

中華民國第 64 屆中小學科學展覽會 作品說明書

科 別：電腦與資訊學科

組 別：高級中等學校組

作品名稱：開發新型演算法於探究異常值在迴歸分析中的影響與應用
－分析照「騙」

關 鍵 詞：激活函數、深度偽造、深度學習

編 號：

摘要

本研究主要分成兩大主軸，第一先深入探討迴歸分析中異常值對模型的影響，發現一般的模型在面對數據中的異常值會降低其預測的能力，因此藉此特性，我們結合了 K-近鄰演算法（K-NN）、模糊理論與深度學習等，自創了新穎的演算法進而找出數據中的異常值；第二為應用此演算法來分析與處理圖像中「不正常區塊或雜訊的輪廓」，達到抑制其對整體美觀的影響，即修復圖像，再進一步分析深度偽造（Deepfake）所合成的圖像。研究結果顯示，相較於現有的模型，我們的演算法較容易處理異常值的影響且分析圖像的應用之表現更佳，未來的研究方向可能包括進一步評估其他演算法的效能與將其優化以提高其效能和實用性。

壹、前言

一、研究動機

隨著資訊科技不斷地推陳出新，資訊分析與處理已成為各學科領域中不可或缺的核心元素。在這個發展趨勢下，資料視覺化和迴歸分析成為處理大量數據的重要工具，而資料中的異常值往往會是分析與處理過程中要被排除的一大挑戰。但我們來說，剛好相反，我們反而想深究其影響，因為若能掌握住資料中異常值的存在，便能善用其影響的範圍，因此我們進一步用來應用於圖像的分析，近年來也因 Deepfake 技術的崛起，它可用來創建高度逼真的虛假影像，這對個人隱私、社會信任和公共安全構成了潛在威脅，其被廣泛的應用也引發了許多法律和倫理問題，例如版權、肖像權、誹謗和隱私侵權等，可見這個技術的潛在威力不可小覷，所以我們希望能在兼顧高效性和精確性的前提下完成新型的演算法，此法的獨特之處在於其能夠先彰顯異常值的影響並成功找出其影響的範圍，即圖像中「偽造的區域」，而進一步辨識圖像的真實性，本研究的目的是在於提供一套有效的檢測和防範手段，保護個人和社會免受虛假信息的侵害，有望對未來資訊分析領域帶來深遠的影響。

二、研究目的

我們為了深入了解迴歸模型在面對數據中異常值的表現，致力於設計找尋被異常值影響的數據之演算方法，以便應用於圖像處理，將研究目的細分成下列幾項：

（一）研究不同深度學習模型

了解與學習深度學習模型於處理與分析圖像之應用，特別是針對多層感知器（MLP）、卷積神經網絡（CNN）、U-Net 與 VGG16 等四個模型。

（二）開發新穎演算法並應用於修復與辨識模型

探究 K-近鄰演算法（K-NN）、模糊理論與深度學習並設計新的演算法，須在兼顧效率和準確性的情況下，能用此法順利找出異常值的存在與影響的範圍，藉此發想圖像的修復與辨識的應用。

（三）探索深偽圖像的特徵

結合回歸分析與深度學習，深入研究 Deepfake 圖像的特徵模式，了解哪些特徵對辨識結果有顯著影響，以及這些特徵之間的關聯性，從而提供更具解釋力的分析結果，應用於解決圖像的修復效果。

（四）辨識精準度之評估

透過深度學習的方式來評估 Deepfake 辨識模型並能夠有效區分真實影像和虛假圖像，進而提高辨識系統的準確性和可靠性。

三、文獻回顧

迴歸分析一直是統計學和數據分析領域中的重要議題 [1]，但在真實的數據（Real data）中，往往會存在著雜訊和異常值 [2]，可能對迴歸模型的估計結果產生不良影響，文獻 [3] 中明確強調了處理訓練資料中異常值的三個方法，首先，第一種方法是透過預先處理數據（Data Preprocessing），排除異常值，但目前缺乏明確的標準來區分或過濾。第二種方法是樣本改進技術（Sample Polishing Techniques），其目的是在構建模型之前校正受損的數據，但這伴隨著時間大量的消耗，且僅適用於少量數據。第三種方法則是試圖增強訓練模型的穩健性。故在這次的作品中，我們嘗試挑戰要設計出一套找出異常值與其影響的範圍之演算法並結合深度學習，應用於Deepfake圖像的分析，本研究所需要自學探究與文獻蒐集的方面大致分為以下

幾個方向：

（一）迴歸分析

迴歸分析是統計學中一種探討變數間關係的方法 [4]，其主要目的在於預測一個變數（被稱為因變數）如何受到一個或多個其他變數（被稱為自變數）的影響。透過建立一個數學模型，迴歸分析致力於描述和解釋因變數與自變數之間的互動關係，進而進行預測或推論。而依據自變數和因變數之間的關係，迴歸分析可區分為線性迴歸分析和非線性迴歸分析兩大類別，本次作品會以後者為我們用來評估Deepfake辨識模型。

（二）K-近鄰演算法（K-NN Algorithm）

KNN是一種監督式學習方法，通常被用來做分類和迴歸。它的原理是基於相似的事物在特徵空間中通常靠近的概念 [5]。當有一個新的沒有標籤的數據點時，KNN會找到最接近它的K個已經標記的數據點，然後根據這些鄰近點的標籤進行預測。使用KNN的步驟依序為選擇K值、選擇距離度量（例如歐氏距離、曼哈頓距離、切比雪夫距離與明可夫斯基距離等）、計算點與點的距離、識別最近的鄰近點與進行預測等。這裡我們就是利用「識別最近的鄰近點之距離」的特色來計算數據的密集程度，藉此設計出「自動萃取出異常值」之演算法。

（三）模糊理論（Fuzzy Theory）

Fuzzy 一詞即指模糊的事物，模糊理論是一門處理不確定性和模糊性的數學工具和理論，根據 [6]，有提及其主要目標是處理那些難以用傳統的精確邏輯或集合理論準確描述的問題，此理論在各個領域都有廣泛的應用，包括人工智慧、控制系統、模式識別與信息檢索等，我們利用了此領域所教的知識「模糊集合」與「隸屬函數」設計一個「數據資料本身自動給予權重」的方法，此方法可弱化異常值之影響，大大地提高辨識圖像中異常區域的準確性。

（四）深度學習（Deep Learning）

此為機器學習的一個重要分支，其為透過模仿人類大腦的神經網路結構來處理和分

析數據的一門技術 [7]，它的核心是人工神經網路，其中每一層都由許多神經元組成，而這些神經元再透過激活函數與權重連接成網路，進而模擬生物神經元的工作方式。網路通常包含多個隱藏層，當層數增加時，使得網路能夠學習並提取到數據中更複雜的特徵和模式，從而實現更高水準的數據分析和預測能力。損失函數用於衡量模型預測值與真實值之間的差異，而梯度下降法則是一種優化算法，通過最小化損失函數來調整網路權重，以提高模型性能。這些技術已在許多領域廣泛應用，然而這次的研究當中，我們將其模型應用在圖像的分析與處理，對於圖像修復的應用會使用到多層感知器（MLP）、卷積神經網路（CNN）與U-Net等模型，而對於辨識深偽圖像的應用會使用到VGG16模型 [8]。

（五）深偽技術（Deepfake）

此技術簡稱為「深偽」，英文命名為「Deepfake」，其分別來自兩個英文單字「deep learning」與「fake」，在圖像合成的技術中，Deepfake是一種結合人工智慧和深度學習技術來生成或修改視覺的方法，其核心技術為利用生成對抗網路（GANs），該網路由生成器和判別器兩部分組成，生成器建構逼真的虛假內容，而判別器則嘗試區分真實和虛假的內容，最後透過這種相互競爭的過程，生成器能夠生成極度逼真的圖像，以至於人眼難以辨別其真偽；然而隨著此技術的發展，如何在法律和倫理層面進行有效的監管和管理成為關鍵議題，這需要建立一套完整的法律框架和技術手段，以防止技術濫用並保障公眾利益 [9]，因此在本研究中，我們設計一套辨識其圖像的真偽之演算法。

（六）模型評估之指標

針對圖像修復的模型，根據 [10] 與 [11] 之模型預測的相關討論中，有提及到均方根誤差（Root Mean Squared Error, RMSE）是一種用來評估迴歸模型預測能力的指標，而在圖像重建或圖像壓縮中，RMSE可以用來衡量重建圖像與原始圖像之間的差異，從而評估圖像處理算法的效果 [12]，如果單獨地使用RMSE值並不能直接衡量模型辨識的精度之優劣，必須要透過對比不同模型的RMSE值；而對於辨識深偽圖像的模型評估，我們會使用到AUC值（Area Under the Curve），它是衡量二元分類模型性能的一個重要

指標 [13]，特別是在處理不平衡數據集時具有顯著優勢。AUC的英文學名為「Area Under the Receiver Operating Characteristic Curve」，簡單來說就是 ROC 曲線下的面積，它反映了模型區分正例（Positive）和反例（Negative）的能力。當AUC值為 0.5 時，表示模型與隨機猜測差不多，當AUC值大於或等於0.9，表示模型具有非常好的辨識能力 [13]，因此在不同的模型中，我們將利用此指標來評估與比較Deepfake辨識的能力，以凸顯我們演算法的優勢。

（七）DFDC 數據集

Dolhansky等人 [18] 於2020年在Kaggle平台上組織了DFDC（DeepFake Detection Challenge）競賽，並釋出了DFDC數據集。DFDC數據集目前（2020年）是最大的公開可用的人臉交換視頻數據集，包含來自3426名有薪演員，共計超過 100,000 筆影像片段，使用了多種深度偽造、基於GAN的和非學習方法進行製作。根據競賽結果，儘管深度偽造檢測非常困難且仍然是一個未解決的問題，但僅在DFDC上訓練的深度偽造檢測模型可以泛化到真實的深偽影像，這樣的模型在分析潛在深度偽造影像時可以是一個很有價值的分析工具。

