

中華民國第 63 屆中小學科學展覽會
作品說明書

高中組 電腦與資訊學科

佳作;(鄉土)教材獎

052506

AI 時光機-利用照片轉換技術重溫在地歷史

學校名稱：國立新竹女子高級中學

作者： 高二 張怡庭 高二 林映君	指導老師： 徐逸娟
---------------------------------	------------------

關鍵詞：電腦視覺、深度學習、照片修復

摘要

目前網路上流傳許多使用人工智慧修復照片的網站或應用軟體。然而，由於這些訓練資料多數來自國外，導致修復台灣「本地」照片的效果欠佳。此外，許多老舊照片因氧化、潮濕而泛黃，使得修復程序比起修復純黑白相片更加困難。因此，本研究旨在建立一個專門修復本地照片的機器學習模型，主要分為以下三個部分：首先，使用機器學習模型對老舊照片進行修復，包括著色、去模糊化和降噪；其次，分析使用不同比例之有色調照片(模擬泛黃照片)訓練模型的效果；最後，研究不同的修復順序(著色、去模糊化、降噪)和模型執行次數對照片修復效果的影響，發現「著色、去噪、去模糊化」的順序修復效果最佳。

壹、前言

一、研究動機：

隨著時間的推移，我們的家鄉逐漸改變，許多古老建築已經消失無蹤，現存的建築也經過多次改造，不再保留昔日的風貌。這讓我們無法透過自己的雙眼看到曾經的家鄉，只能透過照片來回憶。然而，在嘗試使用現有的AI修復模型來修復歷史照片後，卻發現這些模型無法很好地呈現台灣的歷史風貌。我們推測是因為這些模型的數據集來自國外，無法完全反映出台灣獨特的歷史背景和文化氛圍。

為了還原家鄉昔日的風華，本研究首先收集了320張照片來訓練，並使用其中280張照片訓練，40張照片測試。另外我們除了訓練在地化模型照片修復，也更進一步研究了如何提高修復效果。在探索修復老照片的過程中，我們發現有些照片由於年代久遠，產生了泛黃或發霉等問題，使得修復效果不如預期。因此，我們開始尋求解決方案，並決定將訓練資料轉換為有色調的圖片，以更好地模擬這些老照片的特殊情況。在這個基礎上，我們進一步修改模型，以提高修復有色調照片的能力。

多數照片修復程式只能處理單一的修復功能，因此使用者必須進行多次操作，不僅耗費時間且會影響修復效果的精確度。為了克服這個問題，我們探討了三種修復功能的使用順序是否會影響修復結果，並研發了一種整合了「著色」、「去噪」及「去模糊化」三種功能的修復程式，不僅簡化了修復流程，也提高了修復效果的品質，讓使用者能夠輕鬆修復照片。

二、研究目的：

- (一) 對蒐集到效果最佳的現有模型在地化訓練。
- (二) 分析有色調照片在訓練集的比例對照片成果的影響。
- (三) 分析模型執行次數對照片成果的影響。
- (四) 分析修復功能執行順序對照片成果的影響。
- (五) 整合多種修復功能以簡化照片修復的過程。

三、文獻回顧

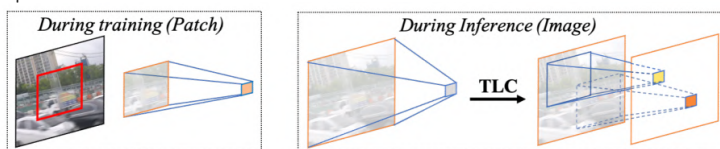
- (一) 著色模型-Deoldify(Salmona, A., Bouza, L., & Delon, J. , 2022)

Deoldify是一種基於深度學習模型自動上色的工具，用於將黑白照片或是黑白電影著色，以下是Deoldify模型中使用到的方法：

1. GAN(Generative Adversarial Network)：GAN模型由兩個神經網路組成，生成器(Generator)和判別器(Discriminator)，這兩個神經網路相互博弈，互相對抗。生成器負責生成新的資料樣本，而判別器負責將這些樣本與原始數據進行區分。在訓練過程中，生成器不斷生成新的樣本，而判別器則不斷試圖區分這些樣本是否為真實數據。透過不斷地訓練，生成器和判別器會逐漸提升自己的能力，最終生成器可以生成與真實數據非常相似的新樣本。
2. NoGAN：此為Deoldify作者開發的新方法，能夠在更短的時間內達到與GAN相當的效果。此方法基於Res-Net使用常規損失函數對生成器進行預訓練，使其更加強大、快速和可靠，同時避免了GAN中可能出現的一些問題，例如模式崩潰、過擬合和不穩定的訓練過程。相比於GAN方法，NoGAN方法可以產生高質量的彩色圖像，並且訓練過程相對較穩定。
3. ResNet(Residual Networks)：一種深度神經網絡模型，它通過引入殘差塊(Residual block)來解決深度神經網絡訓練中的退化問題。在深度神經網絡中，網絡層的增加可能會導致性能下降，這是由於深度神經網絡的非線性層增加，使得在梯度反向傳播過程中梯度消失或爆炸。ResNet通過引入殘差塊，使得神經網絡中的信號可以直接跨層傳遞，從而更好地保持梯度，提高了神經網絡的準確性和可擴展性。ResNet也因此成為了深度學習中非常重要的一個模型。
4. Perceptual Loss：一種損失函數，通常用於生成對抗網絡(GANs)和圖像風格轉換模型中。它是基於感知距離(perceptual distance)的概念，使用預先訓練好的卷積神經網絡(如VGG)計算生成圖像和真實圖像之間的距離，並將其作為訓練過程中的損失函數，以更好地保留圖像細節和結構。通常，使用的是特定層的特徵圖，而不是原始像素值，因為這些特徵圖捕捉了圖像的高層次結構和語義信息。使用perceptual loss可以幫助生成更逼真的圖像，並且通常比傳統的均方差(MSE)損失更有效。

(二) 去模糊化模型-TLC(Chu, X., Chen, L., Chen, C., & Lu, X., 2022)

TLC(Test-time Local Converter)：一種執行圖像恢復任務的深度學習模型，主要任務是將受損的圖像恢復為清晰圖像。由於一般訓練圖像修復模型時，基於gpu的記憶體容量限制，會將單張訓練資料截為較小的圖片，然而使用(測試)時，卻是測試整張圖片，導致測試時的效果不佳，因此TLC便是一個解決此問題之模型(如下圖一)。TLC使的照片在測試階段，也是將照片切為小塊圖片修復，大幅增高正確率。



圖一：TLC執行流程圖

(三) 降噪模型-MPRNet(Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M. H., & Shao, L., 2021)

MPRNet是一種使模糊圖片變得清晰的卷積神經網絡模型，包括初始估計、逐步修復和融合三個步驟。在初始估計步驟中，使用編碼解碼器學習上下文特徵並對圖像進行初步估計。然後，在逐步修復步驟中，使用多個模型對圖像進行精細的修復，利用一個有監督的注意力模塊(SAM)在不同階段進行信息交換。同時，透過雙向方法設置特徵處理塊之間的側連接，以避免任何信息的缺失。最後，在融合步

驟中，使用一個交叉不同階段特徵的融合機制(CSFF)將所有階段的修復結果融合，生成最終的修復圖像。MPRNet通過這個方法，提高圖像的質量，實現圖像清晰、去噪和去模糊等任務，流程如下圖二。

1. 編碼解碼器子網絡

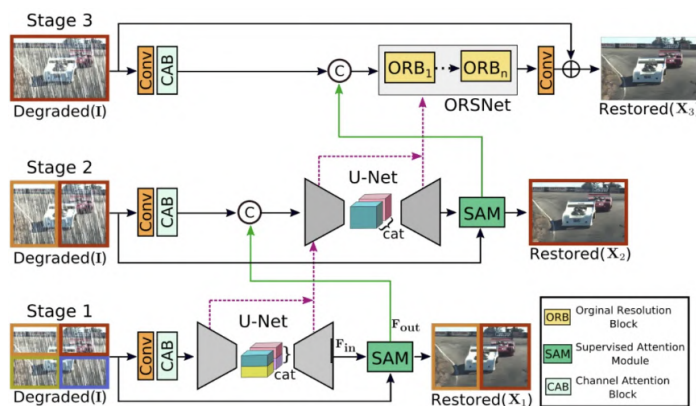
由編碼器解碼器組成。編碼器通過卷積神經網絡將輸入圖像的特徵提取和壓縮，得到一個較低維度的特徵表示。解碼器則通過反卷積神經網絡，將編碼器的特徵表示映射回原始圖像的空間中，完成圖像的重建或修復。

2. 有監督的注意力模塊(SAM, Supervised Attention Module)

生成注意力圖會根據任務的標籤信息調整輸入特徵圖的權重來抑制當前階段信息量較小的特徵，並只把有用的信息傳遞到下一階段。

3. 交叉融合各極端特徵(CSFF, Cross Stage Feature Fusion)

計算每個修復結果與其他修復結果之間的差異，得到一組差異圖。然後利用這些差異圖，採用一個自適應的加權平均方法來融合多個修復結果，從而得到最終的修復結果。



圖二：MPRNet執行流程圖

貳、研究設備及器材

一、硬體設備：

(一)筆電：

- 1、CPU：intel(R)Core(TM)i7-9750H CPU @ 2.60GHz
- 2、記憶體：8.0 GB
- 3、GPU：NVIDIA GeForce RTX 3080

