Street Image Inpainting with Local Binary Pattern and Spatial Attention

Yi-Yun Liu[1], Ya-Ting Hsu[2], Chow-Sing Lin[3,*]

Department of Computer Science and Information Engineering
National University of Tainan
Tainan, 70005, Taiwan

E-mail : [1] a0958581118@gmail.com
[2] susan900501@gmail.com
[3] mikelin@mail.nutn.edu.tw

Abstract

The existing image inpainting methods, such as traditional darkroom techniques and Photoshop inpainting techniques, all require time-consuming manual restoration. The use of automatic restoration functions often result in incomplete predicted restoration structures, leading to unsatisfactory restoration results. To effectively solve this issue, our project first allows users to simulate the damaged area (masked area) of a photo by simply covering it with a brush. Then, we use Local Binary Pattern (LBP) Learning Network to generate the predicted region repair structure through the Unet++ framework and learn the image and spatial information through Gated Convolution with Spatial Attention. We finally use the Coarse-to-Fine method to perform Image Inpainting Network to repair the masked region. Compared with the results of our project, the prediction results of Photoshop and the referenced work are less accurate in repairing the existing structure, and the latter also has obvious color difference. In addition, compared with the referenced work, the repair results of this project were improved by 2.4% and 14.6% to 0.9848 and 38.82 in terms of SSIM and PSNR, respectively.

Keywords ： Image Inpainting, Local Binary Pattern, Spatial Attention, Unet++, Gated Convolution, Deep Learning.

---

* Corresponding author: mikelin@mail.nutn.edu.tw
DOI ： 10.53106/222344892023101302005

利用局部二值模型及空間注意力進行街景圖像修復

劉佾雲，許雅婷，林朝興*

國立臺南大學 資訊工程系

摘要

基於現有的圖像修復方法，如傳統暗房技術以及 Photoshop 修復技術等，皆較需費時的人工修補，而若使用自動修補功能，亦經常造成預測修補結構不完整，導致修補效果不理想。為了有效解決上述的問題，我們的專題先讓使用者透過筆刷進行簡單的塗抹覆蓋，模擬照片破損之區域(mask 區域)。再利用 Local Binary Pattern(LBP) Learning Network 經由 Unet++架構生成預測區域修補結構，並透過門控卷積(Gated Convolution)學習圖像及空間資訊，搭配 Spatial Attention 機制，最後利用 Coarse-to-Fine 方法進行 Image Inpainting Network 修補，產生 mask 區域之修補重繪結果。與本專題的研發成果相比，Photoshop 及原論文的預測結果皆較無法準確修補應有結構，且後者修補結果有明顯的色差。此外，在 SSIM 和 PSNR 兩項指標上，本專題的修補成果與原論文相比，分別提升 2.4%及 14.6%，達到 0.9848 與 38.82。

關鍵詞：圖像繪製、局部二值模式、空間注意力、Unet++、門控卷積、深度學習

1. 緒論

老照片的修復[1]起源於十九世紀，為伴隨傳統攝影誕生的一種暗房技術[2][3]，主要針對底片與照片的修復整理以去除瑕疵，不過不同的照片損壞也需要用不同的方法及化學藥劑進行修復，而且只能修復一次，所以老照片的修復成果品質將取決於暗房技師的水平。

然而，因為科技的日新月異，我們開始希望有些東西能亙古留存，例如照片。倘若有技術能將那些破損或沾有污漬的照片進行處理，修復相片或是移除不需要的地方，使其修復回接近原本的模樣，也許就能實現此想法。

照片修復的方法主要區分為三種：暗房修復技術[2]、PS 修圖技術[3]以及 AI 照片修復：暗房修復技術是利用不同化學藥劑對不同損失程度的照片進行修復，具有不可回溯性，因此不容許失敗。PS 修圖技術會將相片數位化，並利用 Photoshop 內各種小工具進行修復，雖具有可回溯性，但復原成果需要取決於個人技術好壞。AI 照片修復則是將 mask 覆蓋在想修復的區域，會自動預測生成覆蓋區域原本的樣子，使用者不需要過多的技術即可完成修復動作。

現有的多數模型，大多僅針對於已知區域以及生成區域的關聯性，但這往往導致生成的產物不夠精緻、不夠協調，因此，我們希望利用由兩個 networks 所組成的 end-to-end, coarse-to-fine deep generative image inpainting model，此模型不僅針對已知區域以及生成區域的關聯性去做訓練，也考慮到了生成區域內部的關聯性，以使生成區域更注意細部細節，生成較精緻協調的照片。圖 1 為街景圖像修復之範例圖，左邊為有 mask 區域的照片，右邊則為修復完的照片。

