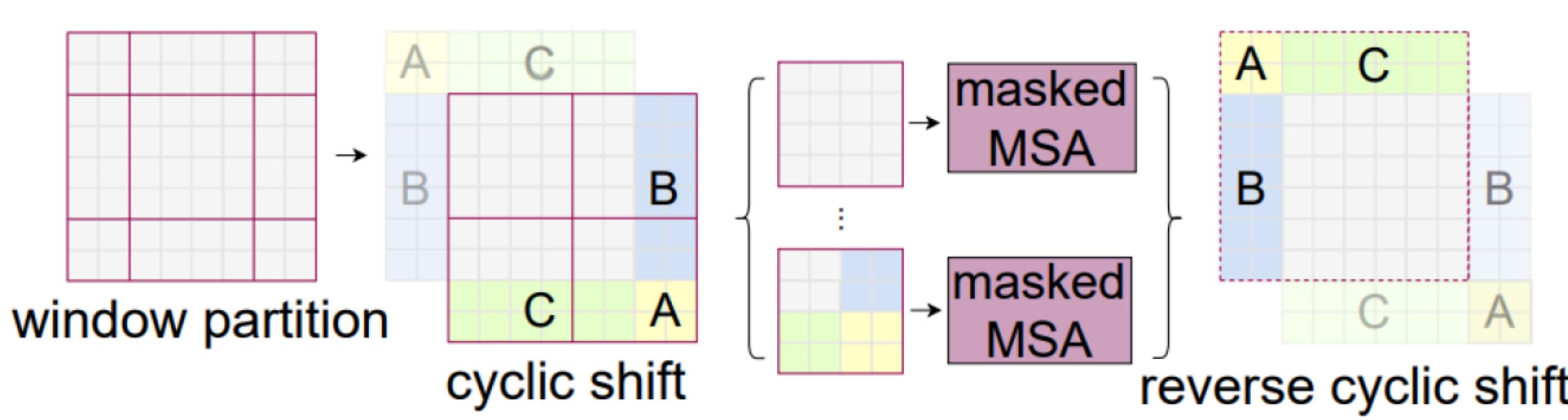
1. Swin Transformer Block

Swin Transformer Block 是 Swin Transformer 的核心運算單元，它負責處理圖像的特徵並提取局部與全局資訊。每個 Swin Transformer Block 主要包含兩個自注意力機制(W-MSA 和 SW-MSA)

2. 窗口自注意力(W-MSA)

W-MSA 的主要目標是降低傳統自注意力(MSA)的計算量，同時保持局部區域內的資訊學習。因此，Swin Transformer 採用「局部窗口」來計算自注意力，每個窗口大小為 $M \times M$ 。對於窗口內的 token，計算標準的 Scaled Dot-Product Attention:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}} + B\right)$$



3. 滑動窗口自注意力(SW-MSA)

雖然窗口化自注意力大大減少了計算量，但它會限制窗口內的特徵學習，並且不同窗口之間的資訊無法交互。為了解決這個問題，Swin Transformer 引入了移動窗口的概念。其操作方式如圖。

三、兩種訓練模型比較

模型	模型一：水果種類辨識	模型二：水果品質辨識
功能	提取多種水果圖像的特徵，並辨識不同水果種類	分析水果圖像的外觀特徵並評估水果的品質等級
類別	apple, orange	apple_A, apple_B, apple_C, apple_D
輸入	不同水果的照片	同水果不同品質的照片
輸出	具體的水果種類名稱	水果的品質等級