(二)協同視覺實驗室器材

- 1、GPU：GeForce RTX 2080 Ti

二、軟體設備：

(一)軟體環境：

- 1、Python3
- 2、Anaconda3

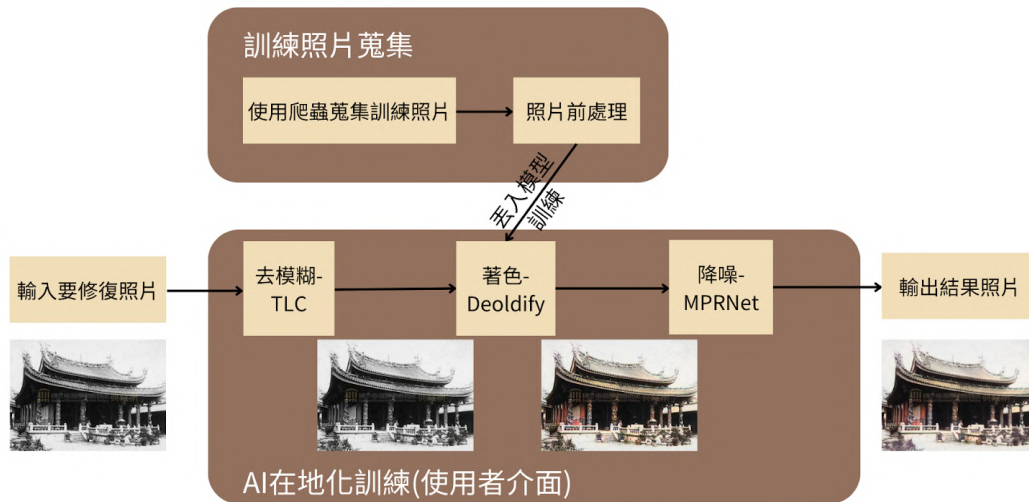
(二)訓練環境：

- 1、PyTorch
- 2、CUDA cuDNN

參、研究過程或方法

一、研究架構

本研究首先對國內外文獻比較分析，研究上色、降噪和去模糊化功能的效能。我們重新訓練了上色功能，對其進行不同的順序排列、改變有套濾鏡照片在訓練集的比例，以了解如何修復功能如何排序才能使修復結果最佳。最後，我們製作了一個互動介面，研究流程如下圖三。



圖三：研究流程圖

二、研究方法

本研究使用公開模型進行訓練，並透過排序、執行次數整合這些修復功能。以下分為三部分：第一部分分別說明著色、去模糊化、降噪模型及重新訓練模型的方式原理；第二部分介紹計算本研究使用的誤差模型的概念；第三部分則說明整合三個功能並利用pygame製作照片修復介面的方法。

(一)修復老舊照片

老舊照片的修復包含了三種方法，分別為著色、去模糊化、降噪。

1.蒐集資料集

本研究從新竹地區的現存歷史古蹟中選取了九個地點，包括孔廟、新竹火車站、東門城、城隍廟、國民戲院、淨業院、第一信用合作大樓、新竹州廳(現新竹市政府)、新竹美術館，並利用自編爬蟲程式透過google伺服器爬取彩色圖片作為訓練資料集，並淘汰不相關的圖片，最後進行模型的訓練

:

```

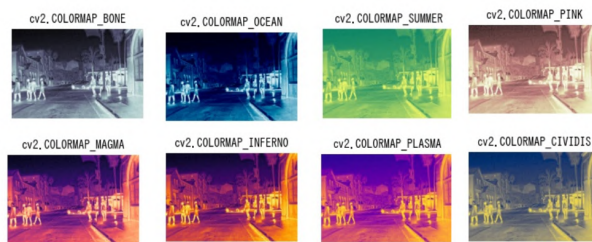
1 import requests
2 import re
3
4 keyword = '新竹車站'
5 save_dir = './images/'
6
7 headers = {
8     'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36'}
9
10 response = requests.get(f'https://www.google.com/search?q={keyword}&tbn=isch', headers=headers)
11
12 image_urls = re.findall('img data-src="(.*?)"', response.text)
13
14 for i, url in enumerate(image_urls):
15     try:
16         response = requests.get(url, headers=headers)
17         with open(f'{save_dir}{keyword}_{i}.jpg', 'wb') as f:
18             f.write(response.content)
19         print(f'Downloaded image {i + 1}/{len(image_urls)}')
20     except:
21         print(f'Error downloading image {i + 1}/{len(image_urls)}')
22

```

圖四：新竹本地照片爬蟲代碼(資料來源：研究者自製)

2. 著色模型

我們基於DeOldify原模型修改資料集的形式，使原本只有黑白照片的訓練資料部分轉為有色調的訓練資料，我們選擇其中部分我們認為最常出現在老照片的數個色調，包括：BONE、OCEAN、SUMMER、PINK、MAGMA、INFERNO、PLASMA、CIVIDIS，如下圖五，利用openCV程式庫將原始黑白照片轉為有色調的照片模擬發霉泛黃(如下圖六)，放入訓練模型中重新訓練，使其能夠更好地處理各種不同的圖像，訓練過程中我們調整有色調照片在訓練集的佔比作為變因，以尋找效果最佳的比例。



圖五：openCV濾鏡風格

```

1 import os
2 import cv2
3 import random
4
5 cmap = [
6     0,
7     cv2.COLORMAP_BONE,
8     cv2.COLORMAP_OCEAN,
9     cv2.COLORMAP_SUMMER,
10    cv2.COLORMAP_PINK,
11    cv2.COLORMAP_MAGMA,
12    cv2.COLORMAP_INFERNO,
13    cv2.COLORMAP_PLASMA,
14    cv2.COLORMAP_CIVIDIS]
15
16 image_dir = r'C:\Users\chang\Downloads\gray - scale'
17
18 image_files = [os.path.join(image_dir, f) for f in os.listdir(image_dir) if os.path.isfile(os.path.join(image_dir, f))]
19
20 for image_file in image_files:
21     gray_image = cv2.imread(image_file, cv2.IMREAD_GRAYSCALE)
22     cmap = random.choice(cmaps)
23     color_image = cv2.applyColorMap(gray_image, cmap)
24     cv2.imwrite(image_file, color_image)

```

圖六：將圖片轉換為有色調之圖片(資料來源：研究者自製)

3.去模糊化模型

此模型使用的是將照片切成小塊圖片然後逐一進行修復的方法。由於這個模型主要是用於照片後處理，並且沒有提供訓練模型的功能，因此我們不會對它進行進一步的訓練。

4.去噪模型

MPRNet的訓練需要先將圖像預處理，包括圖像規一化、裁剪和調整大小等操作，以便輸入模型。模型使用編碼器-解碼器(encoder-decoder)的結構預測輸入圖像部位分割結果。其中所其中編碼器用於提取特徵，解碼器用於生成輸出分割圖像。利用交叉熵損失函數(Cross-entropy loss function)衡量模型輸出與真實值之間的差異訓練過程中，通過反向傳播算法來調整網絡參數，使得模型的輸出分割圖像與真實分割圖像之間的差異最小化。

由於我們重新訓練的模型效果並未比論文提供的模型效果好，因此我們使用論文原本提供的模型。

(二)成效分析

分析照片修復之成效。

1.均方誤差損失(Mean Squared Error Loss)

MSE是回歸問題中常用的損失函數，用來衡量預測值和真實值之間的差異。其計算方法是將預測值與真實值之間距離的平方求和，MSE的值越小，表示預測模型對實驗資料的描述具有更高的精確度。在本研究中，我們使用MSE來計算彩色照片的誤差，作為評估照片修復程度的依據。
$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

```
from skimage.metrics import mean_squared_error as compare_mse
import cv2

total=0.0
# 讀取兩張圖片
for i in range(35):
    img1 = cv2.imread('/media/tsmc/E/hgsh/Photo測試/3.jpg')
    img2 = cv2.imread('/media/tsmc/E/hgsh/DeOldify(all scale)/result_images(bandw)/3.jpg')
    mse_val = compare_mse(img1, img2)
    total += mse_val
total=total/int(35)
print(total)
```

圖八：使用自編代碼計算MSE(資料來源：研究者自製)

2.結構相似性指標(structural similarity index , SSIM index)

常用來評價除噪(De-noising)或除雨(De-raining)的模型或方法，將處理過後的圖與目標圖(Ground Truth)計算SSIM，若越接近1表示處理的越接近我們要的結果。計算方式為比較兩張圖的三個指標，亮度 (Luminance)、對比度(Contrast)和結構(Structure)，再將其相乘。

$$SSIM = \text{Luminance} * \text{Contrast} * \text{Structure}$$

$$\text{Luminance: } L(x, y) = \frac{2\mu_x\mu_y+C1}{\mu_x^2+\mu_y^2+C1}$$


$$\text{Contrast: } C(x, y) = \frac{2\sigma_x\sigma_y+C2}{\sigma_x^2+\sigma_y^2+C2}$$

$$\text{Structure: } S(x, y) = \frac{\mu_{xy}+C3}{\sigma_x \sigma_y +C3}$$

3.人為評估：

由於老照片並無原始照片得以比對、計算「排列模型順序」的均方誤差損失，因此本研究設計了一套基於李克特量表人為評估標準，李克特量表(Likert Scale)：受測者被要求指出對該題目所陳述的認同程度，或任何形式的主觀或客觀評價。