在我們之前，有許多人同樣進行過有關DFDC數據集影像辨識模型的研究。Mittal等人 [14] 於2020年提出了以影像中的音頻和視覺上的情感作為判斷真偽依據的模型，其模型對於DFDC數據集的AUC值為0.844。Montserrat 等人 [17] 同樣於2020年提出基於CNN和RNNs的模型，其模型準確率為0.9188。Thing 等人 [16] 於2023年提出了基於CNN以及Transformer的模型，其準確率為0.9202、AUC值為0.9761。Ishrak等人 [15] 於2024年提出使用CNN和帶有LSTM的CapsuleNet的模型，其準確率為0.88，AUC值為0.9510。

貳、 研究設備與器材

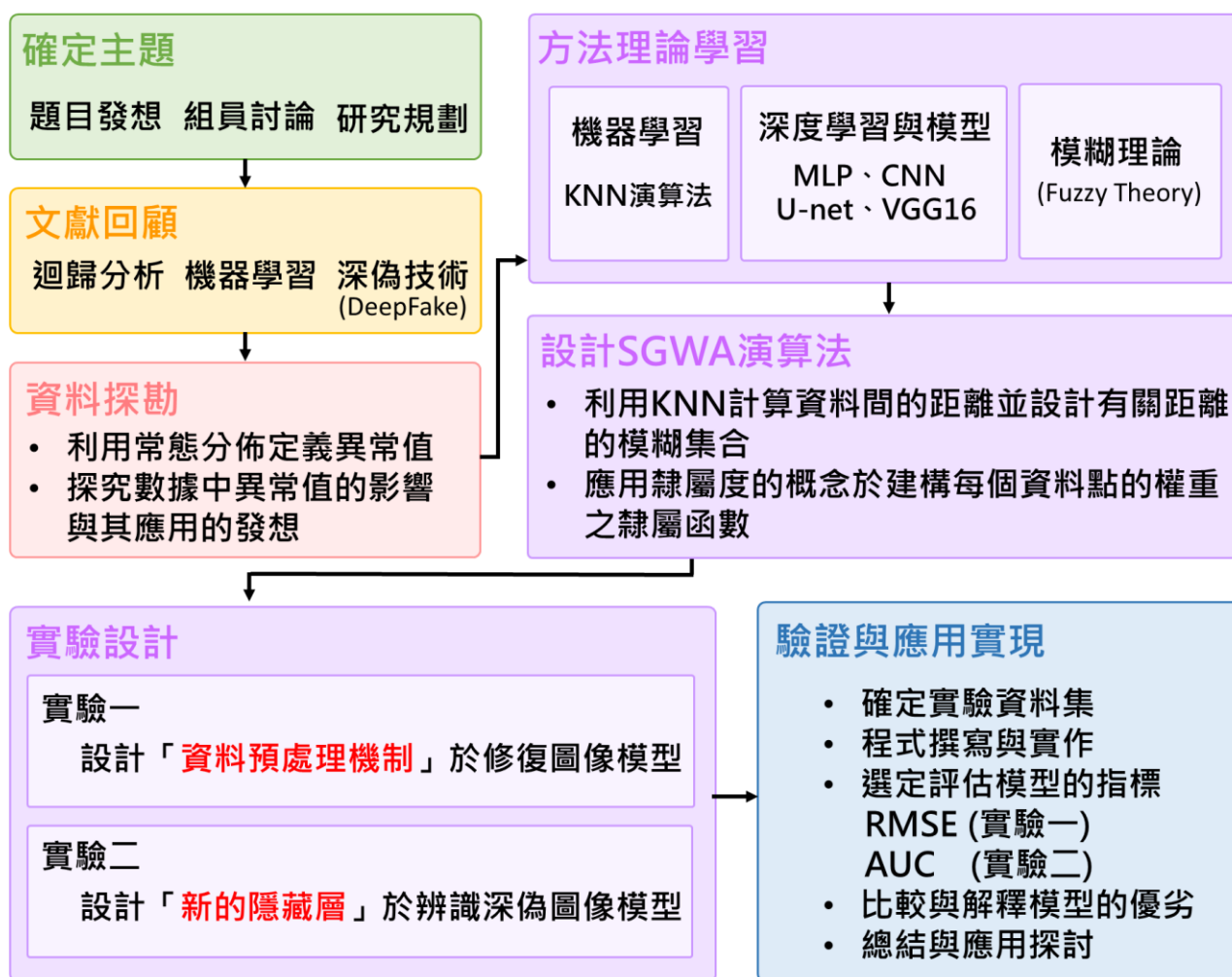
一、硬體部分： 筆記型電腦、桌上型電腦。配備：Intel Core i5。

二、軟體部分：

- (一) Google Colab：是由 Google 免費提供的、基於 Jupyter Notebook 技術的線上開源交互性編成環境，支援眾多流行的 Python 庫和框架，方便地進行機器學習的開發。
- (二) Python 3.10.12 函數、計算與繪圖等套件庫：Numpy、Pandas、Matplotlib、Sklearn、SciPy 與其他標準函式庫。
- (三) Kaggle 數據資料：Kaggle 是全球最大的數據科學和機器學習交流網站平台之一，該網站提供了豐富的數據集，其涵蓋各種領域，並提供用戶進行數據分析、模型訓練和預測用。

參、研究過程或方法

一、研究流程

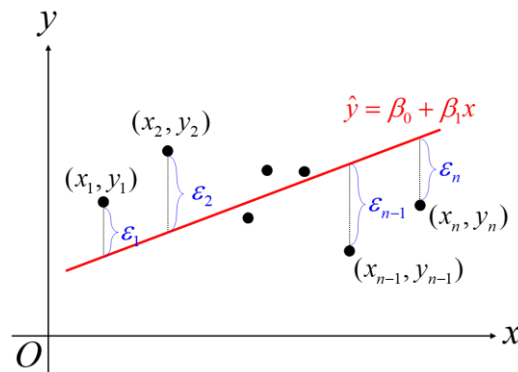


圖一：研究架構圖（圖片來源：作者自行繪製）

一、探討異常值對數據的影響 – 以「利用最小平方法於二維數據之迴歸分析」為例

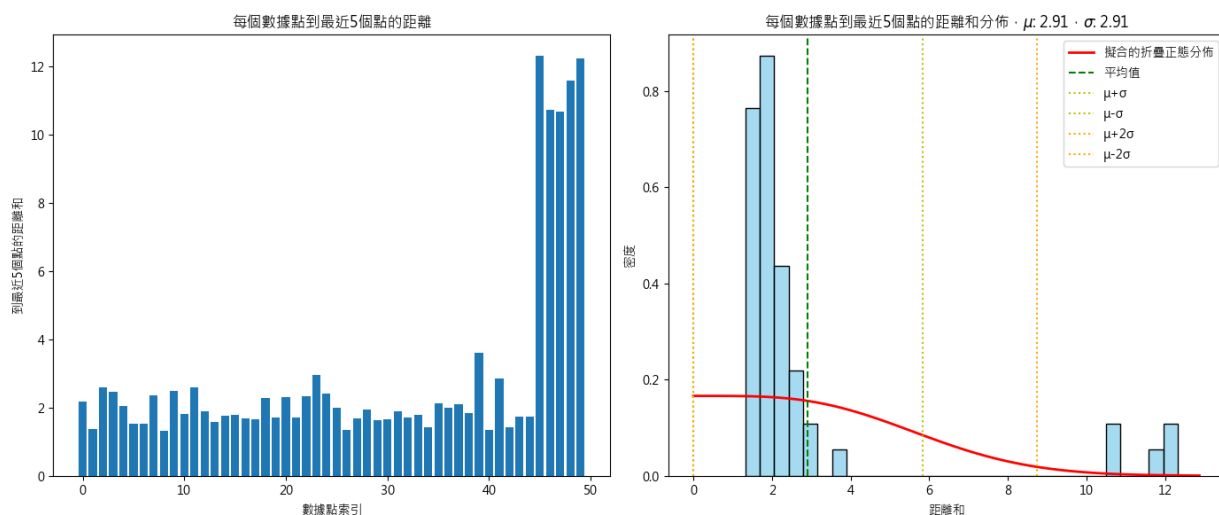
此方法是統計學和數學中一種廣泛應用的迴歸分析方法，特別是在線性迴歸模型中。這種方法的目的是找到一條最佳擬合線，使模型預測值（Predicted Values）和實際觀測值（Observed Values）之間的平方誤差的總和能最小化，進而找到最佳擬合模型的參數。

假設在 n 組二維的數據 (x_i, y_i) ， $i=1,2,\dots,n$ 中，我們希望找到一條最佳擬合的直線 $y = \beta_0 + \beta_1 x$ ，其中 β_0 為直線的截距， β_1 為直線的斜率，如圖二所示。



圖二：數據資料與迴歸直線（圖片來源：作者自行繪製）

先將這 n 組數據代入此直線，可得模型預測值 $y_i = \beta_0 + \beta_1 x_i$ ， $i=1,2,\dots,n$ ，接著考慮實際觀測值 y_i 與模型預測值 y_i 之間的殘差，記為 $\varepsilon_i = y_i - y_i$ ， $i=1,2,\dots,n$ ，我們的目標為透過解 $\sum_{i=1}^n (\varepsilon_i)^2$ 之最小值時，進一步來估計未知模型之最佳的參數 β_0 與 β_1 。然而最小平方法的一個重大缺點是對異常值非常敏感，即在數據中若存在極端值時，模型容易受到異常值的影響，進而影響參數的估計值，為了驗證其異常值對於模型的影響，我們開始撰寫程式來呈現其影響，首先隨機生成符合常態分佈的 45 組二維數據資料與 5 組異常值，其中資料的平均值為 μ ，標準差為 σ ，根據常態分佈的定義，我們可以根據數據分布畫出其機率分布圖，並可以將大於 $\mu + 2\sigma$ 之數據定義為異常值，如圖三所示，左圖是一組關於路徑長的數據，右圖是這組數據的機率分布圖，可以發現機率分布是符合半正態分布的，這是因為路徑長的計算方式並不考慮正負號，所以該數據僅須符合半正態分布。我們從右圖的機率分布情顯示，有覺突兀的分布落在最右端，即對應左圖中，因異常的數據造成異常較大的路徑長，因此我們便可利用此方式來定義資料的異常值。