圖 1 街景圖像修復範例圖

2. 相關文獻

2.1 Local Binary Pattern (LBP)

LBP 最初由 T. Ojala 等人[4]提出，為一個簡單且相當有效的紋理描述方法，其特徵的提取過程是透過對圖像的空間領域進行閾值化來重新標記每個像素，如圖 2 所示，設一個 3*3 的區域，I 為最中心的像素，其他像素的位置以 $I_1, I_2, \ldots, I_8$ 表示。依下列公式進行轉換：

$$b_i = \begin{cases} 0 & if \quad I_i \le I \\ 1 & otherwise \end{cases}, for \quad i = 1, 2, \ldots, 8$$

而後對除中心像素外的其他像素，依照一樣的順序規則排列(即順時針與逆時針皆可)，可得到 8 位元長的二進制字串 $b = b_1, b_2, \ldots, b_8$，再將其轉換為十進位制後做為中心像素 X 的特徵值。

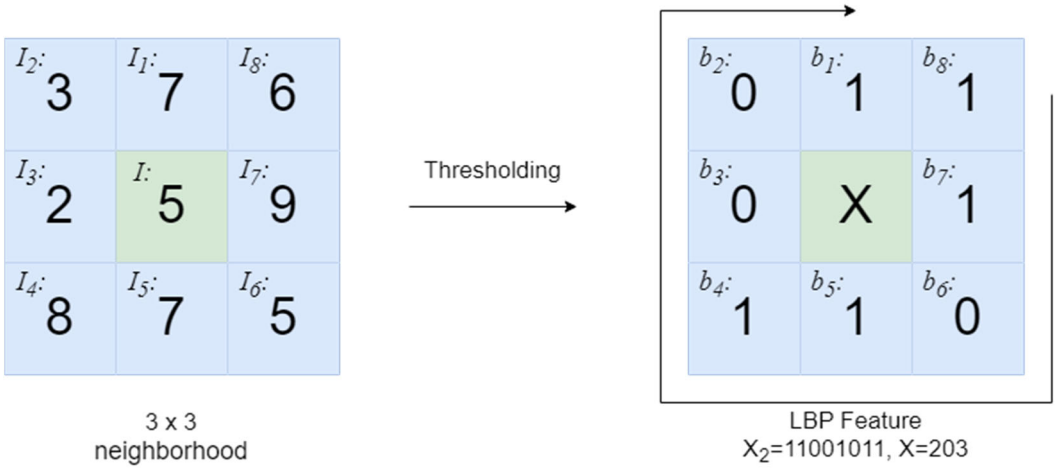


圖 2 LBP 轉換範例圖

LBP 在本質上紀錄了一個像素塊內的相對排序，可以捕捉到邊緣、斑點和其他局部結構的信息，所以能在許多視覺任務上表現出相當好的性能，比起其他方法(例如：邊緣結構信息及輪廓結構信息)，LBP 不僅含有更豐富的結構信息，其特徵提取的複雜度與參數量也較低，因此，我們認為使用 LBP 的重建結果將會帶來較好的效果。

2.2 Spatial Attention Layer

Spatial Attention Layer 由 H. Wu 等人[5]提出，目的是為了增進語意協調性，使其注意已知區域及 mask 區域之間的協調性，也可注意 mask 區域內部的協調性。Spatial Attention Layer 的特徵圖大小為 32×32，從特徵圖中的生成區域以及 mask 區域分別提取所有 1x1 的 patch，並對所有在 mask 區域的 patch 計算其與其他 patch 的餘弦相似度，再透過最相似的幾個 patch 利用非局部均值 [6] 策略去做更新，可以更好地針對 mask 區域內部的關聯性。

為了更詳細的解釋，我們將使用人臉作為範例，如圖 3 所示，其中分別將 Ω 和 $\bar{\Omega}$ 表示為此特徵圖的 mask 區域和已知區域，並且從這兩個區域內提出所有 1x1

的 patch 分別為 P 和 $\bar{\mathrm{P}}$，其定義如下：

$$\mathcal{P} = \{\mathrm{P}_j \mid \mathrm{P}_j \in \Omega\},$$
$$\bar{\mathcal{P}} = \{\bar{\mathrm{P}}_k \mid \bar{\mathrm{P}}_k \in \bar{\Omega}\}.$$

對於每個 patch $\mathrm{P}_j \in \mathrm{P}$，其在 P 內的餘弦相似度和與 $\bar{\mathrm{P}}$ 的餘弦相似度分別計算為：

$$S_{j,k} = < \frac{\mathrm{P}_j}{\|\mathrm{P}_j\|}, \frac{\mathrm{P}_k}{\|\mathrm{P}_k\|} >, \mathrm{P}_k \in \mathcal{P},$$
$$\bar{S}_{j,k} = < \frac{\mathrm{P}_j}{\|\mathrm{P}_j\|}, \frac{\bar{\mathrm{P}}_k}{\|\mathrm{P}_k\|} >, \bar{\mathrm{P}}_k \in \bar{\mathcal{P}}.$$