我們採用的是七個回應等級：強烈反對、不同意、有點不同意、既不同意也不反對、有點同意、同意、堅決同意。以103位受試者進行人為評估，並根據三種面向對輸出圖片進行給分，評估標準如下：

評比項目	評估標準	說明
建築物色彩自然度		建築物指新竹的景點，如：孔廟、新竹火車站、城隍廟等等。
景物(非主要建築物)色彩自然度		景物指新竹景點以外的景物，如：樹、天空、人等等。
清晰程度		清晰指有無噪點以及清晰程度。
	極度自然(7分)	

表一：人為評分表(資料來源：研究者自製)

(三)模型整合

上述提到的著色模型、去模糊化模型、去噪模型均為獨立的模型。在本步驟中，我們透過調整模型執行順序和觀察執行次數來尋找最佳的生成照片效果。此外，我們還將這些模型串聯起來，作為互動式介面的一部分，讓使用者可以直接使用這些模型進行圖像處理。模型整合如下：

1.排序：

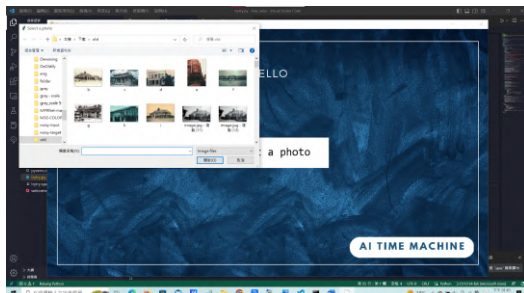
本研究將著色、降噪、去模糊三種功能經排列組合模型得到的六種組合，並藉由人為評估的方式評分不同排序方法對於修復照片之效果。

2.執行次數：

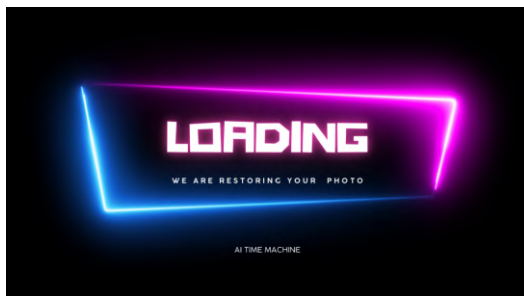
為了尋求最佳效果的平衡點，本研究進行了多次模型的執行與分析，為確保結果的可靠性，我們分別計算各模型與原圖的誤差並以方便觀察，而有最小誤差的次數便為接下來我們對照片修復的執行次數。

3.照片修復介面：

在找出最佳排列順序及最佳執行次數後，我們使用 Pygame 創建了一個互動介面。在介面中，我們添加了一個按鈕，讓使用者可以從檔案中選擇要修復的老照片。在模型運作期間，會顯示“loading”字樣，使畫面更加生動有趣。最終，介面上會展示產生的成果以及它們的存儲位置。此外，我們還添加了“continue”、“exit”按鈕，以讓使用者繼續輸入下一張待修復的老照片或退出現在畫面。我們也考慮到使用者可能會轉換多張照片，因此修復軟體轉出之照片我們都有按照轉換之時間命名。



圖九：照片選擇頁面(資料來源：研究者自製)



圖十：照片修復加載頁面(資料來源：研究者自製)



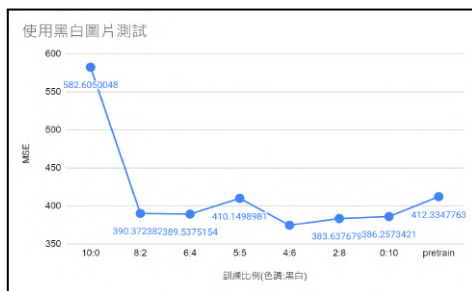
圖十一：照片修復完成顯示頁面(資料來源：研究者自製)

肆、研究結果

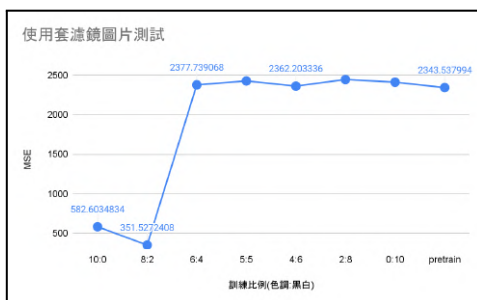
一、照片修復表示

(一)著色模型

1.我們使用未在訓練資料中的40張照片，放入模型中並計算每張圖片的MSE以及SSIM並取其平均，得到下表(表二、表三)。透過下表可觀察到，雖然訓練資料比例為套濾鏡照片：黑白照片比例是8：2之模型在使用黑白圖片測試時表現並不是最佳，但是在使用套濾鏡圖片測試時卻有最好的表現，且與其他模型相差甚遠，因此我們選用此模型作為我們最終使用的模型。




















表二：使用黑白圖片測試不同訓練資料比例模型的MSE趨勢圖









表三：使用套濾鏡圖片測試不同訓練資料比例模型的MSE趨勢圖

2.部分圖片著色之結果比對，如下表：

























表四：著色黑白照片對比原始照片MSE、SSIM

比例 (色調: 黑白)	原始照片A (新竹孔廟)	黑白照片A	訓練結果A	MSE	SSIM
0:10				862.6 63056	0.989 272
2:8				913.0 36520 4	0.974 903
4:6				899.7 30504 6	0.974 032
5:5				988.2 38707 6	0.976 325
6:4				822.0 11062 2	0.960 971
8:2				963.8 38982 6	0.974 803
10:0				2467. 59151 8	0.973 133
預訓練				877.9 23069 4	0.970 108



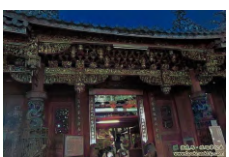







表五：著色黑白照片對比原始照片MSE、SSIM

比例 (色調: 黑白)	原始照片B (新竹城隍廟)	黑白照片B	訓練結果B	MSE	SSIM
0:10				585.6 41824 5	0.980 312
2:8				531.7 53329 8	0.976 238
4:6				604.2 20612 5	0.973 670
5:5				1194.3 3577	0.974 023
6:4				464.4 53935 2	0.973 344
8:2				676.6 89858 7	0.977 950
10:0				719.7 24351 9	0.973 519
預訓練				1237. 24544 9	0.966 181

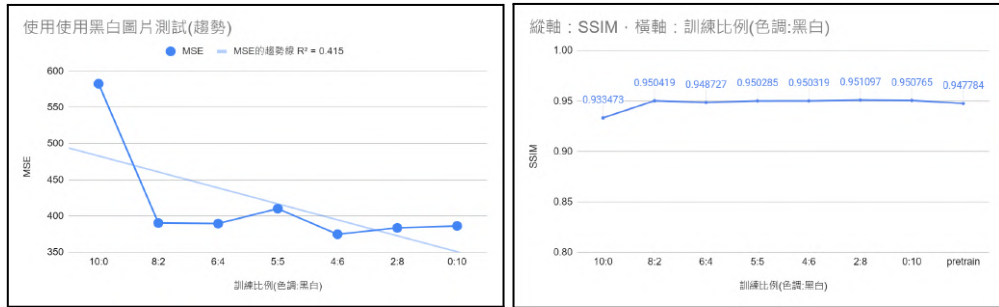
表六：著色套濾鏡照片對比原始照片MSE、SSIM

比例 (色調: 黑白)	原始照片A (新竹孔廟)	套濾鏡照片A	訓練結果A	MSE	SSIM
0:10				4085. 74770 3	0.795 925
2:8				4157. 25241 9	0.795 173
4:6				3474. 29752 4	0.795 995
5:5				3479. 30968 8	0.797 560
6:4				3704. 13531 9	0.798 138
8:2				626.3 45955 5	0.984 158
10:0				2467. 62838 5	0.989 271
預訓練				3342. 02676 4	0.798 159

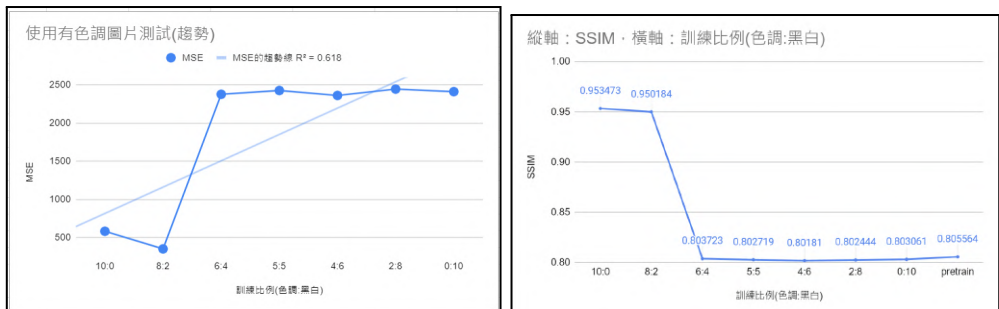
表七：著色套濾鏡照片對比原始照片MSE、SSIM

比例 (色調: 黑白)	原始照片B (新竹城隍廟)	套濾鏡照片B	訓練結果B	MSE	SSIM
0:10				1181.3 9662	0.879 631
2:8				1134.8 25538	0.888 311
4:6				1110.7 83329	0.886 322
5:5				1194.3 3577	0.892 892
6:4				1047. 87581 3	0.886 871
8:2				599.4 27995	0.969 301
10:0				719.7 09773 3	0.980 312
預訓練				1482. 15665 2	0.922 242