四、模型測試

透過測試程式來辨識模型判斷錯誤的圖片以提升模型準確率。程式說明如下：

- (一)透過 `parse_option()` 函式設置模型架構、資料路徑與權重檔等參數。
- (二)圖像預處理並統一尺寸 (224x224)、轉換格式並標準化，符合模型輸入需求。
- (三)根據設定建立模型，載入已訓練權重並切換模式，關閉梯度計算以節省資源與加快速度。
- (四)進行辨識，使用程式取得判斷結果，統計每類別預測數量並計算整體準確率及輸出結果。

```
TERMINAL
Image Name: orange_A(326).jpg, predict: orange_A
Image Name: orange_A(189).jpg, predict: orange_A
Image Name: orange_A(494).jpg, predict: orange_A
Image Name: orange_A(477).jpg, predict: orange_A
Image Name: orange_A(438).jpg, predict: orange_A
Image Name: orange_A(149).jpg, predict: orange_A
Image Name: orange_A(103).jpg, predict: orange_A
Image Name: orange_A(509).jpg, predict: orange_A
Image Name: orange_A(309).jpg, predict: orange_A
Accuracy: 97.36842105263158
照片總數: 532
預測為orange_A orange_B orange_C的照片數量: [518, 14, 0]
```

五、模型使用與使用者程式

透過自行撰寫的程式自動更改參數及路徑，程式分成兩階段，分別執行種類與品質的辨識，使用時，僅需將圖片上傳至指定資料夾用以開始辨識。

- (一)第一階段：水果類別辨識，程式會測試指定路徑中的照片並輸出其對應的水果種類。
- (二)辨認完水果種類後，透過 `if/else` 條件式，將第二階段的變數及模型設定成該種水果的模型權重路徑與類別。
- (三)第二階段：水果品質辨識，程式會辨識指定路徑中的照片並輸出其對應的水果品質。
- (四)輸出結果與反饋：
根據兩個模型的預測結果，利用輸出建議之食用方法。

```
# **根據第一階段分類結果，載入對應的模型**
if fruit_class == 'apple':
    trained_model_path = "/media/ibm/F/pinyan/Swin-Transformer-main/output/swin_base_fine_classes = ('apple_A', 'apple_B', 'apple_C', 'apple_D')"
else:
    trained_model_path = "/media/ibm/F/pinyan/Swin-Transformer-main/output/swin_base_fine_classes = ('orange_A', 'orange_B', 'orange_C')"

# 載入對應模型
fine_model = load_model(config, trained_model_path)
```

```
Image Name: apple_A(3).jpg, predict: apple
Final classification: apple_A
它長得超完美啦！適合食用以及送禮！
```