計算所有 $S_{j,k}$ 及 $\bar{S}_{j,k}$ 後，分別對 Ω 和 $\bar{\Omega}$ 比較 top-T 相似塊，令 $\mathcal{N} = \{n_1, ..., n_T\}$ 和 $\bar{\mathcal{N}} = \{\bar{n}_1, ..., \bar{n}_T\}$ 分別記錄 Ω 和 $\bar{\Omega}$ 中這些 top-T 相似塊的索引，然後通過非局部均值[6]策略更新每個$P_j \in P$：

$$P_j^* = \sum_{k \in \mathcal{N}} \frac{\exp(S_{j,k})}{Z_j} P_k + \sum_{k \in \mathcal{N}} \frac{\exp(\bar{S}_{j,k})}{Z_j} \bar{P}_k,$$

其中 Z_j 是正規化因子：

$$Z_j = \sum_{k \in \mathcal{N}} \exp(S_{j,k}) + \sum_{k \in \mathcal{N}} \exp(\bar{S}_{j,k})$$

假設 T=2，圖 3 中的特徵塊 P_j 將對應於像素域中的左眼，P_{n_1} 和 P_{n_2} 是 Ω 中與 P_j 相似度最高的 patch，而 $\bar{\mathrm{P}}_{\bar{n}_1}$ 和 $\bar{\mathrm{P}}_{\bar{n}_2}$ 是 $\bar{\Omega}$ 中最相似的 patch。當 mask 區域包含在注意範圍內時，我們可以找到最相關的 patch P_{n_1}。P_{n_1} 很可能對應於像素域中的右眼，但如果注意範圍僅限於已知區域，則對於此特徵圖來說最相似的 patch P_{n_1} 將被無視。因此，Spatial Attention Layer 不僅只比較了 mask 區域與已知區域的協調性，還比較了 mask 區域內部的協調性。

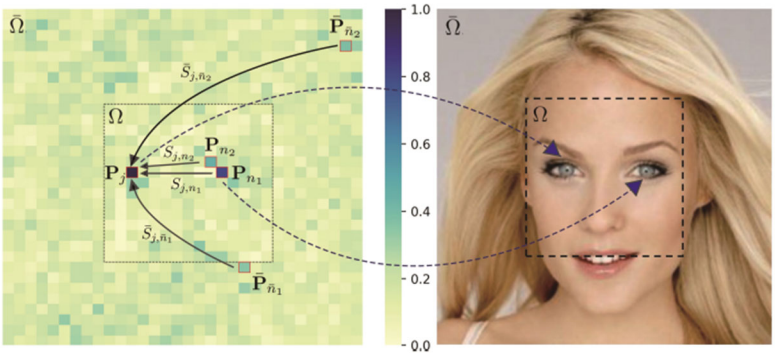


圖 3 Spatial Attention Layer 人臉範例圖

2.3 Gated Convolution

Gated Convoultion 由 J. Yu 等人[7]提出，其方法為使用 CNN 卷積和 Sigmoid 函數去學習區分有效像素及無效像素，公式如下所示：

$$Gating_{y,x} = \sum\sum W_g \cdot I$$
$$Feature_{y,x} = \sum\sum W_f \cdot I$$
$$O_{y,x} = \phi\left(Feature_{y,x}\right) \odot \sigma(Gating_{y,x})$$

其中ϕ為 Activation function(例如 Relu, LeakyRelu)，σ為 Sigmoid 函數，$\odot$為 element-wise 乘法，而W_g和W_f是兩個不同的 convolution filter。

以更簡單的方式來說，假設原本 X 輸入進一層為 64 個 Channel 的 Layer，透過圖 3 的方式會變成 X 輸入進一層為 64 X 2 個 Channel 的 Layer，其中 64 個 Channel 負責原本 CNN 的事情（學習原本圖像資訊），而另外的 64 個 Channel 也是經過 CNN ，但是會再使用 Sigmoid 函數，用意是希望可以學會其相對應 Channel 的 Mask（學習空間資訊），當經過 Sigmoid 函数後，所有值會落入 0 到 1 之間，這可以表示每個局部區域的重要性，藉此得知哪些區塊是 mask 區域，最後將兩個 64 Channel 的 Layer 使用 element-wise 相乘，另外也可使用 guidance 來引導圖像繪製，藉此達到 Gated Convolution 所希望傳達的概念。

3. 系統架構與實作方法

本系統架構參考自 H. Wu 等人[5]所提出的論文，由兩個類神經網路所組成，其中分為 LBP 學習網路(Local Binary Pattern Learning Network)及圖像繪製網路(Image Inpainting Network)。如圖 4 所示，一開始輸入白色像素填滿 mask 區域的照片I_i以及 Mask M，而I_i利用 LBP 特徵萃取轉換成 LBP 結構圖L_i，L_i以及 M 將作為 LBP 學習網路的輸入，此網路的輸出為預測 mask 區域的 LBP 結構圖L_o，其目的是引導下階段圖像繪製網路的生成，不僅可使訓練更快收斂，也可使細節更細緻。接下來將I_i、L_o及 M 作為圖像繪製網路的輸入，並利用 Spatial Attention Layer 來增進語意協調性，最後再產生修補重繪結果。