圖三：利用常態分佈來定義異常值（圖片來源：作者自行繪製）

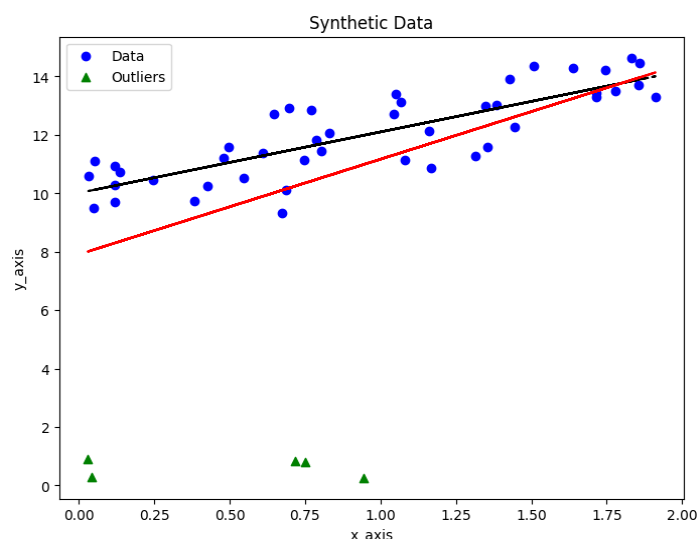
再利用最小平方方法分別計算無包含異常值的數據與有包含異常的數據之迴歸直線，程式碼如圖四所示。

```
#Without outliers
ls_result = optimization.leastsq(func, thelda, args=(X,y))
intercept = ls_result[0][0]
slope = ls_result[0][1]
print(f"[Without outliers] intercept: {intercept:.3f}, slope: {slope:.3f}")
plt.plot(x, slope*x + intercept, 'k--', label = 'Without outliers')

#With outliers
ls_result_ = optimization.leastsq(func, thelda_, args=(X_,y_))
intercept_ = ls_result_[0][0]
slope_ = ls_result_[0][1]
print(f"[With outliers] intercept: {intercept_:.3f}, slope: {slope_:.3f}")
plt.plot(x_, slope_*x_ + intercept_, 'red', label = 'With outliers')
```

圖四：利用統計套件的最小平方方法函數分別計算有、無包含異常值的迴歸直線之程式碼（圖片來源：作者自行繪製）

並在二維平面上分別繪製兩條直線，其中黑色的直線是透過原 45 組的數據資料所生成的迴歸直線，另外一條紅色的直線為考慮 45 組的數據與 5 組離群的數據所生成的迴歸直線，可以清楚地觀察到紅色的迴歸直線受到 5 個極端值的影響，出現被往下偏移的行為，即解出表現較不佳的直線截距 β_0 與斜率 β_1 ，如圖五所示。

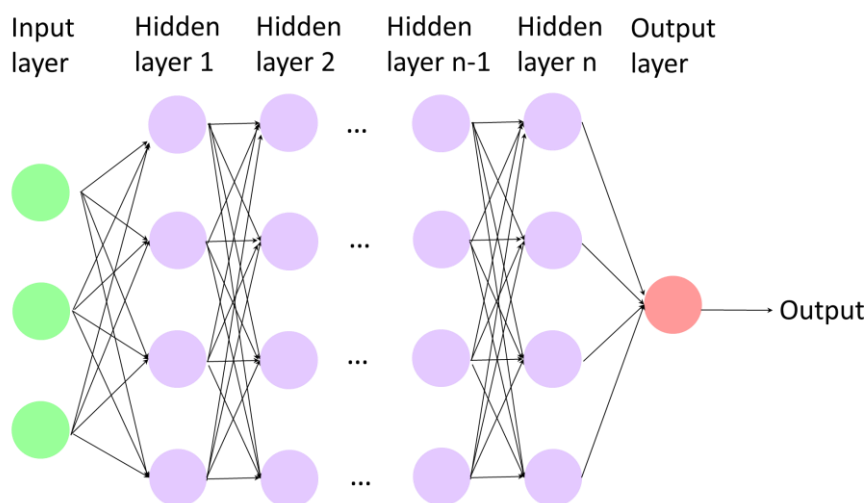


圖五：分別繪製有、無包含異常值之兩條迴歸直線（圖片來源：作者自行繪製）

因此，在實際應用中，我們需要特別留意異常值對模型的潛在影響，便能善用其影響的範圍來作後續的圖像分析。

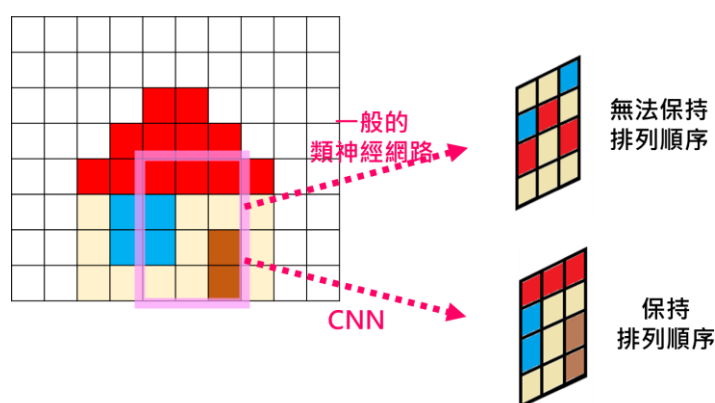
二、探究深度學習與模型

第一個為多層感知器，英文學名為 **Multilayer Perceptron**，簡稱 **MLP**，是一種前向傳遞類神經網路，由輸入層、隱藏層和輸出層組成，常見的結構如圖六所示。**MLP** 的每一層都是完全連接的，即每個神經元都與前一層的所有神經元相連。在圖像修復任務中，**MLP** 主要用於處理低維度的圖像數據。雖然 **MLP** 在處理簡單的圖像任務時表現良好，但由於其無法捕捉到圖像中的空間訊息，因此在處理複雜圖像修復任務時效果很有限。



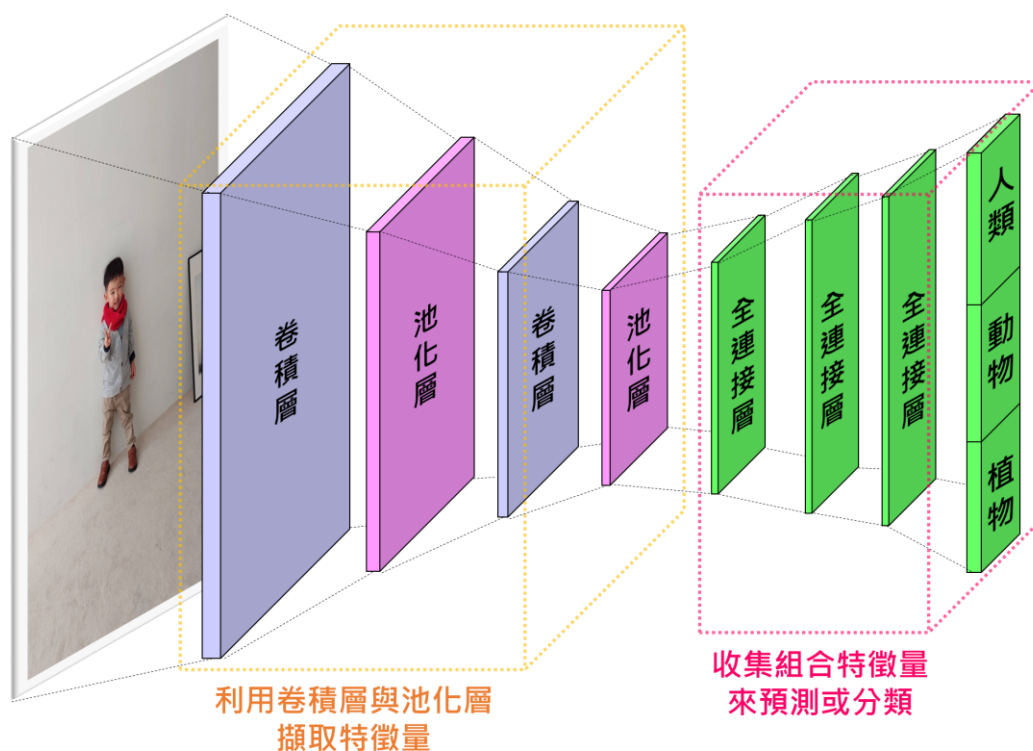
圖六：**MLP** 的常見架構（圖片來源：作者自行繪製結構）

第二個為卷積類神經網路，英文學名為 Convolutional Neural Network，簡稱 CNN，此演算經常用於圖像辨識處理的領域，我們常見的彩色圖像資料會是以三原色（RGB）分別表示灰度值的三維陣列來呈現。一般的類神經網路無法保持排列順序，而 CNN 卻能夠保持多維陣列的像素位置關係來處理資料，也就是指輸入層能夠接收保持位置關係的資料，如圖七所示，後面階層的處理可運用該位置關係的資訊。



圖七：CNN 與一般類神經網路的排列順序與否之差異示意圖（圖片來源：作者自行繪製）

CNN 主要是由卷積層（Convolution Layer）、池化層（Pooling Layer）和全連接層（Full Connected Layer）所構成的，如下圖八。

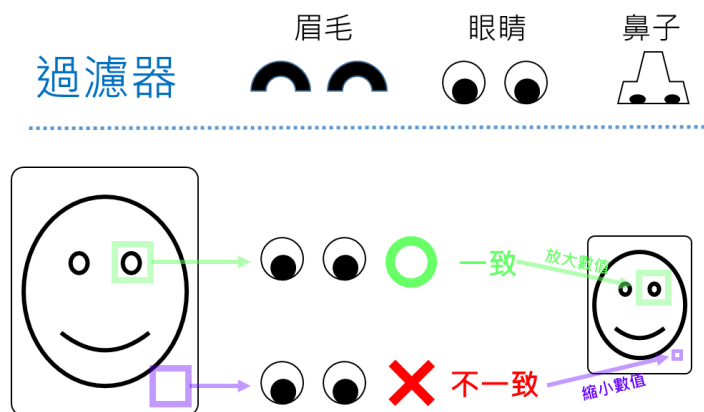


圖八：CNN 的架構（圖片來源：指導老師已授權提供小孩的照片與作者自行繪製結構）

簡單來說，就是卷積層和池化層交替排列，再加上幾個全連接層。前半部分的卷積層和池化層會反覆地擷取圖像的特徵。雖然每一層只能擷取一些簡單的特徵，但經過多層處理後，就能夠提取出更複雜的特徵。而後半部分的全連接層會把這些複雜的特徵轉化成數字，然後根據這些數字來進行預測和分類，接下來我們將每一層的功能整理如下：