3.由下表(表八、表九、表十、表十一)可知，當訓練資料中有套濾鏡之照片比例上升時，使用黑白圖片測試模型的效果又越來越差的趨勢；而使用套濾鏡圖片測試模型則有越來越好的趨勢。

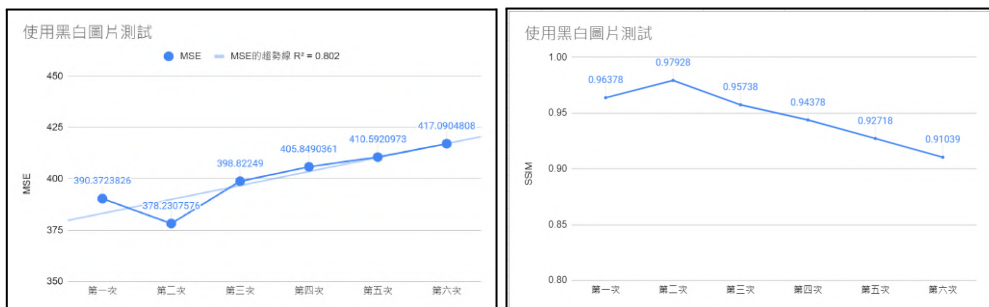


表八、表九：使用黑白圖片測試與預訓練模型之比較(左：MSE，右：SSIM)

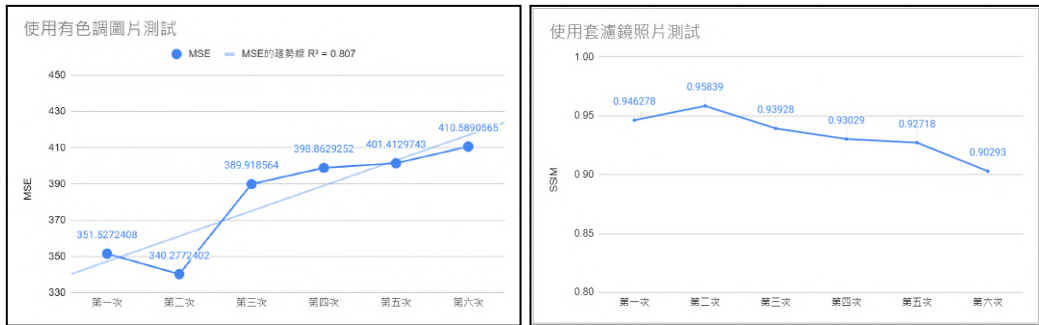


表十、表十一：使用套濾鏡圖片測試與預訓練模型之比較(左：MSE，右：SSIM)

4.執行不同次數對於修復照片之影響：由下表(表十二、表十三)可知，無論使用套濾鏡之圖片亦或是使用黑白照片測試，當將著色功能重複執行兩次時，模型有最佳的修復效果。



表十二、表十三：使用黑白圖片測試著色模型重複執行不同次數的效果

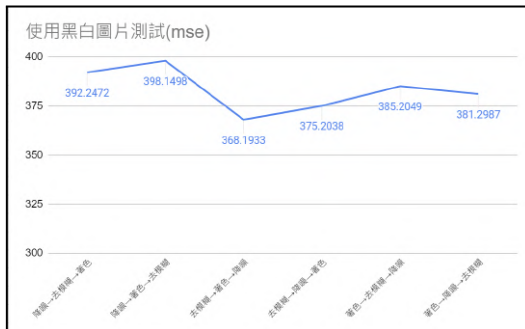


表十四、五：使用套濾鏡圖片測試著色模型重複執行不同次數的效果

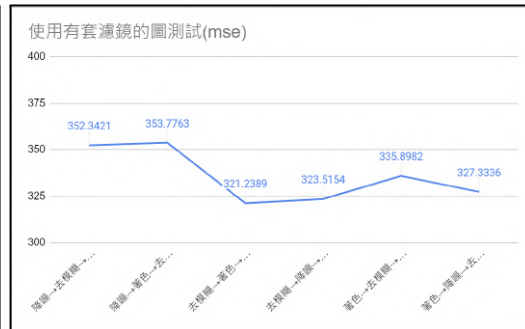
二、模型整合效果

(一)探討排列順序對修復結果的影響

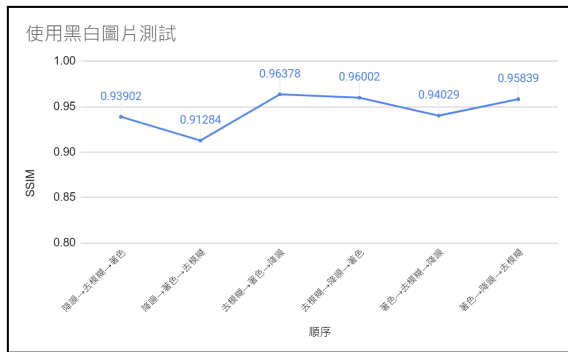
我們將三種修復功能：黑白圖片上色、去模糊、降噪，進行排列組合後，得到六種不同的修復方式，分別是「著色→去模糊→降噪」，「著色→降噪→去模糊」，「去模糊→著色→降噪」，「去模糊→降噪→著色」，「降噪→著色→去模糊」，「降噪→去模糊→著色」。為了量化排序的誤差，我們將清晰的彩色照片轉模糊並加入噪點後，重新放入模型計算MSE(表十六、表十七)、SSIM(表十八、表十九)。我們也綜合比較40張舊照片的修復以及103位受測者的評分結果，由下表(表二十)發現將黑白照片先進行去模糊處理，接著執行著色，最後進行去噪處理，得到的修復效果大眾最接受。以下是部分的比較結果：



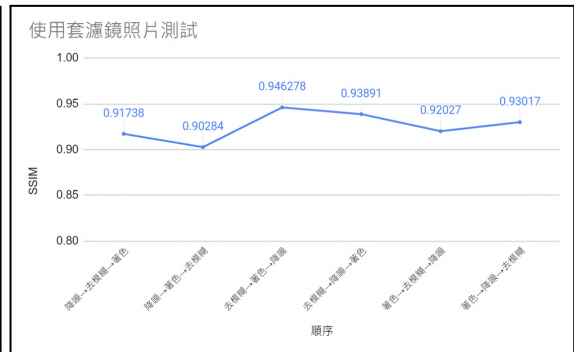
表十六：
使用黑白圖片測試不同排列順序MSE



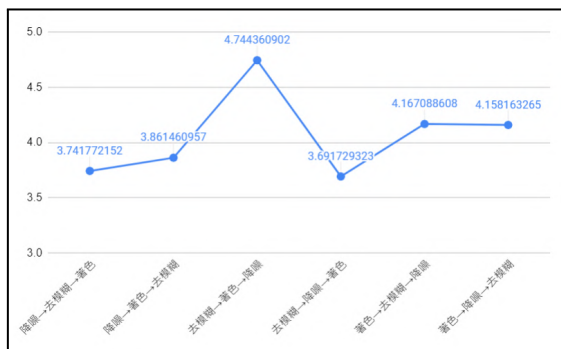
表十七：
使用有套濾鏡圖片測試不同排列順序MSE



表十八：使用黑白圖片測試不同順序之照片的ssim




















表十九：使用有套濾鏡圖片測試不同順序之照片的ssim





















表二十：受試者對不同排列順序的評分結果

表二十一：新竹消防博物館

	原始照片	預訓練結果	訓練結果	評估分數
降噪 ↓ 去模糊 ↓ 著色				3.18
降噪 ↓ 著色 ↓ 去模糊				3.12
去模糊 ↓ 著色 ↓ 降噪				5.08
去模糊 ↓ 降噪 ↓ 著色				3.2
著色 ↓ 去模糊 ↓ 降噪				3.82
著色 ↓ 降噪 ↓ 去模糊				4.08

表二十二：新竹火車站

	原始照片	預訓練結果	訓練結果	評估分數
降噪 ↓ 去模糊 ↓ 著色				4.20
降噪 ↓ 著色 ↓ 去模糊				4.28
去模糊 ↓ 著色 ↓ 降噪				4.59
去模糊 ↓ 降噪 ↓ 著色				3.9
著色 ↓ 去模糊 ↓ 降噪				3.88
著色 ↓ 降噪 ↓ 去模糊				3.98

(二)訓練後模型與預訓練模型之比較

以下我們將分別使用新竹以及外地的老照片、黑白照片、套濾鏡照片來測試訓練過後的成果，這裡套濾鏡照片是使用PIL濾鏡並非訓練時使用的opencv的濾鏡。我們對比訓練後模型以及預訓練模型，最後計算其MSE和附上景點的近期照片作為對比，使更能了解模型的上色效果。

表二十三：使用新竹老照片測試，比較訓練後模型及預訓練模型

	原始照片	預訓練模型	AI在地化訓練	近期照片比對
圖片一 (新竹火車站)				
圖片二 (新竹城隍廟)				
圖片三 (新竹東門城)				
圖片四 (新竹孔廟)				
圖片五 (新竹美術館)				