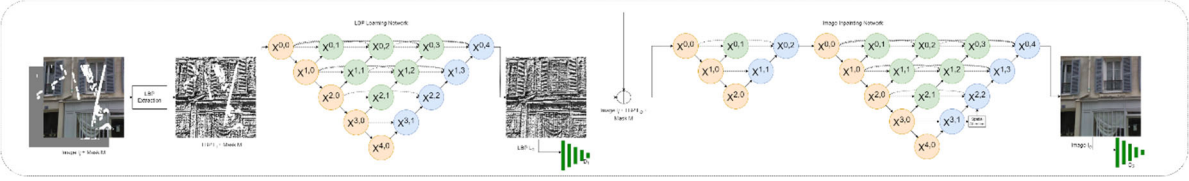


圖 4 街景圖像修復系統整體流程圖

3.1 局部二值模型學習網路(LBP Learning Network)

LBP Learning Network 的輸入為 1 channel 原圖轉換成的 LBP 結構圖及 1 channel 的 mask，而輸出為 1 channel 的預測 LBP 結構圖。此網路由 generator G_1 和 discriminator D_1 組成。我們的 G_1 採用由 encoder 和 decoder 組成的四層 U-Net++架構[8]，其中每層 encoder 由 3 x 3 kernel size 的 Gated Convolution、Group Norm[9]以及 α 參數設為 0.2 的 LeakyReLU 所組成，而 decoder 的部分將 Gated Convolution 及 LeakyReLU 替換成 Gated DeConvolution 及 ReLU，另外，為了更穩定地訓練，我們在 Group Norm 添加了權重標準化演算法(WS Algorithm)[10]，且對 Gated Convolution 及 Gated DeConvolution 做頻譜正規化(Spectral Normalization)，而 D_1 則是採用 PatchGAN 架構[11]。

針對損失函數的部分，我們採用了 H. Wu 等人[5]所提出的 multi-level loss 來處理模型每一層的特徵域偏差，可以更真實地預測 LBP 特徵，提高修復效果，另外也使用了計算均方差的 reconstruction loss 及 adversarial loss[12]。multi-level loss function 定義如下：

$$\mathcal{L}_m = \sum_{h \epsilon \mathcal{H}} \left\| \Phi_h(L_o) - \Phi_h(L_g) \right\|_2$$

其中 L_o 是 LBP 的輸出、L_g 是 LBP 的 ground truth、$\Phi_h(L_g)$ 是對應的高層次特徵、h 是 G_1 階層索引值，而 $\mathcal{H}$ 表示了 G_1 中所有 Convolution 和 DeConvolution 層的 Index。reconstruction loss 定義如下：

$$\mathcal{L}_r = \left\| L_o - L_g \right\|_2$$

而 adversarial loss 定義如下：

$$\mathcal{L}_a = \min_{\boldsymbol{G_1}} \max_{D_1} \mathbb{E}_{L_g}\left[logD_1(L_g)\right] + \mathbb{E}_{L_i}[\log(1 - D_1(G_1(L_i, M)))]$$

最後，LBP learning network 的損失函數將由以上提到的三種公式組成，其損失函數定義如下：

$$\mathcal{L}_{LBP} = \lambda_m \mathcal{L}_m + \lambda_r \mathcal{L}_r + \lambda_a \mathcal{L}_a$$

其中 λ_m、λ_r 和 λ_a 是權衡不同類型損失的參數。

3.2 圖像修復網路(Image inpainting Network)

此架構與 LBP 學習網路相似，但輸入將改成 3 channel 的原始照片、由 LBP 學習網路輸出的 1 channel 預測 LBP 結構和 1 channel 的 mask，另外在 decoder 的使用了 Spatial Attention Layer，最後輸出為 3 channel 的生成照片。此網路另外利用 Coarse-to-Fine 方法，如圖 5 所示，將由兩層 U-net++的 Coarse Network 及四層 U-net++的 Fine Network 組成，先生成較粗糙的結果，再生成較細緻的結果，目的是為了修補地更精緻，從圖 6 將可以看到兩階段的整個架構。

損失函數除了在 LBP 學習網路提到的三種，也額外加了 perceptual loss[13]和 style loss[14]來得知 I_o 與 I_g 是否相似及測量 activation maps 中 covariances 的差異。perceptual loss 定義如下：

$$\mathcal{L}_p = \sum_{h \epsilon \mathcal{A}} \left\| \varphi_h(I_o) - \varphi_h(I_g) \right\|_2$$

其中 φ_h 對應到 VGG-16 network 的第 h 層特徵，集合 $\mathcal{A}$ 是一種 con2_1,con3_1,con4_1 的索引形式。style loss 定義如下：

$$\mathcal{L}_s = \sum_{h \epsilon \mathcal{A}} \left\| \mathrm{G}^{\varphi_h}(I_o) - \mathrm{G}^{\varphi_h}(I_g) \right\|_2$$

