淡江大學

專題題目:

ESG蔬食辨識

專題成果報告書

組員名單:

資工4C 周聖庭

資工4A 蔡宗佑

資工4B 姚昌佑

**目錄**

動機與目的 2

研究背景 2

研究目的 2

使用技術 2

內容描述 5

資料集收集 5

實現方式 5

結論 8

**動機與目的**

**研究背景**

近年來由於AI快速發展，使得AI開發工具更加普及和易於使用，同時GPGPU的普及使得大規模平行運算變得更加可行，為深度學習等AI算法的訓練提供了強大的硬件支持，大家可以較容易的用深度學習訓練自己的模型，這些因素促進了AI社群的蓬勃發展，吸引了越來越多的開發者和研究人員投身於AI領域。

**研究目的**

本專題利用深度學習，實現以往難以實現的一項功能，具體而言，我們的目標是判斷食物圖片是否為蔬食，我們會運用深度學習模型，並利用大量的食物圖片數據集進行訓練。

**使用技術**

**CNN**

卷積神經網路（CNN）是一種深度學習模型，專門用於處理具有網狀結構的數據，如圖像和影片。CNN模擬了人類視覺系統的工作原理，通過一系列的卷積層和池化層來提取圖像特徵。卷積層使用卷積運算將圖像中的特徵提取出來，這些特徵可以是邊緣、紋理或模式等。接著，池化層對提取的特徵進行壓縮和摘要，減少計算量和參數量。最後，通過全連接層和激活函數將提取的特徵映射到輸出空間，進行分類或回歸等任務。CNN具有局部感知性和權重共享等特點，使其在圖像識別、物體檢測和分割等領域取得了巨大成功。例如，在圖像識別任務中，CNN可以通過學習大量圖像數據中的特徵模式，來識別和分類不同的物體或場景。總的來說，CNN以其強大的特徵提取能力和良好的泛化性能，成為了許多視覺任務的首選模型之一。

**relu**

在深度學習中，損失函數用於衡量模型預測與實際目標之間的差異。對於分類任務，常見的損失函數包括交叉熵損失函數（cross-entropy loss），特別是對於多類別分類問題。對於回歸任務，通常使用均方誤差損失函數（mean squared error loss）。

至於 activation='relu'，這是指在神經網路的各層中所使用的激活函數。ReLU（Rectified Linear Unit）是一種常用的激活函數，它能夠在輸入為正時保持線性關係，從而減緩了梯度消失的問題，並且計算速度快。當然，對於不同的任務和模型結構，也可以使用其他的激活函數，如Sigmoid、Tanh等，這取決於具體的應用場景和需求。

**正規化**

正規化是一種對資料的預處理，按照某種函數把數值映射到正規化空間，函數可以是線性、log、對數之類，最常見的是Min-Max正規化，以線性方式縮放到一個固定的範圍內，通常是[0,1] 或[-1,1]，而Z-score正規化通過減去均值並除以標準差則沒有固定範圍，正規化不一定是1對1函數，函數可能會經過離散化、裁減掉極值的處理。

正規化的重要性體現在它有助於處理數據中的不同尺度，使得模型更具強健性(Robustness)，例如在某些機器學習算法中，如果不進行正規化，那些具有較大數值範圍的特徵可能會在模型訓練過程中主導著對較小範圍的特徵的學習，從而影響模型的性能。

**VSCode**

VSCode是一個微軟開發開源免費的程式碼編輯器，VSCode可以在Windows、macOS、Linux上執行。

VSCode的主要特點之一是它的多語言支持。它能夠辨識並高亮顯示多種程式碼，包括但不限於JavaScript、Python、Java、C++等，這提供了一個統一的編輯環境，方便開發者在不同項目中切換，提高工作效率。

VSCode提供了豐富的自動完成功能，它支援各種程式碼的分類高亮，讓代碼更易讀，同時具有智能的代碼建議，幫助開發者更迅速地撰寫代碼。

VSCode的擴展生態系統是它受歡迎的原因之一，可以透過安裝擴展插件來擴展其功能，這些擴展涵蓋了各種方面，如語言支持、版本控制、調試、主題風格等，這使得VSCode成為一個高度可定制的工具。

調試是VSCode的一大強項，它支援多種程式語言的調試，提供了豐富的調試功能，包括設置斷點、單步執行、監視變數、查看呼叫堆疊等，調試控制台則提供了實時的反饋，幫助開發者更容易追蹤和解決問題。

**Python**

Python是一種物件導向、直譯式、開源免費的高階程式語言，由Guido van Rossum在1991年發明，現在已是全球流行的程式語言之一，特色是簡單易學，不需要編譯，有豐富的標準函式庫和第三方模組，可以在Windows、Linux、macOS等主流系統上執行。