表二十四：使用外地老照片測試，比較訓練後模型及預訓練模型

	原始照片	預訓練模型	AI在地化訓練	近期照片比對
圖片一 (打狗英國領事館官邸)				
圖片二 (億載金城)				
圖片三 (台中火車站)				
圖片四 (赤崁樓)				
圖片五 (台北賓館)				

表二十五：使用新竹黑白照片測試，比較訓練後模型及預訓練模型

mse	原始照片	黑白照片	預訓練模型	AI在地化訓練
圖片一 (新竹城隍廟)				
MSE	N/A	N/A	2024.1827	883.1847
SSIM	N/A	N/A	0.664723	0.731521
圖片二 (新竹東門城)				
MSE	N/A	N/A	1758.4628	302.4728
SSIM	N/A	N/A	0.692171	0.740573
圖片三 (新竹孔廟)				
MSE	N/A	N/A	1946.2681	648.2846
SSIM	N/A	N/A	0.629015	0.461526
圖片四 (新竹火車站)				
MSE	N/A	N/A	1989.2846	729.4728
SSIM	N/A	N/A	0.742357	0.768071
圖片五 (新竹美術館)				
MSE	N/A	N/A	1678.2846	518.2746
SSIM	N/A	N/A	0.852466	0.979759

表二十六：使用外地黑白照片測試，比較訓練後模型及預訓練模型

	原始照片	黑白照片	預訓練模型	AI在地化訓練
圖片一 (紅毛城)				
MSE	N/A	N/A	1294.2647	593.1846
SSIM	N/A	N/A	0.938807	0.984199
圖片二 (億載金城)				
MSE	N/A	N/A	1591.4627	583.1748
SSIM	N/A	N/A	0.69365	0.980598
圖片三 (台北北門)				
MSE	N/A	N/A	2463.2718	482.2748
SSIM	N/A	N/A	0.814691	0.98891
圖片四 (李騰芳古宅)				
MSE	N/A	N/A	1728.3627	647.2748
SSIM	N/A	N/A	0.93763	0.993419
圖片五 (台中火車站)				
MSE	N/A	N/A	1493.2848	647.2748
SSIM	N/A	N/A	0.927261	0.995053

表二十七：使用新竹套濾鏡照片測試，比較訓練後模型及預訓練模型





	原始照片	套濾鏡照片	預訓練模型	AI在地化訓練
圖片一 (新竹城隍廟)				
MSE	N/A	N/A	2017.2748	428.2784
SSIM	N/A	N/A	0.883234	0.930944
圖片二 (新竹東門城)				
MSE	N/A	N/A	2519.2748	284.4728
SSIM	N/A	N/A	0.950846	0.974585
圖片三 (新竹孔廟)				
MSE	N/A	N/A	2729.1849	729.1748
SSIM	N/A	N/A	0.925138	0.951163
圖片四 (新竹火車站)				
MSE	N/A	N/A	2719.2482	529.1748
SSIM	N/A	N/A	0.942334	0.914653
圖片五 (新竹美術館)				
MSE	N/A	N/A	3391.2482	529.1847
SSIM	N/A	N/A	0.876783	0.917767

表二十八：使用新竹套濾鏡照片測試，比較訓練後模型及預訓練模型

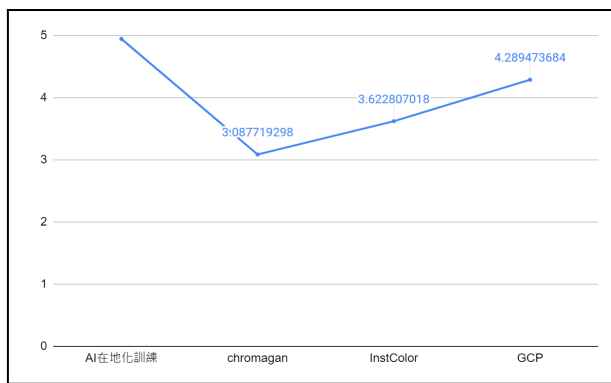
	原始照片	套濾鏡照片	預訓練模型	AI在地化訓練
圖片一 (紅毛城)				
MSE	N/A	N/A	3627.2784	629.5628
SSIM	N/A	N/A	0.895782	0.880276
圖片二 (億載金城)				
MSE	N/A	N/A	2994.2748	657.2678
SSIM	N/A	N/A	0.681518	0.890336
圖片三 (台北北門)				
MSE	N/A	N/A	3183.4627	391.8294
SSIM	N/A	N/A	0.789721	0.954401
圖片四 (李騰芳古宅)				
MSE	N/A	N/A	2047.2748	462.2819
SSIM	N/A	N/A	0.921057	0.978648
圖片五 (台中火車站)				
MSE	N/A	N/A	1507.3472	455.7534
SSIM	N/A	N/A	0.90443	0.932138

表二十九：AI在地化模型與其他論文成果之比較

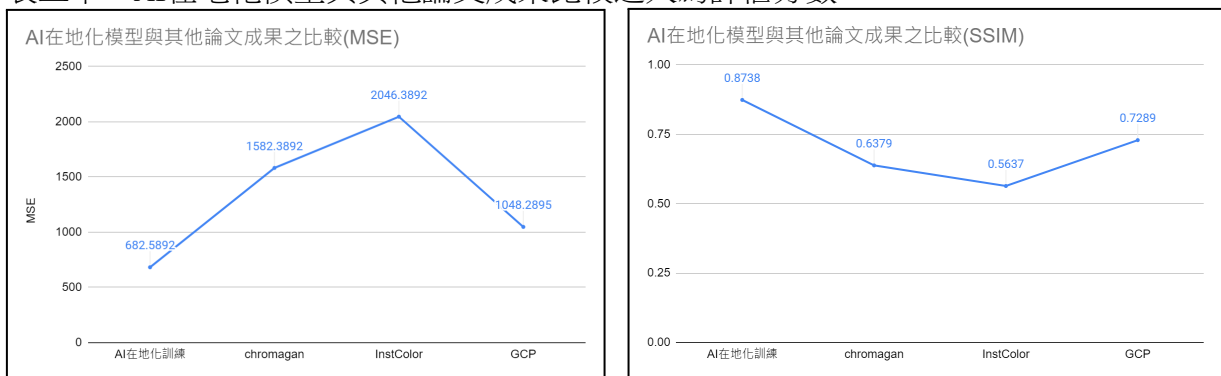
	AI在地化訓練	chromagan	InstColor	GCP
新竹東門城(黑白)				
MSE	302.4728	1974.2789	1037.2894	1284.2942
SSIM	0.740573	0.5738	0.7839	0.7583
人為評估	4.315789474	2.736842105	5.105263158	4.789473684
新竹東門城(濾鏡)				
MSE	284.4728	1988.2985	1647.2785	1547.2795
SSIM	0.974585	0.6894	0.8295	0.8275
人為評估	4.210526316	2.736842105	4.842105263	4.473684211
台北北門(黑白)				
MSE	482.2748	1658.9891	1758.9289	1648.3985
SSIM	0.98891	0.6738	0.7849	0.8395
人為評估	5.315789474	2.947368421	3.473684211	3.315789474
台北北門(濾鏡)				
MSE	391.8294	1895.2895	1549.3892	1478.3895
SSIM	0.954401	0.5728	0.6839	0.6783
人為評估	5.473684211	2.789473684	2.789473684	2.684210526
新竹美術館(老照片)				
人為評估	5.052631579	4.421052632	3.368421053	3.210526316

億載金城(老照片)				
人為評估	5.315789474	2.894736842	2.157894737	4.526315789
總平均MSE	682.5892	1582.3892	2046.3892	1048.2895
總平均SSIM	0.8738	0.6379	0.5637	0.7289

由上表可知，不論是使用量化的誤差像是MSE以及SSIM，或是使用人為評估的方式，我們的模型都有比較佳的表現。



表三十：AI在地化模型與其他論文成果比較之人為評估分數



表三十一、表三十二：AI在地化模型與其他論文成果比較之MSE、SSIM誤差

伍、討論

一、探討黑白轉彩色模型輸出結果

(一)使用全是有色調的照片訓練之模型修復黑白照片時的效果與使用含有黑白照片的訓練資料訓練之模型相差甚遠：

我們認為，造成此結果主要由於：此模型並無使用任何單純黑白的照片進行訓練，只針對有特殊色彩的圖進行學習，因此缺乏對其他細節(如：純黑白照片)的參考和學習，導致照片修復效果不佳。

(二)某些模型照片修復結果朦朧：

由表七可知，當使用有色調照片測試訓練資料之有色調圖片：黑白圖片比例為0：10、2：8、4：6、5：5、6：4的模型時，輸出之照片彷彿多了一層白色的濾鏡，且部分使用套濾鏡照片測試這些模型，均會出現此狀況。