其中 G^{φ_h} 是從 φ_h 建構出的 3×3 Gram 矩陣。

對於整個 Image Inpainting Network 的損失函數，我們也加入 3.1 小節提到的 multi-level loss、reconstruction loss 以及 adversarial loss，最終的損失函數定義如下：

$$\mathcal{L}_{Img} = \lambda_m \mathcal{L}_m + \lambda_r \mathcal{L}_r + \lambda_a \mathcal{L}_a + \lambda_p \mathcal{L}_p + \lambda_s \mathcal{L}_s$$

其中 λ_m、λ_r、λ_p 、λ_s 和 λ_a 是權衡不同類型損失的參數。

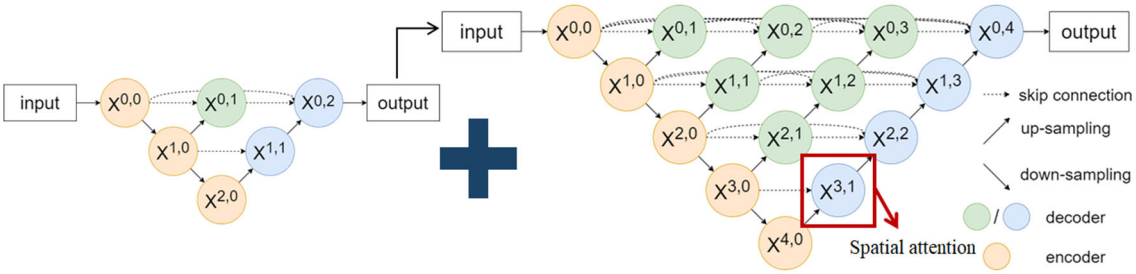


圖 5 Coarse-to-Fine 方法概念圖

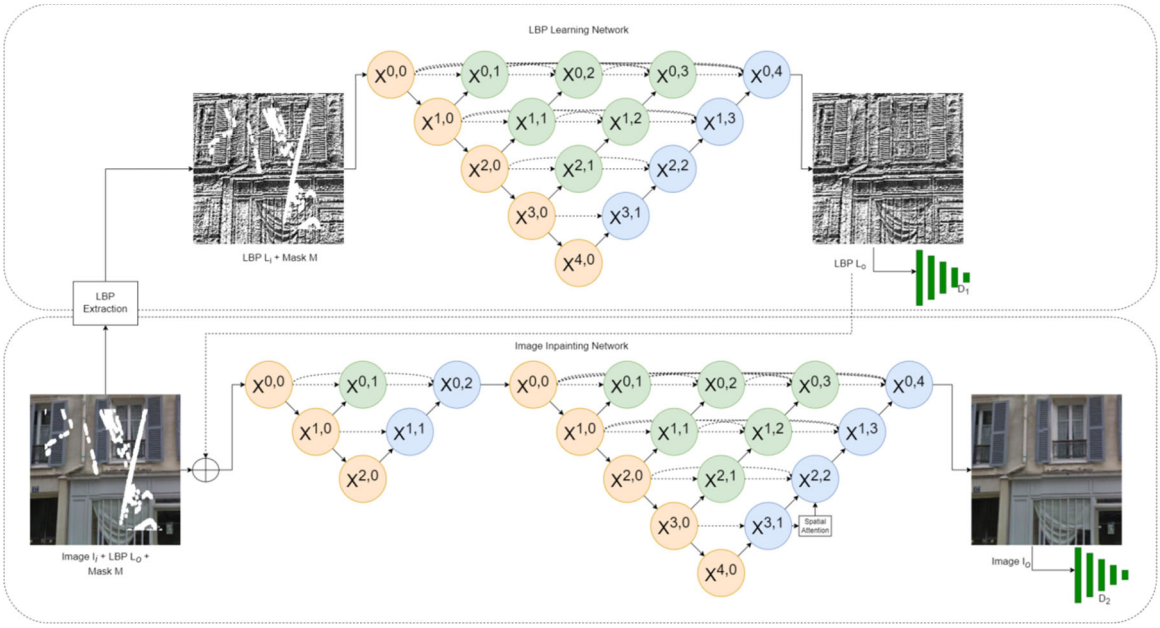


圖 6 街景圖像修復系統整體架構圖

4. 實驗結果與討論

4.1 實驗環境設定

如表 1 所示，此模型利用 PyTorch 框架實現，在環境設定中，我們使用配備了 Core-i5 以及一個 NVIDIA GeForce RTX 3060 的桌機進行訓練，並搭載了 PyTorch 版本 1.11.0+cu113 及 Python 版本 3.9.12。此外，我們利用 Paris Street-view[15]資料集去做訓練，此資料集包含 14900 張訓練照片以及 100 張測試照片，其中，我們以 9:1 的比例從訓練照片中區分訓練集及測試集，每次訓練時間大約為三天。