Python內建高級資料結構，如列表、字典和集合，使Python非常適合進行數據處理和操作

Python的擴展性使其能夠與其他語言輕鬆集成。開發者可以使用Python/C API或Cython等工具擴展Python的功能，也可以透過各種橋接工具實現與其他語言的互通，這使得Python在整合現有代碼、加速性能等方面具有更大的靈活性。

**遷移學習**

對於網路上找不到的圖片種類，可以利用遷移學習，使用stable diffusion生成寫實風格的食物圖片，加入資料集。

資料集擴增:

對收集到的所有圖片進行預處理,包括水平翻轉、垂直翻轉、旋轉等,可以增加模型的泛化能力。

用測試集測試模型:

使用自己拍的食物照片或不包含在資料集的圖片測試訓練完的模型，調整辨識是否判斷為疏食的閾值。

持續優化:

往訓練集補充更多沒包含的圖片種類，追蹤新訓練完的模型在後續的表現。

**TensorFlow**

TensorFlow是一個開源的機器學習框架，由Google開發並於2015年釋出。它提供了一個強大的平台，用於構建和訓練機器學習模型，包括神經網絡。TensorFlow的名稱來自於它的核心概念"tensor"，它是多維數組的數學對象，在計算機科學中常用於表示數據。TensorFlow提供了一個靈活而強大的框架，使得開發者可以輕鬆地建立各種類型的機器學習模型，包括圖像識別、語音識別、自然語言處理等。

TensorFlow採用了一個靜態圖的概念，這意味著用戶首先定義計算圖，然後將數據通過該圖流動以進行計算。這種方式使得TensorFlow能夠進行高效的計算和自動微分，這對於訓練複雜的神經網絡模型特別有用。

TensorFlow提供了豐富的工具和庫，使得開發者能夠更加方便地進行模型的構建、訓練和部署。例如，TensorFlow提供了高級API，如Keras，使得構建和訓練神經網絡變得更加容易。

TensorFlow是一個強大的機器學習框架，它為開發者提供了一個靈活且高效的平台，用於建立各種類型的機器學習模型，從而推動了機器學習和人工智能的發展。

**Early stopping**

Early stopping 是一種用於監控機器學習模型訓練過程的技術，旨在防止過度擬合並提高模型的泛化能力。在訓練過程中，模型的性能通常是根據驗證集的表現來評估的。Early stopping 通常會在模型在驗證集上的性能開始下降時停止訓練，而不是等到模型在訓練集上的性能下降。

Early stopping 的工作原理是定期檢查模型在驗證集上的性能，當性能在一定數量的連續迭代中沒有改善時，訓練過程就會被停止。這個數量通常被稱為 "patience"，即容忍度，它是一個超參數，用於控制在模型停止之前允許性能下降的次數。當模型的性能在設定的容忍度內都沒有提升時，Early stopping 會將訓練過程提前終止，從而避免了模型對訓練數據的過度擬合。

Early stopping 是一種有效的防止過度擬合的技術，它可以提高模型的泛化能力並改善模型的訓練效率。通過使用 Early stopping 技術，開發者可以更好地控制模型的訓練過程，提高模型的性能並加快模型的收斂速度，從而更有效地應用機器學習模型於各種任務中。

**資料集收集**

目標是訓練一個辨識圖片是不是蔬食的模型，輸入整張圖片，直接判斷是不是蔬食。

定義蔬食:

蔬食指包含蔬菜或水果的食物，不能包含任何肉類，蛋、魚肉、烘焙食物、飲料和冰塊都不算是蔬食。

收集圖片資料:

在網路上搜尋蔬食和非蔬食的相關圖片，可以使用關鍵詞像是 vegtable、vegetarian、meat dishes、non-vegetarian food 等進行搜尋。可以從免費圖片資料庫中下載圖片，例如 Unsplash、Pexels、Pixabay 等。也可以使用現成的免費食物圖片資料集，例如 Fruits-360，盡量包含所有食物種類圖片。

**實現方式**

我們建立了一個蔬食辨識模型，旨在區分圖像中的蔬菜和非蔬菜。模型採用了卷積神經網絡（CNN）架構，具有三個卷積層和池化層，以及兩個全連接層。每個卷積層之後都有激活函數 ReLU，以及部分層中的批量標準化和 Dropout，以減少過擬合。為了防止學習過快，使用了指數衰減的學習率。訓練過程中，將數據集分為訓練集和測試集，並通過數據增強來擴充訓練數據。而後，使用二元交叉熵作為損失函數，並在模型訓練過程中使用了早停法。通過模型的訓練和測試，在評估了模型的性能並繪製了準確度和損失的曲線圖。最終，將訓練好的模型保存下來，以便後續使用。