我們推論這是由於本研究用於將黑白圖片轉換為有色調圖片使用的部分濾鏡過於強烈，使模型過度關注輸入圖片之色彩而忽略了黑白照片中的紋理和細節，導致修復結果不如預期。

(三)探討訓練結果以及預訓練結果

1.經過觀察每一張測試的圖片，發現經過訓練的模型輸出的圖針對屋瓦或是紅磚牆等特徵有比較好的效果。

我們推論這是因為大部分的古蹟都有紅屋瓦、紅磚的特徵，因此使的訓練後的模型針對此類特徵有較好的輸出效果。

2.經過觀察每一張測試的圖片，發現經過訓練的模型輸出的圖針對外圍景物(如：樹木)等特徵有比較差的的效果。

我們推論這是因為訓練模型時，我們過度針對「本地建築物」進行訓練，使模型忽略了較細節部分的修復，使模型對外圍景物的修復效果較差。

二、探討整體輸出效果

(一)探討排列順序對照片修復結果的影響

經過調查黑白照片先進行去模糊處理，接著執行著色，最後進行去噪處理，可以得到最佳的修復效果。我們推論，由於老照片大多因拍照技術還不夠成熟，歷史的照片均十分模糊、噪點多，因此當修復照片時，先將照片去模糊，能夠提高照片的可見度，因此修復照片時能夠提高照片上色的效能。

(二)重複執行兩次著色功能的效果較好

本研究經過比較發現在重複執行第二次時效果最佳，並且在重複兩次後，隨著執行次數的增加效果有越來越差的趨勢。

我們推論，第一次執行時，模型還沒有完全修復所有細節，例如樹的尖端、部分梁柱等等，所以第二次執行時已可以完全利用模型學習到的顏色，從而提高色彩真實度。然而，當執行次數增加時，模型可能會開始因為過度的修復導致輸出結果與建築實際顏色有落差、失真。

(三)探討使用本地照片以及使用外地照片測試

經過測試，發現我們訓練的模型對於不論是本地老照片或是外地老照片均有不錯的修復效果。

我們推論，這是因為全臺灣大部分的古蹟都有差不多的特徵，例如紅磚、橘紅屋瓦等，導致模型對於台灣的照片表現均不錯。

陸、結論

(一)結論

本研究旨在使用多個深度學習模型來修復照片中的缺陷。在實踐、分析和討論的過程中，我們整合DeOldify、TLC和IMPRNET模型的程式和功能，並利用pygame製作成一個互動介面。在實作測試中，我們發現了這些模型可能存在的一些限制，並且在未來可以進一步改進。根據我們對大量資料進行實驗後的結果，得出以下結論：

1.著色模型：

(1)以比例為百分之八十為有色調照在訓練集中的比例重新訓練DeOldify模型有最好的訓練效果。

(2)將著色修復模型重複執行多次後，經比較測試，重複執行兩次可以得到最佳的修復效果。

2.模型整合：

(1)將三種照片修復模型以不同順序應用後，經表單調查，黑白照片先進行去模糊處理，接著執行著色，最後進行去噪處理，可以得到最佳的修復效果。

(2)模型對於本地照片以及外地照片均有不錯的修復效果。

(二)未來展望

- 1.增加訓練次數和訓練集圖片數量，以更精準修復老照片。
- 2.增加模型訓練的地點地標將訓練集擴增至全台各地歷史建築，使的修復範圍變得更加廣泛，進而讓更多的照片能夠得到完整的修復。
- 3.增加不同的排序方法探討更加的修復效果，例如著色→去模糊化→著色→降噪、著色→降噪→著色→去模糊化等等。
- 4.將修復的結果3D立體化，以各種角度觀賞歷史地標亦有身歷其境之感。

柒、參考文獻資料

- 1.Salmona, A., Bouza, L., & Delon, J. (2022). DeOldify: A Review and Implementation of an Automatic Colorization Method. *Image Processing On Line*, 12, 347-368.
- 2.Chu, X., Chen, L., Chen, C., & Lu, X. (2022, November). Improving image restoration by revisiting global information aggregation. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII* (pp. 53-71). Cham: Springer Nature Switzerland.
- 3.Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M. H., & Shao, L. (2021). Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14821-14831).
- 4.[機器學習 ML NOTE]Generative Adversarial Network, GAN 生成對抗網路. (2018, September 15). GGWithRabitLIFE. <https://www.cnblogs.com/popodynasty/p/14391442.html>
- 5.什麼是均方誤差 Mean-Square Error, MSE ? . (2019, December 21). 新創駭客. <https://www.cnblogs.com/popodynasty/p/14391442.html>
- 6.Jason chen. (2019, January 23). 【影像處理】峰值信噪比 Peak Single-to-Noise Ratio. Weebly.Com. <https://jason-chen-1992.weebly.com/home/-peak-single-to-noise-ratio>。

【評語】 052506

1. 本作品起源於老照片因年代久遠，多半泛黃或發霉，不是純黑白照片，導致效果不佳。因此，作品將訓練資料轉為有色調的圖片，使其能夠修正有色調的照片修復不佳的問題，以提高修復效果。修復的目的是要加上彩色。作品具實用性。
2. 本作品可以持續改善加強的建議如下：
 - (1) 如何將模型在地化訓練，以推展到其他的城市、文化、不同的時光隧道。
 - (2) 如何有個客觀有效率的評估機制來輔助整個計畫的進行。

作品海報

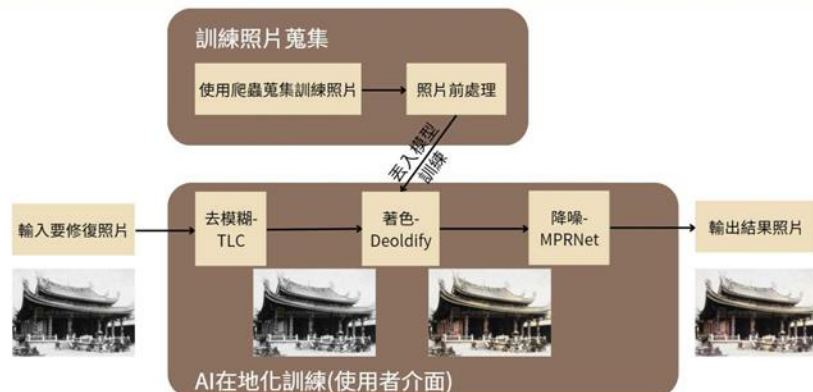
摘要

目前網路上流傳許多使用人工智慧修復照片的網站或應用軟體。然而，由於這些訓練資料多數來自國外，導致修復台灣「本地」照片的效果欠佳。此外，許多老舊照片因氧化、潮濕而泛黃，使得修復程序比起修復純黑白相片更加困難。因此，本研究旨在建立一專門修復本地照片的機器學習模型，主要分為以下三個部分：首先，使用機器學習模型對老舊照片進行修復，包括**著色**、**清晰化和降噪**；其次，分析使用不同比例之有色調照片(模擬泛黃照片)訓練模型的效果；最後，研究不同的**修復順序**(著色、清晰化、降噪)和**模型執行次數**對照片修復效果的影響，發現「**去模糊、著色、降噪**」的順序修復效果最佳。

研究目的

研究流程

- (一)對蒐集到效果最佳的現有模型**在地化訓練**。
- (二)分析模型**執行次數**對照片成果的影響。
- (三)分析修復功能**執行順序**對照片成果的影響。
- (四)分析**有色調照片**在訓練集的**比例**對照片成果的影響。
- (五)**整合**多種修復功能以簡化照片修復的過程。

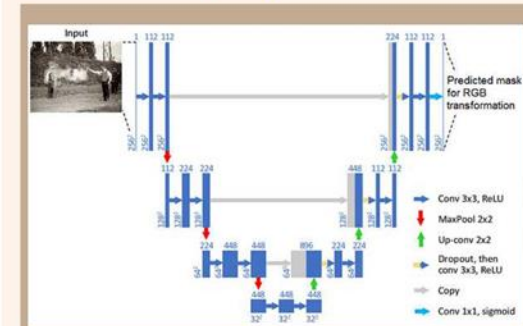
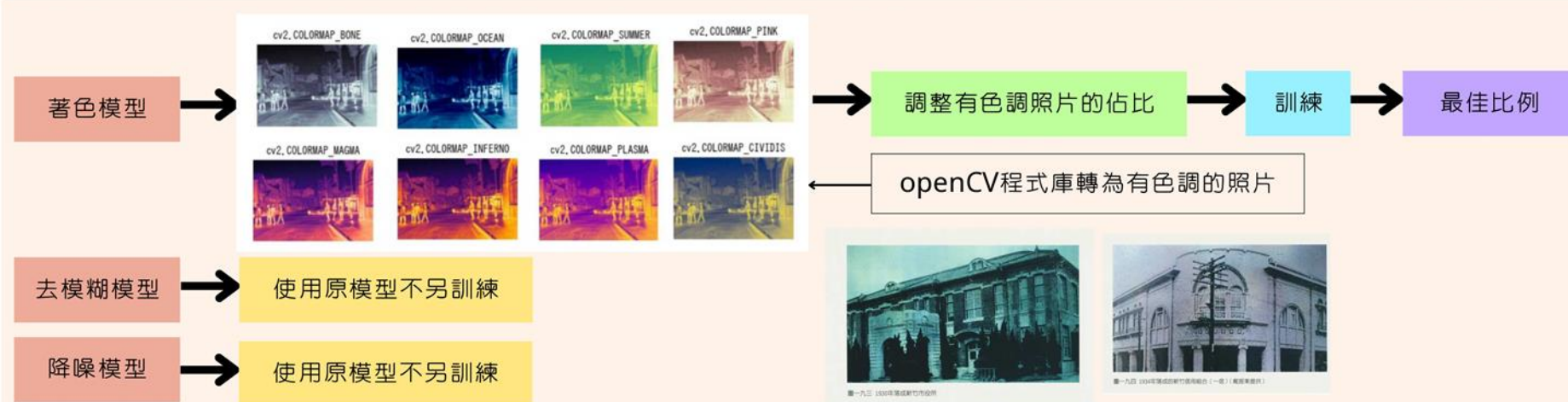