（一）卷積層

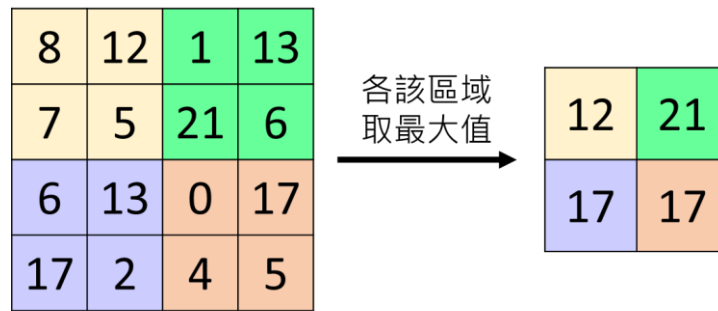
每張圖像中都會有特定的形狀，因此當經過此層中，會針對有反應的卷積過濾器來處理圖像。這些過濾器會透過學習，而變成能夠辨認特定標籤的形狀。例如，學習辨認人像的 CNN，會變成能針對人的眉毛、眼睛、鼻子有反應的過濾器，如圖九所示。當輸入新的圖像時，卷積過濾器會逐一掃描圖像的每個像素，把過濾結果生成新的圖像。在這個過程中，圖像中與神經網路學到的特徵一致的部分會被強調出來。這些強調特徵的圖像被稱為特徵圖。而且每個卷積層都會生成一個特徵圖，所以原來的一張圖像會變成多張特徵圖。



圖九：卷積層的運作流程之示意圖（圖片來源：作者自行繪製）

（二）池化層

在此層中，常會用一個固定大小的區域掃描整個圖像，然後從該區域中提取一個數值來生成新的圖像，事實上有很多方法可以提取數值，其中在 CNN 模型最常用的為最大池化法 (Max Pooling)，最大池化法就是從每個區域中選出最大的數值。如圖十所示，當進行最大池化後，原來 4x4 的數據會縮小成 2x2 的數據，每個數值都是各個該區域中的最大值，所以當數據的縮小時，可以大幅減少資料的計算。



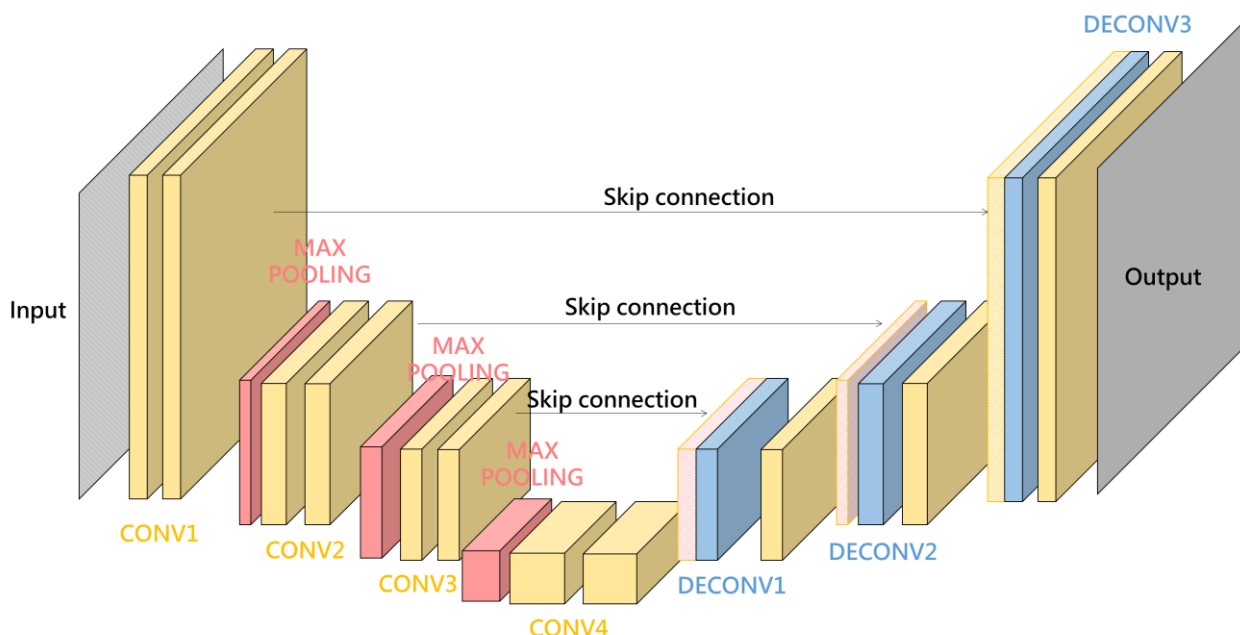
圖十：最大池化法為從每個區域中選出最大的數值（圖片來源：作者自行繪製）

（三）全連結層

這一層的結構其實跟一般的神經網路差不多，它讀取前面兩層處理過的特徵圖，擷取其中的特徵量，最後在輸出層給出分類和預測的結果。這部分也跟卷積處理的方式一樣，堆疊多層可以處理更複雜且有用的特徵量。

因此學會以上每一層的基本功能後，然而使用這種 CNN 模型，我們可以應預其的預測的能力來修圖像。

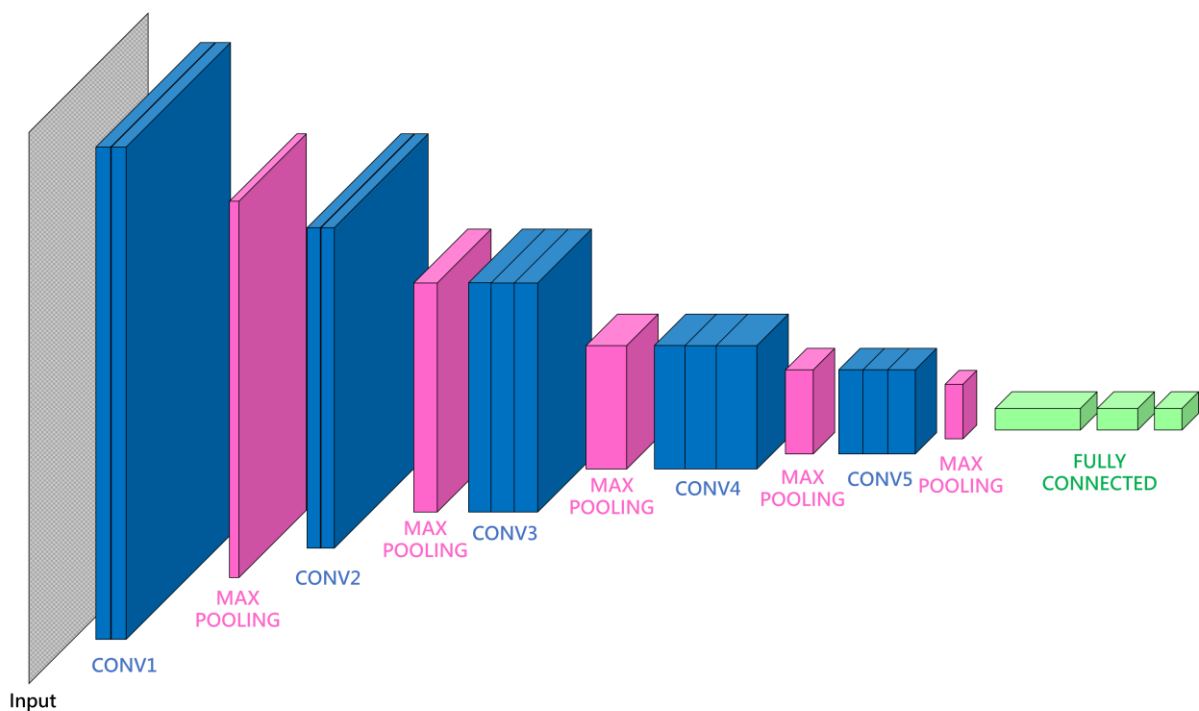
第三個為 U-Net，英文學名為 Convolutional Networks for Biomedical Image Segmentation，此為一種特別的神經網路，常用來進行圖像處理，特別是在醫學圖像分割方面。它的命名來自於它的結構，看起來像一個英文字母 U，如圖十一所示。



圖十一：U-Net 的架構（圖片來源：作者自行繪製結構）

此結構有兩個主要部分，分別為編碼器與解碼器。編碼器就像一個過濾器，逐步縮小圖像尺寸並提取重要特徵；而解碼器則是將這些特徵重新放大，還原成原始圖像的大小。另外最特別的部分是「跳躍連接」，這些連接把編碼器的細節直接傳給解碼器，這樣不會丟失任何重要的細節訊息，這讓 U-Net 能夠更準確地處理圖像，尤其是在醫學影像中能夠更清楚地分辨出不同的組織和器官，因此這是一種能幫助我們更好地理解與處理修復複雜的圖像之工具。

第四個為 VGG16，此模型是來自於英國牛津大學 Visual Geometry Group，簡稱 VGG，所提出，而數字 16 表示結構中的網路有 16 層，包括 13 個卷積層和 3 個全連接層，如圖十二所示。



圖十二：VGG16 的架構（圖片來源：作者自行繪製結構）

這些層就像濾鏡一樣，逐步提取圖像中的特徵，讓電腦能夠更好地辨識與分類圖像。特點是每個卷積層的濾波器都是 3×3 的大小，這樣可以捕捉到圖像中更多的細節。這個網路雖然很深，但結構相對簡單，而且效果非常好，所以常被應用在圖像的處理上。

四、探究 K-近鄰演算（K-NN Algorithm）法並應用其計算結果於異常值的權重之設計

K-近鄰演算法（K-Nearest Neighbors，簡稱 KNN）是一種廣泛應用於分類與迴歸問題之常見的監督式機器學習算法。其基本思想是利用周圍鄰近的資料點的信息來預測新數據點的

標籤或值。在 KNN 中，**K** 表示鄰近的鄰居數量。在解決分類的問題上，算法尋找與新數據點最相似的 **K** 個訓練數據點，然後基於這 **K** 個鄰居的標籤進行多數投票決定新數據點的分類。然而應用於迴歸的問題中，則是計算 **K** 個最相似鄰居的平均值作為新數據點的預測值。，此演算法相對好理解與操作，主要包括以下步驟：