表 1 實驗環境設定

類別	詳細內容	
Python 版本	3.9.12	
Pytorch 版本	1.11.0	
Cuda 版本	11.3	
硬體設備	Core-i5 + NVIDIA GeForce RTX 3060	
訓練時間	3 天	
資料集	Paris Street-View Dataset	
資料集數量	訓練集數量	13410 張
	驗證集數量	1490 張
	測試集數量	100 張

4.2 實驗成果

本專題實驗成果利用數量為 100 張的 Paris Street-View Dataset 測試集做測試，專題前期，我們對 H. Wu 等人[5]所提出的論文(方法 A)做了不同參數的訓練，詳細內容如表 2 所示。每個訓練的時間大約為兩天，並對測試照片做測試。而專題中期，我們希望能有更穩定有效的訓練，所以思考出如表 3 所示的架構(方法 B)，由於訓練的 batch_size 是 1，但是從論文[9]可以發現 IN 的訓練效果並不是那麼好，所以我們使用了論文[9]和[10]所提 GN(group normalization)及 WS(weight standardization)，並使用了專題前期訓練最好的參數來做表 3 架構的訓練，所有訓練方法及參數可以從表 4 看到，而對每個訓練方法做測試的結果數據統計在表 5。

表 2 方法 A 所使用架構

層數	LBP 學習網路
Layer1	Conv(64,4,2,1)
Layer2	LReLU; Conv(128,4,2,1); IN;
Layer3	LReLU; Conv(256,4,2,1); IN;
Layer4	LReLU; Conv(512,4,2,1); IN;
Layer5	LReLU; Conv(512,4,2,1); IN;
Layer6	LReLU; Conv(512,4,2,1); IN;
Layer7	LReLU; Conv(512,4,2,1); IN;
Layer8	LReLU; Conv(512,4,2,1); ReLU; DeCONV(512,4,2,1); IN;
Layer9	Cat(Layer 8, Layer 7); ReLU; DeCONV(512,4,2,1); IN;
Layer10	Cat(Layer 9, Layer 6); ReLU; DeCONV(512,4,2,1); IN;
Layer11	Cat(Layer 10, Layer 5); ReLU; DeCONV(512,4,2,1); IN;
Layer12	Cat(Layer 11, Layer 4); ReLU; DeCONV(512,4,2,1); IN;
Layer13	Cat(Layer 12, Layer 3); ReLU; DeCONV(256,4,2,1); IN;
Layer14	Cat(Layer 13, Layer 2); ReLU; DeCONV(128,4,2,1); IN;
Layer15	Cat(Layer 14, Layer 1); ReLU; DeCONV(64,4,2,1); Tanh;
層數	圖像繪製網路
Layer1	Conv(64,4,2,1)
Layer2	LReLU; Conv(128,4,2,1); IN;
Layer3	LReLU; Conv(256,4,2,1); IN;
Layer4	LReLU; Conv(512,4,2,1); IN;
Layer5	LReLU; Conv(512,4,2,1); IN;
Layer6	LReLU; Conv(512,4,2,1); IN;
Layer7	LReLU; Conv(512,4,2,1); IN;
Layer8	LReLU; Conv(512,4,2,1); ReLU; DeCONV(512,4,2,1); IN;
Layer9	Cat(Layer 8, Layer 7); ReLU; DeCONV(512,4,2,1); IN;
Layer10	Cat(Layer 9, Layer 6); ReLU; DeCONV(512,4,2,1); IN;
Layer11	Cat(Layer 10, Layer 5); ReLU; DeCONV(512,4,2,1); IN;
Layer12	Cat(Layer 11, Layer 4); ReLU; DeCONV(512,4,2,1); IN;
Layer13	Cat(Layer 12, Layer 3); (SpatialAttention); ReLU; DeCONV(256,4,2,1); IN;
Layer14	Cat(Layer 13, Layer 2); ReLU; DeCONV(128,4,2,1); IN;
Layer15	Cat(Layer 14, Layer 1); ReLU; DeCONV(64,4,2,1); Tanh;