研究過程或方法

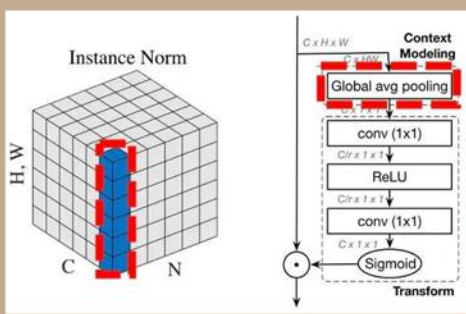
一、在地化資料庫建立

從新竹現存的**歷史古蹟**中選取九個地點，包括孔廟、新竹火車站、東門城、城隍廟、國民戲院、淨業院、第一信用合作大樓、新竹州廳(現新竹市政府)、新竹美術館，並利用自編爬蟲程式透過google伺服器爬取彩色圖片作為訓練資料集，並淘汰不相關的圖片，最後進行模型的訓練。

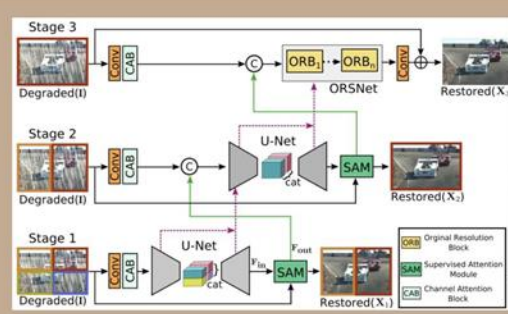
二、AI模型訓練



圖一：Deoldify 模型流程圖



圖二：TLC 模型流程圖



圖三：MPRNet 模型流程圖

- (一)著色模型：使用Deoldify模型。(Salmona et al., IPOL 2022)此模型是一個用深度學習技術的AI模型，用於恢復老照片的顏色和細節，主要使用生成對抗網絡 (GAN) 和殘差神經網絡 (ResNet)。
- (二)去模糊模型：使用TLC模型。(Chu et al., ECCV 2022)此模型是一種執行圖像恢復任務的深度學習模型，主要方法是將照片切成小塊圖片然後逐一進行處理的方法。
- (三)去噪模型：使用MPRNet。(Zamir et al., CVPR 2021)此模型是基於深度學習的圖像增強模型，使用MC-DNN和注意力機制進行超分辨率、去霧、去噪和增強等任務。

三、分析照片修復之成效

- (一)均方誤差損失(Mean Squared Error Loss) 在本研究中，我們使用MSE來計算彩色照片的誤差，作為評估照片修復程度的依據。
- (二)結構相似性指標(SSIM index) 在本研究中，我們亦將SSIM作為評估照片修復程度的指標。

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

$$\text{Luminance: } L(x, y) = \frac{2\mu_x\mu_y + C1}{\mu_x^2 + \mu_y^2 + C1}$$

$$\text{Contrast: } C(x, y) = \frac{2\sigma_x\sigma_y + C2}{\sigma_x^2 + \sigma_y^2 + C2}$$

$$\text{Structure: } S(x, y) = \frac{\mu_{xy} + C3}{\sigma_x \sigma_y + C3}$$

(二)人為評估

由於老照片並無原始照片得以比對、計算「**排列模型順序**」的均方誤差損失，因此本研究設計了一套基於李克特量表人為評估標準。

評比項目	評估標準	說明
建築物色彩自然度	極度不自然(1分)	建築物指新竹的景點，如：孔廟、新竹火車站、城隍廟等等。
景物(非主要建築物)色彩自然度	↓	景物指新竹景點以外的景物，如：樹、天空、人等等。
清晰程度		極度自然(7分)

四、模型整合與效能優化

(一)執行次數

本研究執行多次模型，我們分別**計算各模型與原圖的mse、ssim誤差**以方便觀察，而有最小誤差的次數便為接下來我們對照片修復的執行次數。

(二)模型整合

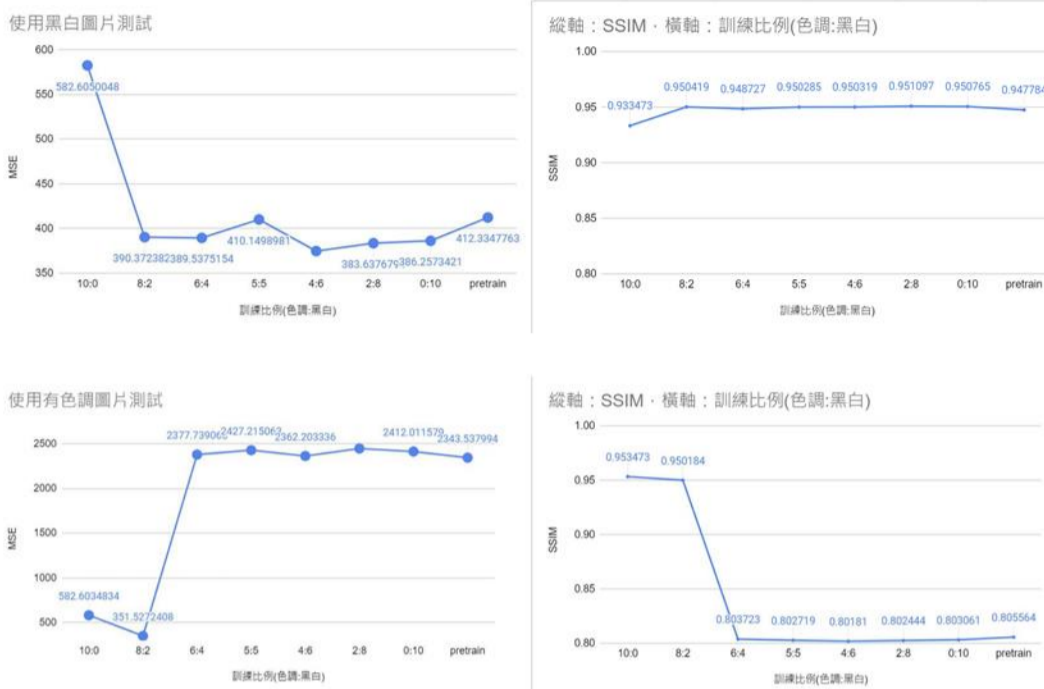
著色、降噪、去模糊三種功能經排列組合模型得到的六種組合，並藉由**人為評估**的方式評分不同排序方法對於修復照片之效果。

研究結果

一、照片修復表示

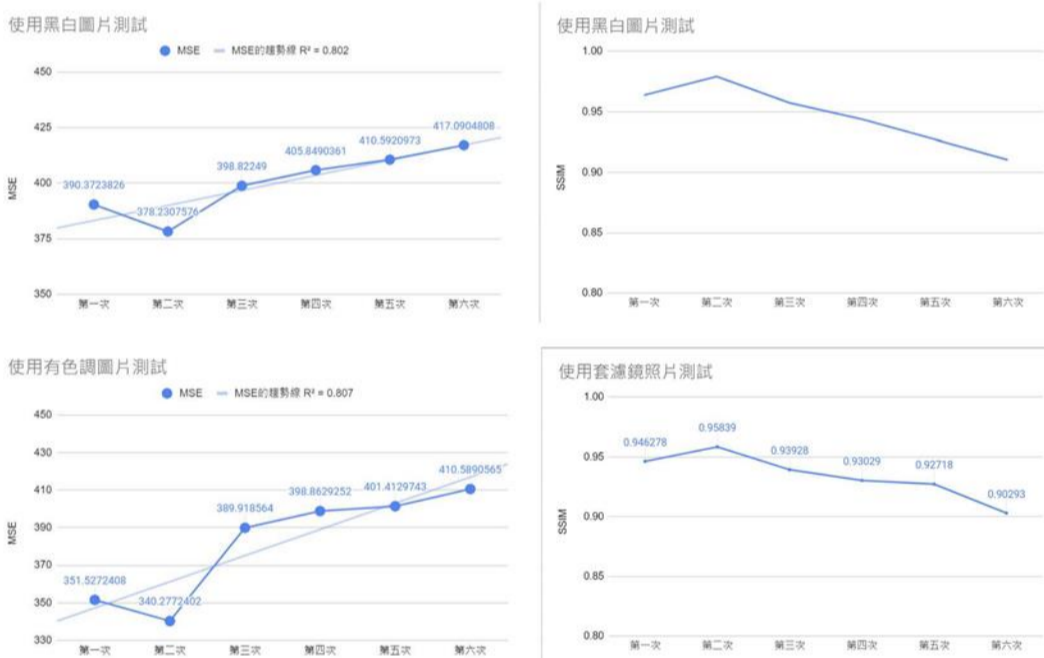
(一) 著色模型

使用未在訓練資料中的40張照片，放入模型中並計算每張圖片的MSE、SSIM並取其平均，得到下表。並選擇訓練資料比例為有色調照片：黑白照片比例是8：2之模型為最終使用的模型。