- （一）先決定鄰近數量 **K** 值：**K** 值的選擇對 KNN 的性能至關重要。較小的 **K** 值使模型對異常值更敏感，而較大的 **K** 值可能使模型過於平滑。選擇一個合適的 **K** 值通常需要透過交叉驗證（Cross-Validation）等方法進行調參，而此次模擬數據實驗時，我們選擇先設定 **K** 值為 5，在程式碼中，如圖十三所示，其值為 6 的原因是為了要計算一個點到其他鄰近的 5 個資料點的距離，其值包含本身的點到本身的距離為 0，也必須要存取在各點之距離的陣列中。
- （二）計算距離：對於每一個新數據點，計算它與訓練數據集中每個樣本點的距離。距離可以使用歐氏距離、曼哈頓距離或其他度量方式來衡量，這裡是選用我們熟悉的歐氏距離來計算點與點之間的距離。
- （三）鄰近資料點的選擇：選擇距離最近的 **K** 個樣本點，這些樣本點即為新數據點的鄰近鄰居，這裡是利用 sklearn.neighbors 套件下的函數 NearestNeighbors（）來幫我們完成計算，再將其結果存放在變數 result 中，如圖十三所示。

```
k = 6 #include self
result = NearestNeighbors(n_neighbors=k)
result.fit(data_comb)
```

圖十三：決定鄰近數量 **K** 值之程式碼（圖片來源：作者自行繪製）

- （四）進行分類或迴歸：對於分類的問題時，根據 **K** 個鄰居的多數決定新數據點的分類；對於迴歸的問題時，計算 **K** 個鄰居的平均值作為新數據點的預測值。

KNN 演算法的優點之一就是簡單易懂，無需進行模型的訓練，並且適用於多類別問題。不過在這次研究中，我們不會用來解決分類或迴歸等問題，而是要利用此演算法計算出來的

各點到其他點的距離之結果，要將其推廣於設計異常值的權重，我們希望在整個數據資料中，能大幅降低異常值的影響，即會給予其「較低的權重」，簡單來說就是加權平均的概念。

首先我們繼續利用原先隨機生成 45 組數據與 5 組異常值之數據資料來操作 KNN 演算法，為了視覺化每個資料點到其他點的距離，我們利用迴圈隨機選色，讓點與點的距離呈現不同的顏色，程式碼如圖十四所示。

```
#make connect line different colors
number_of_colors = m+m1
color = ["#" + ''.join([random.choice('0123456789ABCDEF') for j in range(6)])
        for i in range(number_of_colors)]
```

圖十四：撰寫迴圈隨機將點與點之間的距離配色之程式碼（圖片來源：作者自行繪製）

接著再設計函數 `connectpoints()`，將 KNN 演算法所算出來的點與點之間的距離之結果代入此函數中，程式碼如圖十五所示。

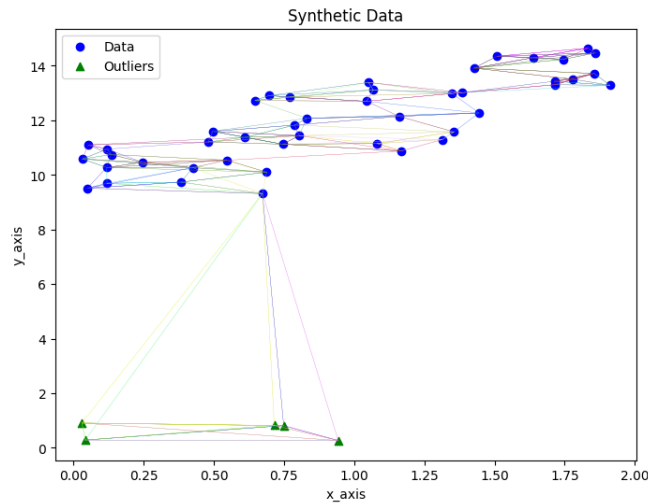
```
def connectpoints(x, y, p1, p2):
    x1, x2 = x[p1], x[p2]
    y1, y2 = y[p1], y[p2]
    plt.plot([x1, x2], [y1, y2], color=color[p1], linewidth=0.2)

i=0
j=1

while i < m+m1:
    while j < k:
        connectpoints(x_, y_, i, result.kneighbors([data_comb[i]])[1][0][j])
        j=j+1
    i=i+1
    j=1
```

圖十五：使用函數 `connectpoints()` 描繪距離之程式碼（圖片來源：作者自行繪製）

最後再將其連線後的結果繪製出來，如圖十六所示。因為是代入 KNN 演算法所算出的結果並存入多維變數 `result` 中，且當初選定的 `K` 值為 5，所以可以清楚地觀察到每個點所連線出去於鄰近的距離都皆有五條線，同時也將 KNN 演算法所算出的每五條線之歐氏距離值存入在變數 `result[0][0]` 中，以便於我們之後計算每個資料點的權重用。



圖十六：利用 KNN 演算法視覺化數據資料點與點的距離（圖片來源：作者自行繪製）

五、探究模糊理論

模糊理論提供一個強大且靈活的數學框架，用於處理現實世界中的不確定性和模糊性。其理論的基本思維是模擬人類認知中的模糊性，將不確定性引入數學模型，得以更好地處理模糊和模糊信息，例如：評價一家餐廳的服務品質，傳統的評價方式可能是使用具體的數字或詞語，例如「服務很好」、「食物很美味」等，但這樣的描述可能仍然無法完全涵蓋人們對整體餐廳體驗的感受，因為這些描述通常是非常主觀的，可能認為服務品質在「好」和「非常好」之間，但不確定具體屬於哪一個，這時就可以應用模糊理論提出的「模糊集合（Fuzzy Set）」與「隸屬函數（Membership Function）」來進行「較彈性的空間」來討論人類的感受。我們研讀與探究了比較重要且實用的觀念，將其探究的內容整理如下：

（一）模糊集合

模糊集合可說是此理論最核心的基本且重要的觀念，其集合允許元素「部分地」屬於集合，而不是侷限於傳統集合（Crisp Set）理論那樣以二進制方式，即所謂的 0 或 1，分別對應表示完全屬於或不屬於兩種，例如：假設有一集合 $A = \{x | x^2 - 3x + 2 = 0\}$ ，則 $1 \in A$ 但 $100 \notin A$ 。然而在模糊集合中，每個元素都有一個屬於該集合的程度，稱為「隸屬度（Membership Degree）」，其值的範圍通常為 $[0, 1]$ ，此範圍的度量允許元素同時屬於多個集合，並且可以用連續的方式表示模糊性，這一概念在描述真實世界中的模糊和不確定性問題時非常有用。我們再次以評價一家餐廳的服務品質為例，使用模糊集合來

描述不同層次的服務品質，例如「非常差」、「差」、「一般」、「好」、「非常好」。每個層次都有一個相應的隸屬度，表示人們對該層次的評價程度，當評價服務品質時，就可以同時給予多個層次的歸屬度。

(二) 隸屬函數

隸屬函數，常以數學符號 α_A 表示，是模糊集中的一個重要元素，它描述了元素對於模糊集合的隸屬程度，其值通常在 $[0, 1]$ 的範圍內，記為 $\alpha_A(x): X \rightarrow [0, 1]$ ，而隸屬函數的常見圖形為三角形或梯形型，這樣的函數能夠更靈活地建模不同元素的隸屬度，以下是我們簡單地設計隸屬函數，用以描述「服務品質」這個模糊集合，整理如下：

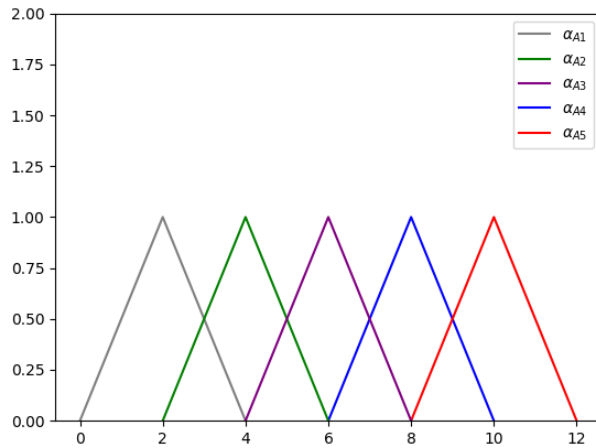
1. 非常差：隸屬函數圖形為三角形型，峰值位置（peak）為 2，左邊界（left base）為 0，右邊界（right base）為 4。
2. 差：隸屬函數圖形為三角形型，峰值位置為 4，左邊界為 2，右邊界為 6。
3. 一般：隸屬函數圖形為三角形型，峰值位置為 6，左邊界為 4，右邊界為 8。
4. 好：隸屬函數圖形為三角形型，峰值位置為 8，左邊界為 6，右邊界為 10。
5. 非常好：隸屬函數圖形為三角形型，峰值位置為 10，左邊界為 8，右邊界為 12。

我們統一把以上五個對應的模糊集合 A_1, A_2, \dots, A_5 與隸屬函數 $\alpha_{A_1}, \alpha_{A_2}, \dots, \alpha_{A_5}$ ，以數學的形式整理如下：

$$\alpha_{A_1}(x) = \begin{cases} \frac{1}{2}x, & 0 \leq x < 2 \\ 1, & x = 2 \\ -\frac{1}{2}(x-4), & 2 < x \leq 4 \end{cases}, \alpha_{A_2}(x) = \begin{cases} \frac{1}{2}(x-2), & 2 \leq x < 4 \\ 1, & x = 4 \\ -\frac{1}{2}(x-6), & 4 < x \leq 6 \end{cases}, \alpha_{A_3}(x) = \begin{cases} \frac{1}{2}(x-4), & 4 \leq x < 6 \\ 1, & x = 6 \\ -\frac{1}{2}(x-8), & 6 < x \leq 8 \end{cases}$$

$$\alpha_{A_4}(x) = \begin{cases} \frac{1}{2}(x-6), & 6 \leq x < 8 \\ 1, & x = 8 \\ -\frac{1}{2}(x-10), & 8 < x \leq 10 \end{cases}, \alpha_{A_5}(x) = \begin{cases} \frac{1}{2}(x-8), & 8 \leq x < 10 \\ 1, & x = 10 \\ -\frac{1}{2}(x-12), & 10 < x \leq 12 \end{cases}$$