表 3 方法 B 所使用架構

層數	LBP 學習網路
Layer1	Conv(64,4,2,1)
Layer2	LReLU; SN(Conv(128,4,2,1)); GN;
Layer3	LReLU; SN(Conv(256,4,2,1)); GN;
Layer4	LReLU; SN(Conv(512,4,2,1)); GN;
Layer5	LReLU; SN(Conv(512,4,2,1)); GN;
Layer6	LReLU; SN(Conv(512,4,2,1)); GN;
Layer7	LReLU; SN(Conv(512,4,2,1)); GN;
Layer8	LReLU; SN(Conv(512,4,2,1)); ReLU; SN(DeCONV(512,4,2,1)); GN;
Layer9	Cat(Layer 8, Layer 7); ReLU; SN(DeCONV(512,4,2,1)); GN;
Layer10	Cat(Layer 9, Layer 6); ReLU; SN(DeCONV(512,4,2,1)); GN;
Layer11	Cat(Layer 10, Layer 5); ReLU; SN(DeCONV(512,4,2,1)); GN;
Layer12	Cat(Layer 11, Layer 4); ReLU; SN(DeCONV(512,4,2,1)); GN;
Layer13	Cat(Layer 12, Layer 3); ReLU; SN(DeCONV(256,4,2,1)); GN;
Layer14	Cat(Layer 13, Layer 2); ReLU; SN(DeCONV(128,4,2,1)); GN;
Layer15	Cat(Layer 14, Layer 1); ReLU; SN(DeCONV(64,4,2,1)); Tanh;
層數	圖像繪製網路
Layer1	Conv(64,4,2,1)
Layer2	LReLU; SN(Conv(128,4,2,1)); GN;
Layer3	LReLU; SN(Conv(256,4,2,1)); GN;
Layer4	LReLU; SN(Conv(512,4,2,1)); GN;
Layer5	LReLU; SN(Conv(512,4,2,1)); GN;
Layer6	LReLU; SN(Conv(512,4,2,1)); GN;
Layer7	LReLU; SN(Conv(512,4,2,1)); GN;
Layer8	LReLU; SN(Conv(512,4,2,1)); ReLU; SN(DeCONV(512,4,2,1)); GN;
Layer9	Cat(Layer 8, Layer 7); ReLU; SN(DeCONV(512,4,2,1)); GN;
Layer10	Cat(Layer 9, Layer 6); ReLU; SN(DeCONV(512,4,2,1)); GN;
Layer11	Cat(Layer 10, Layer 5); ReLU; SN(DeCONV(512,4,2,1)); GN;
Layer12	Cat(Layer 11, Layer 4); ReLU; SN(DeCONV(512,4,2,1)); GN;
Layer13	Cat(Layer 12, Layer 3); (SpatialAttention); ReLU; SN(DeCONV(256,4,2,1)); GN;
Layer14	Cat(Layer 13, Layer 2); ReLU; SN(DeCONV(128,4,2,1)); GN;
Layer15	Cat(Layer 14, Layer 1); ReLU; SN(DeCONV(64,4,2,1)); Tanh;

表 4 訓練參數表

參數	訓練一	訓練二	訓練三	訓練四	訓練五
方法	A	A	A	A	B
優化器	Adam	Adam	Adamax	Adadelta	Adamax
初始學習率	0.0002	0.001	0.001	0.001	0.001
學習率遞減	無	Cosine	Cosine	無	Cosine
說明	參考論文[5]			loss 一開始較高，因此先不使用學習率遞減	

表 5 測試數據結果表

LOSS 與評估指標	LOSS	SSIM	PSNR	MAE
訓練一	1.1035	0.9717	36.19	0.0086
訓練二	1.0403	0.9727	36.42	0.0084
訓練三	0.9553	0.9741	36.66	0.0081
訓練四	訓練階段的數值明顯比其他三個訓練差很多，因此不另做測試			
訓練五	1.0578	0.9725	36.44	0.0084

表 4 中訓練一的參數皆依照參考論文[5]，而訓練四一開始的損失值較高，因此不使用學習率遞減，以免拖慢訓練時間。輸出數據如表 5 所示，訓練四的損失值最後並沒有收斂，因此不另測試。原本預期的最佳數據是訓練五，但可以從結果看到訓練五略遜於訓練二及訓練三，不過誤差都在可接受範圍內，所以之後的訓練架構依然會參考訓練五(即表 3 所示之架構)。

由以下的圖 7、圖 8 及圖 9 可以發現每項訓練的圖片輸出結果，而我們發現到大部分的照片容易有色差和邊界較明顯以及修補較不完整的情況，因此，我們尋找了三種方法去改善此結果，首先利用 Gated Convolution 改善色差及邊界明顯的情況，選擇此方法的原因是它支援單 channel 的 guidance 引導繪製，可以使我們將 LBP 結構圖當成輸入來更好地引導輸出，而後我們還利用了 U-net++來讓結構可以更好的被學習到，雖然會使訓練時間及等待修復時間變長，但它可以更好地保留到該有的結構性，最後再利用 Coarse-to-Fine 方法讓修補更加細緻。

圖 7 輸出結果範例-1

圖 8 輸出結果範例-2

圖 9 輸出結果範例-3

最後透過 Gated Convolution、Unet++及 Coarse-to-Fine 方法的實作，我們也做了消融實驗來觀察改善的結果，如表 6 所示，除了損失值(LOSS)及平均絕對誤差(MAE)，我們還使用結構相似度(SSIM)及峰值訊噪比(PSNR)做為評估指標，其中第一行數據是使用論文[5]的模型參數得到的測試結果；第二行使用 Photoshop 中內容感知填色方法的測試結果，其方法為選取 mask 區域，再對整張照片做內容感知再修復，並不做任何人工修復；第三行是只有使用 Gated Convolution(GC)之結果；第四行使用了 Gated Convolution(GC)及 U-net++(UN)；第五行則是使用了 Gated Convolution(GC)、U-net++(UN)及 Coarse to Fine 方法(C2F)。最後可以從第五行的數據看出明顯的進步，對於第一行之數據，我們讓 LOSS 及 MAE 分別下降了 74.9%及 50.0%，而評估指標 SSIM 及 PSNR 也上升 2.4%及 14.6%。