(二) 執行不同次數對於修復照片之影響

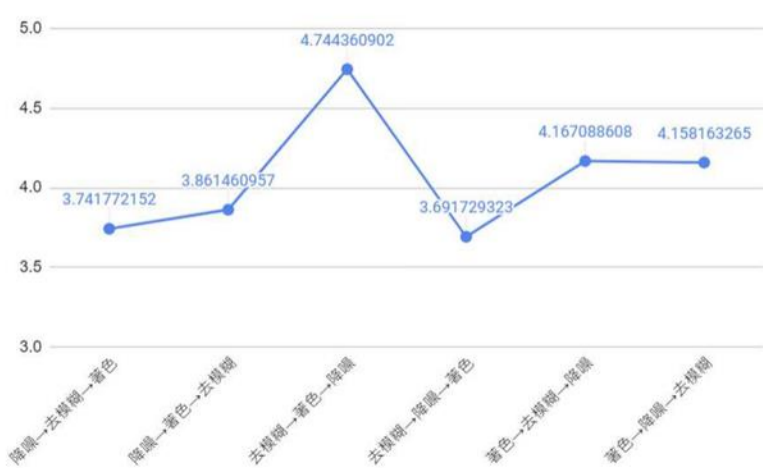
由下表可知，無論使用有色調之圖片亦或是使用黑白照片測試，當將著色功能重複執行兩次時，模型有最佳的修復效果。



二、模型整合效果

將三種修復功能：黑白圖片上色、去模糊、降噪，進行排列組合後，得到六種不同的修復方式，分別是「著色→去模糊→降噪」，「著色→降噪→去模糊」，「去模糊→著色→降噪」，「去模糊→降噪→著色」，「降噪→著色→去模糊」，「降噪→去模糊→著色」。

綜合比較40張舊照片的修復以及103位受測者的評分結果後，「去模糊→著色→降噪」，得到的修復效果大眾最接受。



1. 著色黑白照片對比原始照片MSE

比例 (色調: 黑白)	原始照片A (新竹孔廟)	黑白照片A	訓練結果A	MSE	SSIM
0:10				862.6 63056	0.989 272
2:8				913.0 36520 4	0.974 903
4:6				899.7 30504 6	0.974 032
5:5				988.2 38707 6	0.976 325
6:4				822.0 11062 2	0.960 971
8:2				963.8 38982 6	0.974 803
10:0				2467. 59151 8	0.973 133
預訓練				877.9 23069 4	0.970 108

2. 著色有色調照片對比原始照片MSE

比例 (色調: 黑白)	原始照片A (新竹孔廟)	套濾鏡照片A	訓練結果A	MSE	SSIM
0:10				4085. 74770 3	0.795 925
2:8				4157. 25241 9	0.795 173
4:6				3474. 29752 4	0.795 995
5:5				3479. 30968 8	0.797 560
6:4				3704. 13531 9	0.798 138
8:2				626.3 45955 5	0.984 158
10:0				2467. 62838 5	0.989 271
預訓練				3342. 02676 4	0.798 159

研究結果

討論

一、探討著色模型輸出結果

- (一)使用全是有色調的照片訓練之模型修復黑白照片時的效果與使用含有黑白照片的訓練資料訓練之模型相差甚遠。
- (二)某些模型照片修復結果朦朧。
- (三)經過訓練的模型輸出的圖針對屋瓦或是紅磚牆等特徵有比較好的效果。
- (四)經過訓練的模型輸出的圖針對外圍景物(如：樹木)等特徵有比較差的的效果。

二、探討整體輸出效果

- (一)「去模糊→著色→降噪」，得到的修復效果大眾最接受。
- (二)訓練的模型對於不論是本地老照片或是外地老照片均有不錯的修復效果。

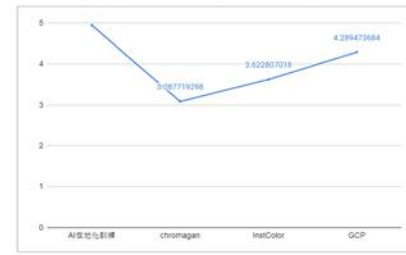
結論

一、著色模型：

- (一)使用比例為百分之八十為有色調照訓練 DeOldify模型有最好的訓練效果。
- (二)著色修復重複執行兩次得到最佳修復效果。

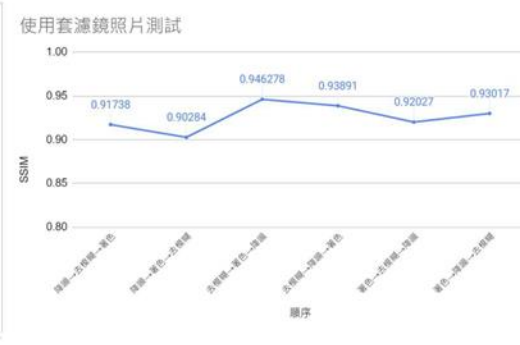
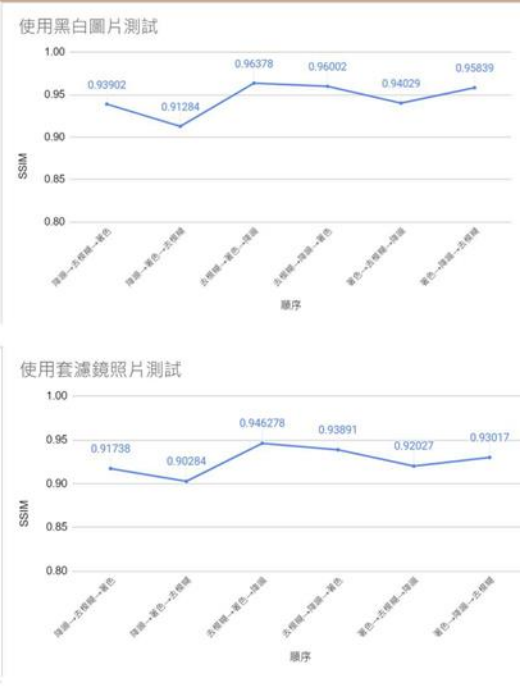
二、模型整合：

- (一)將三種照片修復模型以不同順序應用後，經表單調查，黑白照片先進行去模糊處理，接著執行著色，最後進行去噪處理，可以得到最佳的修復效果。
- (二)模型對於本地照片以及外地照片均有不錯的修復效果。
- (三)與其他論文模型之比較結果也相當出色。



左圖：
論文比較：人為評估

	AI 在地化訓練	chromagan	InstColor	GCP
新竹東門城 (黑白)				
MSE	302.4728	1974.2789	1037.2894	1284.2942
SSIM	0.740573	0.5738	0.7839	0.7583
人為評估	4.315789474	2.736842105	5.105263158	4.789473684
新竹東門城 (濾鏡)				
MSE	284.4728	1988.2985	1647.2785	1547.2795
SSIM	0.974585	0.6894	0.8295	0.8275
人為評估	4.210526316	2.736842105	4.842105263	4.473684211
台北北門(黑白)				
MSE	482.2748	1658.9891	1758.9289	1648.3985
SSIM	0.98891	0.6738	0.7849	0.8395
人為評估	5.315789474	2.947368421	3.473684211	3.315789474
台北北門(濾鏡)				
MSE	391.8294	1895.2895	1549.3892	1478.3895
SSIM	0.954401	0.5728	0.6839	0.6783
人為評估	5.473684211	2.789473684	2.789473684	2.684210526
新竹美術館 (老照片)				
人為評估	5.052631579	4.421052632	3.368421053	3.210526316
億載金城(老照片)				
人為評估	5.315789474	2.894736842	2.157894737	4.526315789
總平均 MSE	682.5892	1582.3892	2046.3892	1048.2895
總平均 SSIM	0.8738	0.6379	0.5637	0.7289

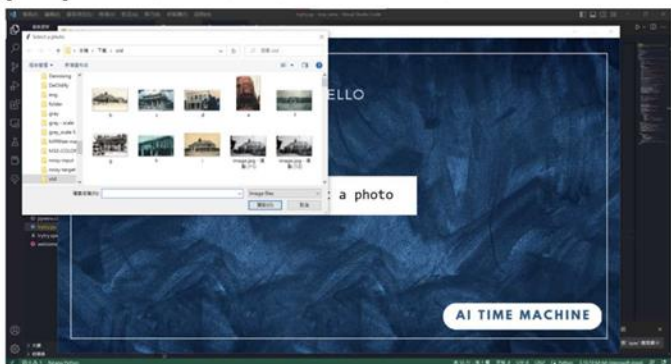


1. 排列順序對修復結果的影響

	原始照片	現有模型	AI在地化訓練	評估分數
降噪 ↓ 去模糊 ↓ 著色				3.18
降噪 ↓ 著色 ↓ 去模糊				3.12
去模糊 ↓ 著色 ↓ 降噪				5.08
去模糊 ↓ 降噪 ↓ 著色				3.2
著色 ↓ 去模糊 ↓ 降噪				3.82
著色 ↓ 降噪 ↓ 去模糊				4.08

三、使用者介面

(一) 選擇要修復的照片



(二) 顯示已修復之照片



(三) 按下"continue"按鍵可以再次選擇照片