陸、研究結果與討論

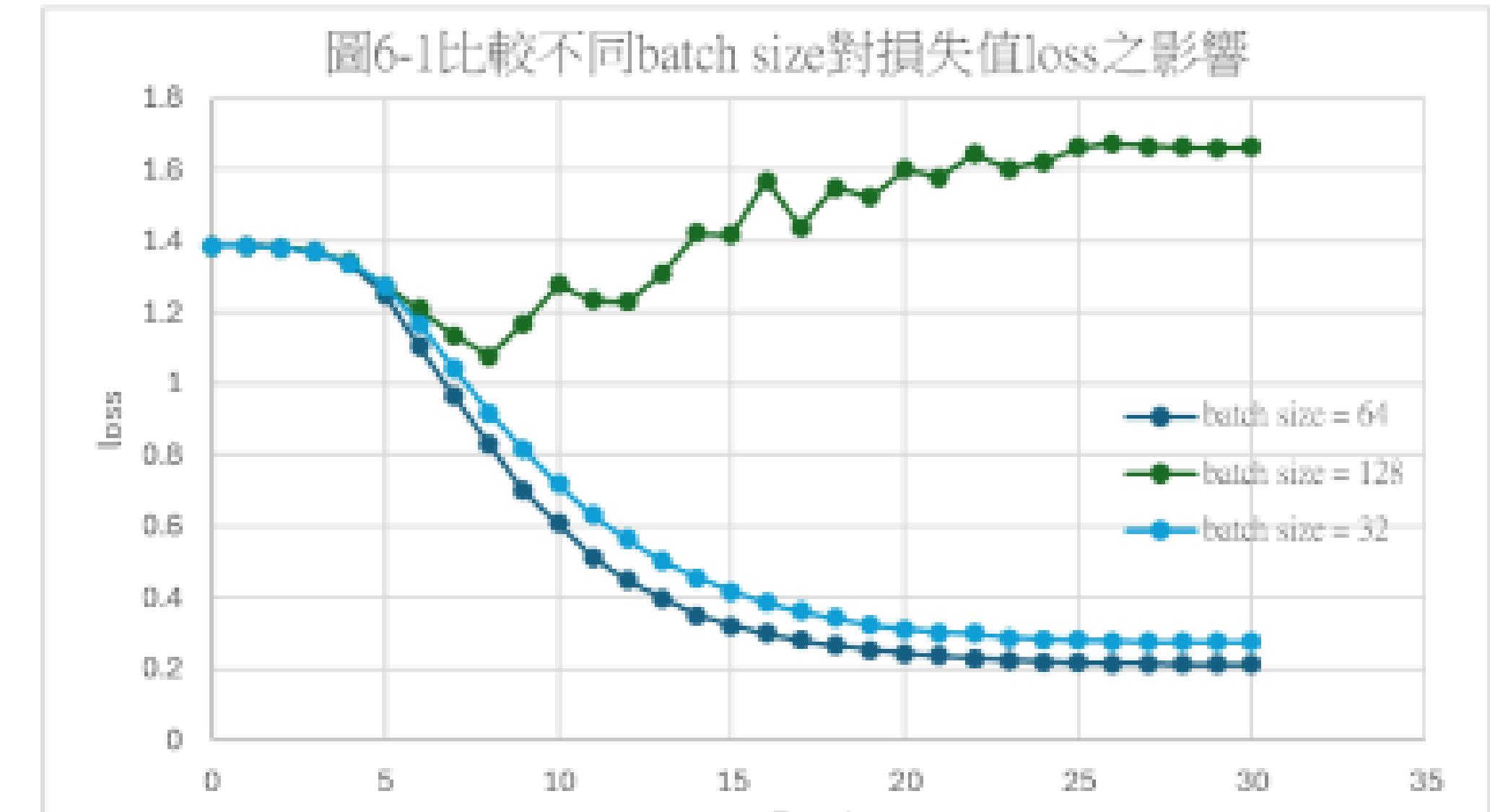
一、超參數調整結果

當 Batch Size 設定為 64 時，模型的 loss 值達到最小。

不同 batch size 之比較

Batch Size	Batch Size = 32	Batch Size = 64	Batch Size = 128
最終損失值	0.2744	0.2133	1.6621
模型準確率	84.488%	85.042%	85.596%

說明
模型更新時樣本較少，在學習特徵與維持梯度樣本較完整但更新頻率導致梯度更新穩定，影更新的穩定性中取得平衡，達到最優損失值。響模型收斂效果。



二、蘋果模型準確率之分析與討論

(一) 將蒂頭誤判成黑斑缺陷

推測是由於中間蒂頭部分和黑斑相近，而讓模型將蒂頭誤判成黑斑缺陷。因此，我們將這類照片進行資料增強，將其進行旋轉、平移、上下或左右顛倒，增加訓練照片數，使模型學習到此特徵。