並繪製在圖一個二維平面上，可以觀察到，服務品質分為五個層次，但每個層次都可以用隸屬函數來描述其品質的層度，而不再只是 2、4、6、8 與 10 等五個評分值，如圖十七所示。



圖十七：用來描述「服務品質」的五個層次之隸屬函數圖形（圖片來源：作者自行繪製）

六、設計新型演算法

我們將此演算法命名為「自我生成權重演算法（Self-Generated Weighting algorithm）」，簡寫為 **SGWA**，顧名思義就是在給定一組 n 維數據資料中，各點本身可自動計算自己的權重，我們都知道異常值對於評估線性迴歸模型會造成不良的影響，因此我們盡可能要減低或甚至消弭其對於評估過程的影響，最核心的思維就是針對每個資料點，給予一個參與計算的程度，即為權重值，因為異常值為一群少數與其他數據點相比時具有顯著不同或偏離常態分佈的觀測值，所以要給予它們較低的權重，讓一般正常且常態分佈的資料得以凸顯其重要性，進一步地再計算迴歸模型時能大幅地提高其預測的能力，因此如何針對每個資料點，給定較高或較低的權重值，是這個演算法最重要的關鍵，我們要透過模糊理論所教的隸屬函數來實現。

為了簡化異常值在數據預測中的影響，我們選擇將距離作為判定異常值的標準之一。我們發現，異常值往往與真實數據之間有相較真實數據和真實數據間大的距離，這個距離大多是數據觀測值之間的差，可以是一維數據之間的差，或多維數據的分量差平方和（歐幾里得距離），考慮數據間的距離，可以幫助我們更好的辨識及弱化異常值。

已知 **KNN** 演算法是一個根據數據間距離進行學習的非監督式學習演算法，它會根據數據間的距離進行分類，這對於我們分類真實數據以及異常值可以得到極大的幫助。我們在自創演算法中使用 **KNN** 演算法進行距離相關的計算，這樣可以讓我們的計算結果基於數據之間的相似性，而使數據能不脫離其原始的樣貌而改善預測結果。

我們可以改變 **K** 值以選擇我們要找 **K** 個最相近的鄰點數據，並計算該像素點與此 **K** 個

鄰點數據的差將其相加。在圖像處理中，我們可以列出像素點周圍的數據，並計算與其最近的 K 個像素點的距離並相加，當我們遍歷完所有像素點，我們便完成該圖片中像素點與像素點之間的絕對差距集合，首先 K 值設定為 5，設 d_i ， $i=1,2,\dots,n$ 為任選一點到其他鄰近五個資料點之歐氏距離總和並建立一個有關距離總和之集合 $S = \{d_1, d_2, \dots, d_n\}$ 與一個模糊集合 $F = \{(d, \alpha) | d \in S, \alpha \in [0,1]\}$ ，在撰寫程式碼中，我們用陣列變數 `data_dist_sum` 並利用兩層的迴圈與 KNN 演算法所算出的結果來存取所有 d_i 值，如圖十八所示。

```
#knn_calculate a point to other 5 points distance sum
i=0
j=1
data_dist_sum = np.zeros(m+m1)

while i < m+m1:
    while j <= k-1:
        data_dist_sum[i] = data_dist_sum[i] + result.kneighbors([data_comb[i]])[0][0][j]
        j=j+1
    i=i+1
    j=1
```

圖十八：建立有關距離總和的模糊集合之程式碼（圖片來源：作者自行繪製）

我們會發現，計算出的像素點差距是絕對的，無法直接比較像素點與像素點之間的重要性，我們必須使用前面所提到的模糊理論中的模糊邏輯以及隸屬函數將絕對差距轉成相對差距才可以做比較。我們必須先定義怎樣的距離算是必須處理的異常值，首先算出距離集合中的平均數 μ 及標準差 σ ，假設我們的像素點之間的差距（以下簡稱差距）符合常態分布，則根據常態分佈的經驗法則，我們可以將差距大於 $\mu + 2\sigma$ 的值作為異常值，將差距小於 $\mu + \sigma$ 的值當作不需調整權重之數據，則我們可以根據以上條件並針對剛建立的模糊集合 F ，可以設計一個對應集合的隸屬函數 $\alpha: S \rightarrow [0,1]$ ，定義如下：

$$\alpha(d) := \begin{cases} 1 & , d \leq \mu_s + \sigma_s \\ \frac{\mu_s + 2\sigma_s - d}{\sigma_s} & , \mu_s + \sigma_s < d \leq \mu_s + 2\sigma_s \\ 0 & , \mu_s + 2\sigma_s < d \end{cases}$$

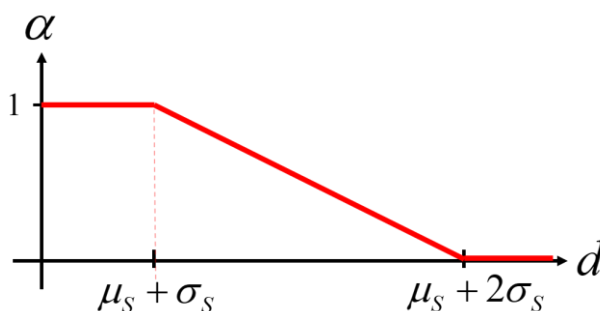
其中 μ_s 為所有 $d_i \in S$ ， $i=1,2,\dots,n$ ，的算術平均數， σ_s 為所有 $d_i \in S$ ， $i=1,2,\dots,n$ ，的母體標準差，在程式碼中，如圖十九所示，我們先使用 `statistics` 套件，協助計算 d_i 的算術平均數與

母體標準差後，再利用條件判斷，當 d_i 值小於 $\mu_s + \sigma_s$ 時，我們給定權重值為 1，當 d_i 值介於 $\mu_s + \sigma_s$ 與 $\mu_s + 2\sigma_s$ 時，給定權重值為 $\frac{\mu_s + 2\sigma_s - d}{\sigma_s}$ ，當 d_i 值大於 $\mu_s + 2\sigma_s$ 時，給定權重值為 0，最後用陣列變數 **alpha** 來存取所有 d_i 值對應的權重值。

```
while t < m+1:
    if data_dist_sum[t] <= mu+std:
        alpha[t] = 1
    elif mu+std < data_dist_sum[t] <= mu+2*std:
        alpha[t] = ((mu+2std) - data_dist_sum[t])/std
    else:
        alpha[t] = 0
    t=t+1
```

圖十九：定義距離的隸屬函數之程式碼（圖片來源：作者自行繪製）

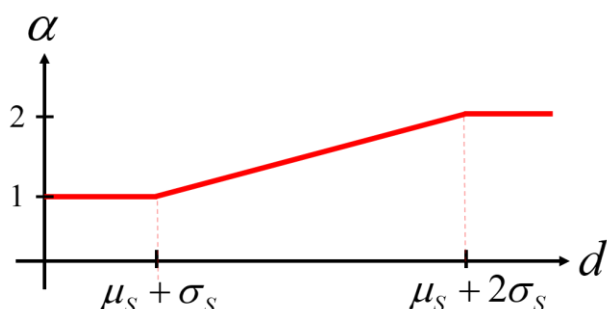
我們透過以上的隸屬函數之設計為的是可以確保異常值的資料所計算出來的 d_i 值必定大於 $\mu_s + 2\sigma_s$ ，因此其對應的權重值設定為 0，即可以直接消弭對於評估模型的計算，但如果是符合常態分佈的資料點，則會落在 $\mu_s + \sigma_s$ 與 $\mu_s + 2\sigma_s$ 之間或小於 $\mu_s + \sigma_s$ ，我們會希望保留其計算的重要性，給予非零的權重值，即可完成每個資料點都可以達到自我生成權重的能力，為了方便觀察隸屬函數圖形之構造的正确性，我們將函數圖形繪製出來，其圖形屬於梯形型，如圖二十所示。



圖二十：弱化異常值的影響時，距離對應的隸屬函數圖（圖片來源：作者自行繪製）

因此現在有了隸屬函數所生成的權重 α 後，我們可以來建構新的預測模型我們透過這樣的方式為差距大的異常值賦予較低的權重，而在模型中弱化了異常值的影響，反之如果要強

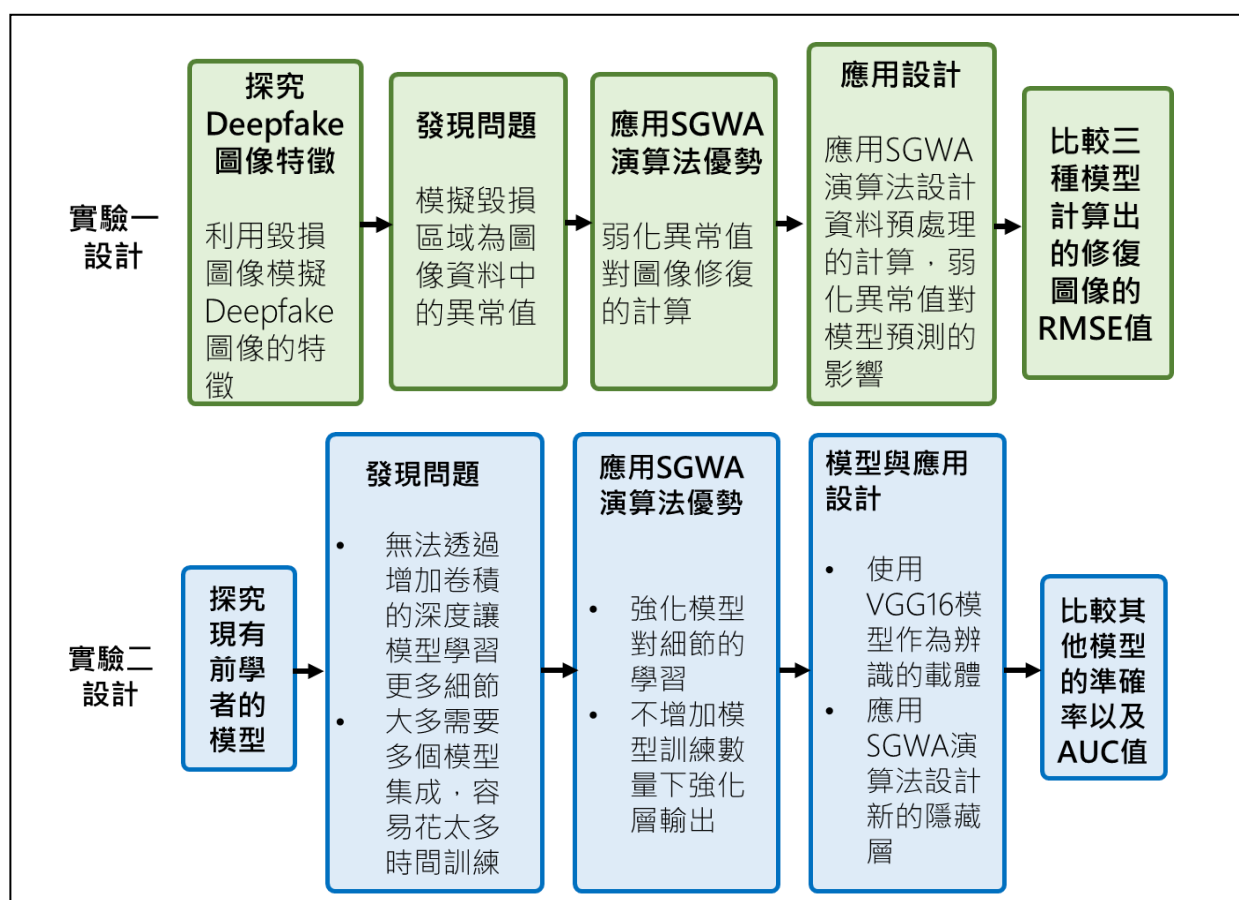
化異常值的影響，我們可以將權重函數反過來計算，能得到強化異常值的結果，對應的隸屬函數圖形，如圖二十一所示。