**模型架構**

1. 卷積神經網絡（CNN）設計：
   * 各層的類型、激活函數、濾波器數量、濾波器大小等參數設置。
   * 解釋每一層的作用以及為什麼選擇這樣的結構。
2. 全連接層及正規化：
   * 神經元的數量、激活函數等。
   * 說明正規化（Dropout）的作用和影響。

**學習率衰減策略**

1. Exponential Decay 方法：
   * 說明指數衰減策略的原理，包括初始學習率、衰減步數和衰減速率的設置。
   * 分析這種策略對模型訓練的影響，以及為什麼選擇使用它來優化模型。

**數據增強方法**

1. ImageDataGenerator 使用：
   * 解釋數據增強的目的和原理。
   * 詳細描述你使用的各種數據增強技術，如縮放、剪切、翻轉、旋轉等。
   * 討論這些技術如何幫助提高模型的泛化能力和性能。

**早停法策略**

1. Early Stopping 回調：
   * 解釋早停法的原理和作用。
   * 描述你使用的早停法的設置，包括監控的指標、耐心值等。
   * 討論早停法對模型訓練的影響，以及它如何幫助防止過擬合。

| 層數 | 類型 | 激活函數 | 濾波器數量 | 濾波器大小 | 池化大小 | 丟棄率 |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 卷積層 (Conv2D) | ReLU | 64 | 3\*3 | - | - |
| 2 | 池化層 (MaxPooling2D) | - | - | - | 2\*2 | - |
| 3 | 卷積層 (Conv2D) | ReLU | 128 | 3\*3 | - | - |
| 4 | 池化層 (MaxPooling2D) | - | - | - | 2\*2 | - |
| 5 | 卷積層 (Conv2D) | ReLU | 256 | 3\*3 | - | - |
| 6 | 池化層 (MaxPooling2D) | - | - | - | 2\*2 | - |
| 7 | Flatten | - | - | - | - | - |
| 8 | 全連接層 (Dense) | ReLU | 215 | - | - | - |
| 9 | Dropout | - | - | - | - | 0.5 |
| 10 | 全連接層 (Dense) | Softmax | 2 | - | - | - |

**名詞解釋**

1. 層數（Layer）：指模型中的每一個層級。從輸入層到輸出層的過程中的每個階段都稱為一層。

2. 類型（Type）：指該層的類型，通常有卷積層、池化層、全連接層等。

3. 激活函數（Activation Function）：在神經網絡的每一層中，激活函數用於引入非線性特性，以幫助模型學習複雜的模式。

4. 濾波器數量（Number of Filters）：指卷積層中使用的濾波器（也稱為卷積核）的數量。濾波器用於提取圖像中的特徵。

5. 濾波器大小（Filter Size）：指卷積層中每個濾波器的大小，通常以高度和寬度表示。

6. 池化大小（Pooling Size）：指池化層中使用的池化窗口的大小，池化操作用於減少特徵圖的尺寸，同時保留重要信息。

7. 丟棄率（Dropout Rate）：指在 Dropout 層中隨機丟棄的神經元比例。Dropout 是一種正則化技術，有助於防止模型過擬合。

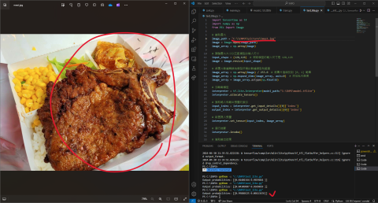
**結論**

**後臺測試**

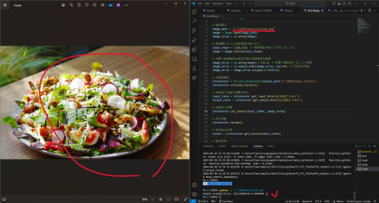
class 0: noveg(非蔬食) class 1: veg(蔬食)

越接近1代表AI判斷次圖越接近此Class

[[1,0]]代表AI判斷為非蔬食可能信較高(信心指數)



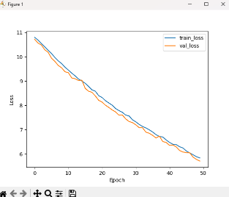
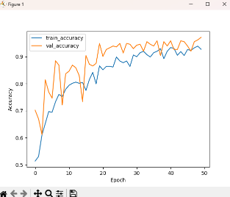
[[0.9988625 0300113476]]



[[0.01401564 0.9859844]]

「train\_accuracy」和「val\_accuracy」表示模型在訓練集和驗證集上的準確率，而「train\_loss」和「val\_loss」是訓練集和驗證集上對應的損失值。 這些指標用於評估模型的表現和訓練期間的收斂性。

train\_accurary和val\_accuracy train\_loss和val\_loss



**前臺測試**

這是將AI嵌入Android studio的結果，適用於Android 8~14。