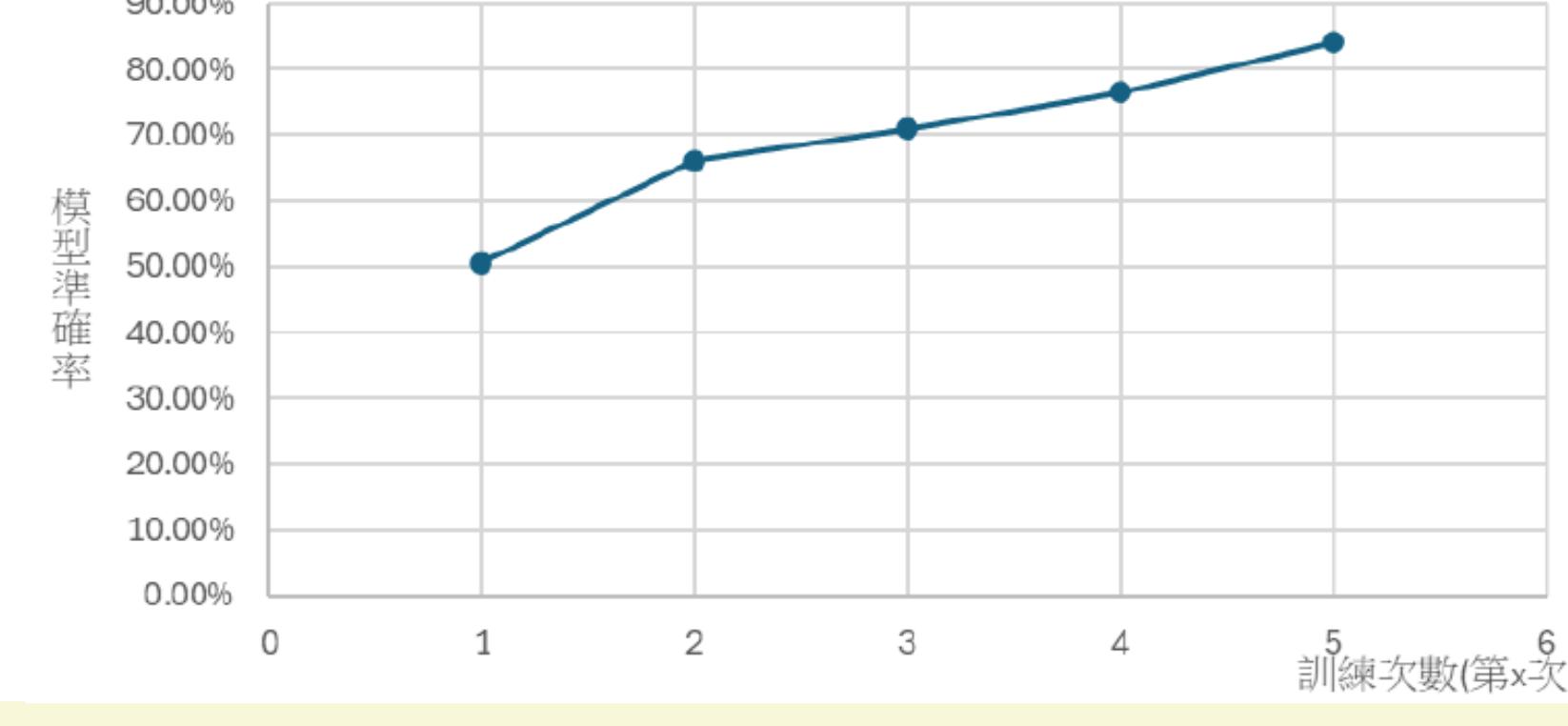
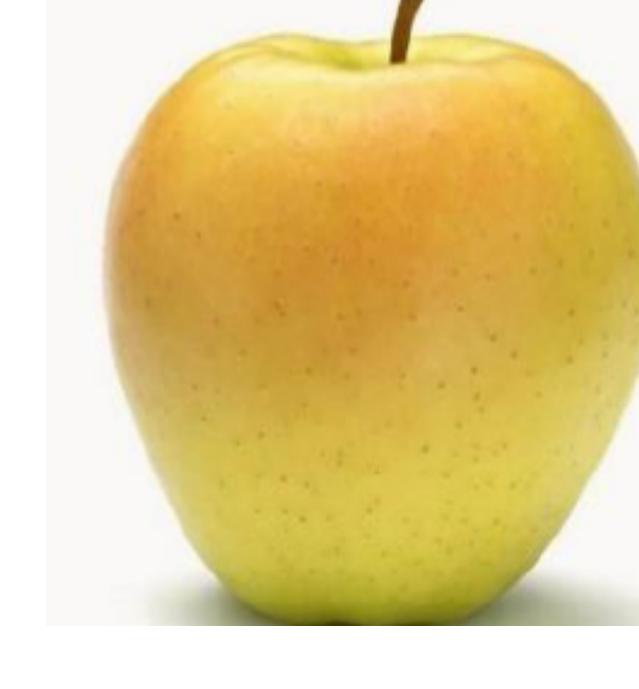
(二) 輕微疤痕判斷錯誤