圖二十一：強化異常值的影響時，距離對應的隸屬函數圖（圖片來源：作者自行繪製）

肆、研究結果

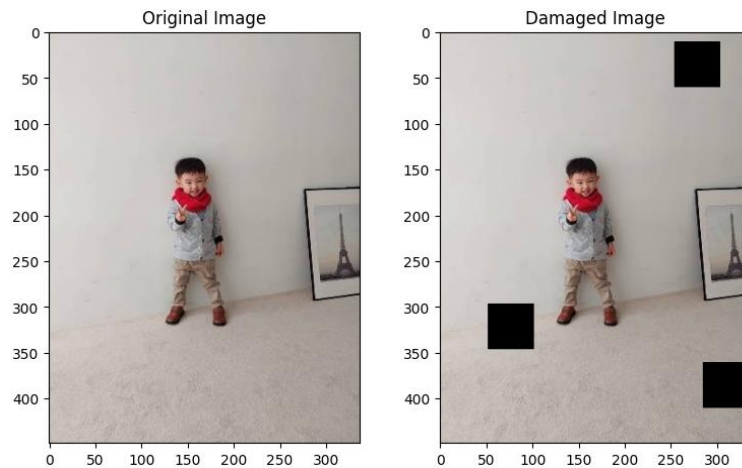
設計完 SGWA 演算法後，我們設計了兩個實驗來驗證我們的方法。首先透過 SGWA 演算法進行圖像的數據預處理，進而達到修復圖像，再進一步使用改良 VGG16 辨識深偽圖像模型的能力，先以圖二十二來呈現我們的實驗一與二的設計流程如下，再分別論述兩個實驗。



圖二十二：實驗一與二的設計流程圖（圖片來源：作者自行繪製）

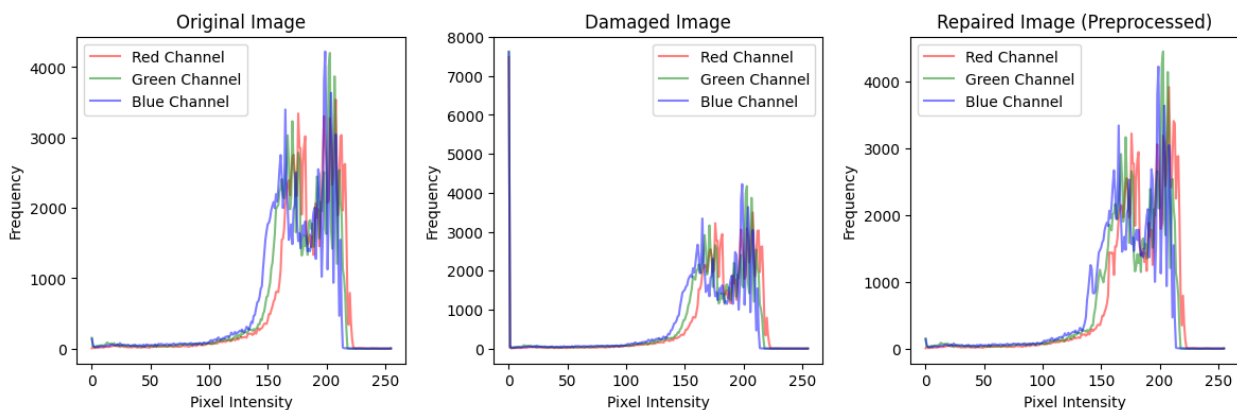
（一）實驗一：圖像修復

在生活中，我們常常會遇到被污染或毀損的圖片，當照片中擁有足以影響判讀的污染或毀損時，對於圖片中所擁有的資訊會產生一定干擾，我們先以正常的圖呈現，如圖二十三之左圖，接著為了模擬被毀損的動作，我們刻意加入一些異常區塊，如圖二十三之右圖所示。



圖二十三：左圖為正常的人像圖片，右圖為被模擬污染的人像圖片（圖片來源：指導老師已授權提供照片，作者自行繪製污染區塊）













這些與圖片資訊不相關的區塊定義為異常值，並應用 **SGWA** 演算法進行圖像中數據的預處理之迴歸分析於 **MLP**、**CNN** 與 **U-net** 等三個深度模型中，從而修正其模型預測，以弱化異常值後重新繪圖達到將圖片資訊完整的目的，首先我們從圖像中提取了 **RGB** 值和每個像素的位置。這些數據是模型訓練的基礎，如圖二十四所示，我們透過顯示原始圖像、損壞圖像以及修復圖像的 **RGB** 曲線圖來進行視覺化比較，可以清楚看到修復圖像的 **RGB** 曲線圖與原始圖像的非常相似，表示有達到修復的效果。



圖二十四：原始圖像、損壞圖像以及修復圖像的 **RGB** 曲線圖（圖片來源：作者自行繪製）

最後我們以均方根誤差（RMSE）來衡量原始圖片與修復圖片之間的差異程度，此值越小表示兩張圖片越相似，換句話說就是修復能力效果佳，並針對 MLP、CNN 與 U-net 等三個模型下，以「用 SGWA 進行數據的預處理（Preprocessed）」與「未預處理（Unpreprocessed）」的 RMSE 值做比較，實驗操作共計四次，每次進行 10 次的修復試驗，並計算 RMSE 值的平均，整理如下表一。

表一：實驗一操作四次，每次進行 10 次後取平均的結果

毀損圖像與 Damaged RMSE	各模型的修復結果與 Repaired RMSE Average Value		
	MLP Model	CNN Model	U-net Model
 <p>Damaged RMSE: 2.379</p>	 <p>Preprocessed RMSE: 1.260 Unpreprocessed RMSE: 1.817</p>	 <p>Preprocessed RMSE: 1.447 Unpreprocessed RMSE: 1.876</p>	 <p>Preprocessed RMSE: 1.094 Unpreprocessed RMSE: 1.276</p>
 <p>Damaged RMSE: 2.296</p>	 <p>Preprocessed RMSE: 1.543 Unpreprocessed RMSE: 1.889</p>	 <p>Preprocessed RMSE: 1.401 Unpreprocessed RMSE: 1.802</p>	 <p>Preprocessed RMSE: 1.372 Unpreprocessed RMSE: 1.627</p>
 <p>Damaged RMSE: 2.129</p>	 <p>Preprocessed RMSE: 1.371 Unpreprocessed RMSE: 1.464</p>	 <p>Preprocessed RMSE: 1.517 Unpreprocessed RMSE: 1.769</p>	 <p>Preprocessed RMSE: 1.278 Unpreprocessed RMSE: 1.433</p>
	MLP Model	CNN Model	U-net Model

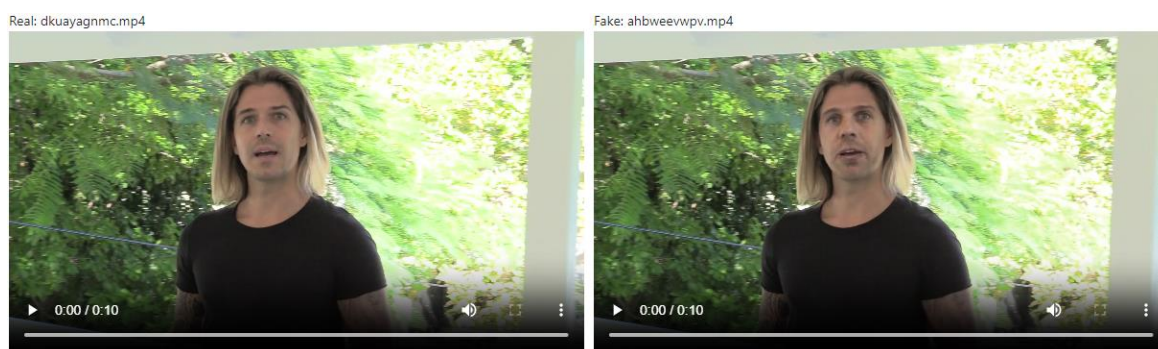


在實驗一中，我們隨機地模擬毀損圖像四次，過程中計算了原始圖像、損壞圖像以及修復圖像的 RMSE 值並繪製出其對應的 RGB 曲線圖。從表一結果顯示，在修復圖像的 RMSE 值的表現中，經過預處理的模型在修復效果上略優於無預處理的模型，其值皆偏小，這表示了我們應用 SGWA 於預處理的過程在一定程度上提高了模型的修復能力。