我們發現部分被分錯的蘋果雖然有黑斑或是爛掉，但紅色部分還是佔有一定面積，因此推測是因紅色部份過多而讓模型判斷錯誤。



實際類別	apple_C	apple_D	apple_D	apple_D
程式判別	apple_B	apple_C	apple_C	apple_A

圖6-10 蘋果模型準確率折線圖



(三) 不同品種蘋果

不同品種的蘋果，會因為顏色的緣故使電腦誤判。

(四) 錯誤影像分析之改良結果

經過重複測試訓練並改善後，我們將準確率從初次的 50.6% 逐步提升至最終 97.7%。

橘子品質辨識模型表現

橘子類別簡化後結果

	orange_A	orange_B	orange_C	orange_D		orange_A	orange_B	orange_C
訓練照片數	200	50	60	200	訓練照片數	600	600	550
驗證照片數	100	11	29	100	驗證照片數	200	200	200
測試準確率	100.0	0.0	0.0	89.27	測試準確率	96.99	95.54	90.46

三、橘子模型趨於二分法之原因討論

- 由於在訓練時給模型的照片數過於不平均，B與C類照片過少，導致模型只可明確分出A類與D類的橘子
- 在增加B與C類照片時，發現數量嚴重不足，因此作者合併B與C類，將橘子簡化為A,B,C 三類。
- 增加資料量並重新訓練後，最終將橘子品質辨識模型的準確率提升至97.67%。

橘子品質辨識模型表現

橘子類別簡化後結果

一、總結

(一) 水果訓練資料集

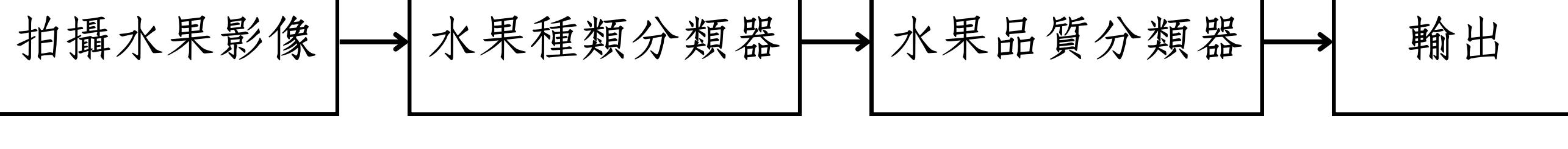
本研究蒐集多種水果照片，先依據種類分類，再參考 OECD 的分類方法，根據不同水果的標準進一步細分品質等級。最終，本研究提供多組標註完善的資料集，以利模型進行訓練與應用。

(二) 模型訓練結果

當batch size = 64時，模型學習效果最佳。

模型	水果種類分類器	蘋果品質分類器	橘子品質分類器
準確率	99.0%	85.0%	97.7%

(三) 模型使用與系統運作



(五) 本研究與現有研究之比較

相較於現有的水果品質檢測技術，本研究透過影像辨識與深度學習技術，達成非破壞式檢測，無需額外購置高價設備，即可進行水果品質分類。此外，本研究能夠根據水果的品質與種類進行分類，並提供適當的食用方式，使其在農產品檢測、品質評估等應用中發揮更高的應用價值。

(四) 誤判情形討論與改善

1. 蘋果誤判情形討論

- 透過資料增強技術，改善將蒂頭誤判成黑斑缺陷的問題
- 透過補充相同特徵的訓練數據，強化模型對輕微缺陷的識別能力

2. 橘子誤判情形討論

- 由於訓練時不同類別的樣本數量不均，導致模型傾向於將橘子分類為 A 類或 D 類，而忽略 B 類與 C 類。
- 由於 B 類與 C 類的影像特徵相似，且數量不足，因此將兩類合併為新的 B 類，簡化分類標準，提高各類別準確率

二、未來展望

(一) 增加水果種類

(二) 使用者介面

(三) 擴大應用範圍

捌、參考文獻

- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. arXiv. <https://arxiv.org/abs/2103.14030>
- Organisation for Economic Co-operation and Development (OECD). (2024). Apples. https://www.oecd-ilibrary.org/agriculture-and-food/apples_421fcacc-en-de