（二）實驗二：辨識深偽圖像

此實驗以 VGG16 模型實作加入了 SGWA 演算法之隱藏層進行訓練，並與其他四種模型進行比較，藉此可觀察各模型對深偽圖像的辨識能力。

首先我們使用 DFDC 數據集 [18] 進行模型的訓練，從數據集中隨機取一段原始影像片段，如圖二十五之左圖所示，與一段被深偽技術處理過的影像片段，如圖二十五之右圖所示，而只要經過深偽技術處理過的影像片段，必定在「特定偽造的區域」會存在資料的異常值。



圖二十五：左圖為原始影像片段、右圖為深偽影像片段（圖片來源：作者取自[18]）

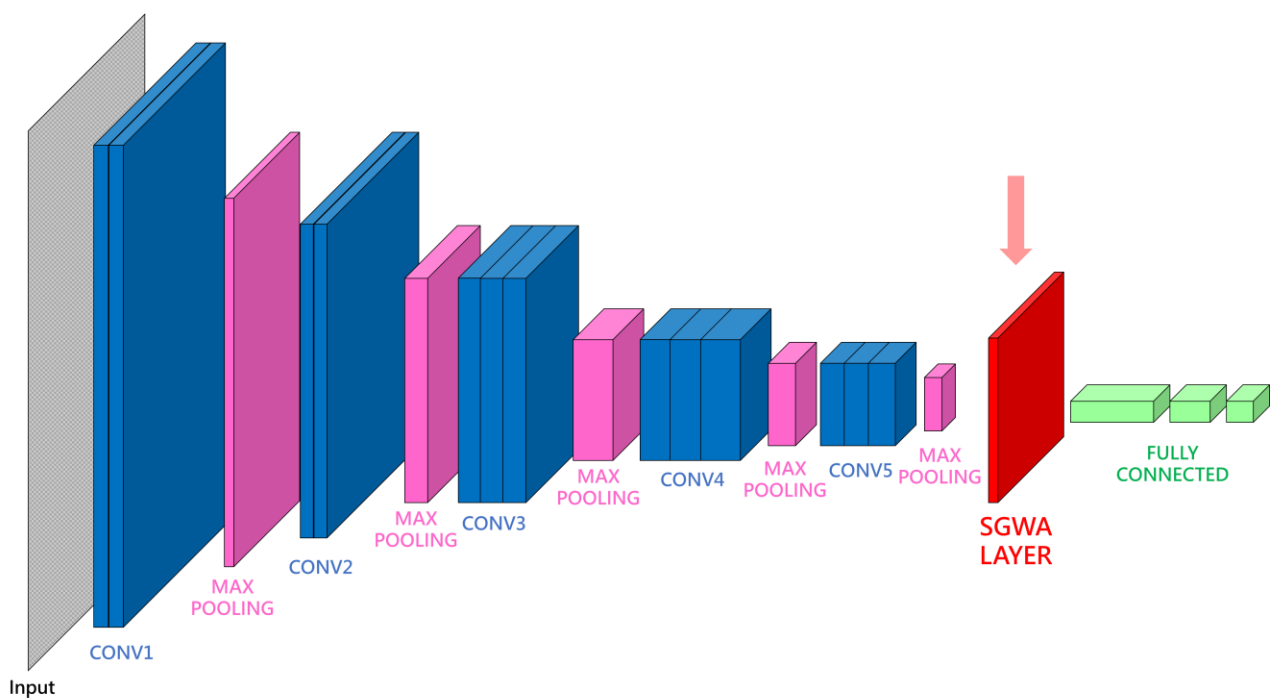
接著透過 MTCNN（Multi-task Cascaded Convolutional Networks）技術將人臉的位置以及關鍵點進行定位，如圖二十六所示，並將該數據保存進 json 檔案中，這樣的好處是可以重複

使用以及方便進行模型檢測，可以更精準的識別是否為深偽影片。



圖二十六：使用 MTCNN 技術框住人臉之示意圖（圖片來源：作者自行繪製）

我們應用 SGWA 演算法並配合圖二十一之隸屬函數，設計一個新的隱藏層，命名為「SGWA LAYER」，並在 VGG16 模型結構中的卷積層到線性分類層之間，插入 SGWA LAYER，如圖二十七所示，此層將會強化卷積層算出來的結果，即針對來自「特定偽造的區域」的異常值，賦予較高的權重，進而強化異常值的影響，這樣就可以把「偽造的細節」凸顯出來，便可更有效且準確地辨識深偽圖像。




圖二十七：加入 SGWA LAYER 於 VGG16 模型中之示意圖（圖片來源：作者自行繪製）

最後我們從 DFDC 數據集中隨機挑選測試 (test data) 影像片段當作輸入來源，將辨識圖像之真偽的實驗結果視覺化整理如下表二，並根據 [18]，使用相同的訓練數據集於五個模型作訓練，針對前學者們研究的四個模型，與我們用 SGWA 演算法改良後的 VGG16 模型之準確率與 AUC 值進行比較並整理成表三 (AUC 值大於等於 0.9 表示模型辨識能力非常好 [13])。

表二：隨機挑選影像片段來源並使用改良後的 VGG16 模型進行辨識真偽的結果

影像片段來源	實際真假值 (Real/Fake)	模型判斷結果
	Fake	Fake
	Real	Real
	Real	Real
	Fake	Fake

			Fake	Fake
--	---	--	------	------

從表二結果顯示，在辨識圖像之真偽的表現中，辨識的結果皆符合原始圖像的實際真假值，這表示了我們應用 SGWA 於改良 VGG16 模型，在一定程度上提高了模型的辨識能力。

表三：各模型辨識的準確率與 AUC 值之比較

模型名稱	準確率	AUC 值
加入 SGWA 的 VGG16	0.9223	0.9805
Mittal 等人(2020) [14]	N/A	0.8440
Montserrat 等人(2020) [17]	0.9188	N/A
Thing 等人(2023) [16]	0.9202	0.9761
Ishrak 等人(2024) [15]	0.8800	0.9510

從表三結果顯示，在辨識圖像之真偽的表現中，無論是準確率亦或是 AUC 值，我們實驗結果之數值皆高於前學者們研究的模型，表示我們改良後的 VGG16 模型有較佳的辨識真偽的能力。

伍、 討論

SGWA 演算法之所以有用，在於他對於數據間距離的掌握可以很好的跟數據的分布進行連結，對於特別突出的數據可以進行弱化或者強化的操作，模擬人類可以辨識與其他模式特別不同的操作，並將其排除或管理的操作。所以 SGWA 可以在實驗一中降低對圖片汙損部分的計算，也可以在實驗二中對 VGG16 所提取之較特殊的特徵進行強化，讓模型可以考慮更細節的部份，使模型可以正確辨識更多影片的真偽。

我們會發現，SGWA 演算法是對距離敏感的，例如距離差距太小可能計算不出來數值。根據我們演算法的設計核心，我們會發現這樣的情況是不怎麼需要用到 SGWA 演算法的，因為那意味著數據的突出部分並不明顯，並不需要對數據進行特別的弱化或強化，且我們所賦

予的權重必定在設定的範圍中，並不會產生權重過大的問題，直接使用便也不會有甚麼問題。

陸、 結論

一、我們成功地結合 KNN 演算法與模糊理論，設計出 SGWA 演算法對我們認定的數據的異常值進行強化或者弱化。

二、在實驗一中，模擬毀損圖像的 RMSE 值約為 2.1 至 2.4，而在修復過後圖像的 RMSE 表現中最佳結果有降低至 0.522。

三、在實驗二中，比起前學者們研究的模型，我們利用 SGWA 演算法改善過後的 VGG16 模型，對於 DFDC 數據集中影像片段之真偽的辨識率最佳，準確率高達為 0.9223，AUC 值高達為 0.9805。

四、未來展望：

（一）利用更多不同的函數設計我們的權重函數來符合模型需要。

（二）延伸推廣與設計其他的演算法至辨識其它多媒體資源的真偽性，如：聲音、影像等，幫助更多使用者能判斷資訊來源的正確性。

柒、 參考文獻資料

- [1] Rousseeuw, P.J., & Leroy, A.M. (1987). *Robust Regression and Outlier Detection*. John Wiley & Sons.
- [2] Huber, P.J. (1964). Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1), 73-101.
- [3] García, S., Luengo, J., & Herrera, F. (2015). Dealing with noisy data. In *Data preprocessing in data mining* (pp. 72). Intelligent Systems Reference Library. Springer.
- [4] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
- [5] 洪錦魁 (2021)。演算法：最強彩色圖鑑 + Python 程式實作 王者歸來。深智數位。
- [6] 王文俊 (2017)。認識 Fuzzy 理論與應用 (第四版)。全華圖書。

- [7] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- [8] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [9] Chesney, B., & Citron, D. (2019). Deepfakes and the new disinformation war: The coming age of post-truth geopolitics. *Foreign Affairs*, 98(1), 147-155.
- [10] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer.
- [11] Yoo, J. H., & Oh, S. Y. (2019). Comparison of goodness-of-fit indices for structural equation models. *Multivariate Behavioral Research*, 54(5), 625-646.
- [12] Chambon, S., & Crouzil, A. (2004). Similarity measures for image registration. *Pattern Recognition Letters*, 24(1-3), 401-414.
- [13] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861-874.
- [14] Mittal, T., Bhattacharya, U., Chandra, R., Bera, A., & Manocha, D. (2020, October). Emotions don't lie: An audio-visual deepfake detection method using affective cues. In *Proceedings of the 28th ACM international conference on multimedia* (pp. 2823-2832).
- [15] Ishrak, G. H., Mahmud, Z., Farabe, M. D., Tinni, T. K., Reza, T., & Parvez, M. Z. (2024). Explainable Deepfake Video Detection using Convolutional Neural Network and CapsuleNet. *arXiv preprint arXiv:2404.12841*.
- [16] Thing, V. L. (2023, July). Deepfake detection with deep learning: Convolutional neural networks versus transformers. In *2023 IEEE International Conference on Cyber Security and Resilience (CSR)* (pp. 246-253). IEEE.
- [17] Montserrat, D. M., Hao, H., Yarlagadda, S. K., Baireddy, S., Shao, R., Horváth, J., ... & Delp, E. J. (2020). Deepfakes detection with automatic face weighting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 668-669).
- [18] Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., & Ferrer, C. C. (2020). The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*.