表 6 消融實驗數據表

	LOSS	SSIM	PSNR	MAE
參考論文[6]	6.4257	0.9608	33.86	0.0118
Photoshop		0.9742	35.78	0.0086
GC	0.8366	0.9781	36.90	0.0077
GC+UN	0.7036	0.9829	38.21	0.0063
GC+UN+C2F	**0.6243**	**0.9848**	**38.82**	**0.0059**
提升▲/下降▼	▼74.9%	▲2.4%	▲14.6%	▼50.0%

圖 10 及圖 11 為分別使用 CNN Convolution 及 Gated Convolution 的輸出照片結果比較，可以很明顯地從使用 CNN Convolution 照片中修補好的 mask 區域看出色差及邊界，而使用 Gated Convolution 的修補結果則比較接近 Ground Truth，由此得知使用 Gated Convolution 將達到更好的效果。

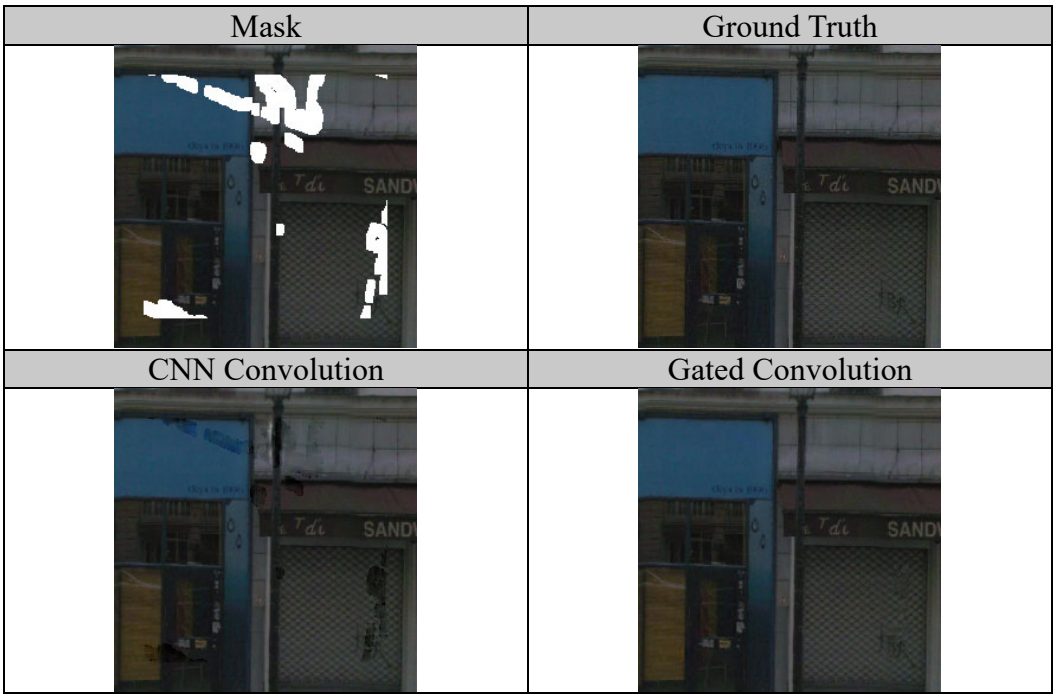


圖 10 使用不同卷積之輸出照片結果範例比較-1

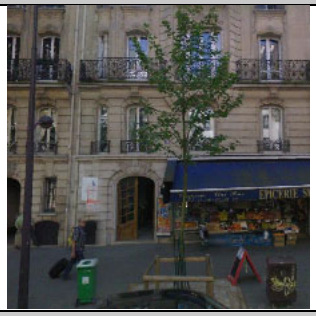
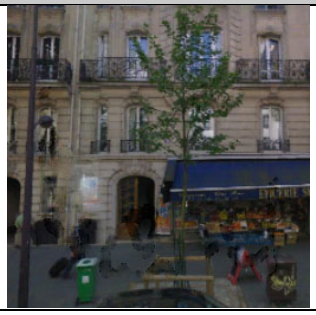
Mask	Ground Truth
	
CNN Convolution	Gated Convolution
	

圖 11 使用不同卷積之輸出照片結果範例比較-2

圖 12 及圖 13 為使用 Photoshop 內容感知填色方法、使用 U-net 架構的論文[5]照片輸出結果、使用 Gated Convolution(GC)和 U-net++(UN)的輸出結果及使用 Gated Convolution(GC)、U-net++(UN)和 Coarse to Fine 方法(C2F)的輸出結果，其中，從使用 Photoshop 內容感知填色方法和只使用 U-net 的結果中發現樹幹不見及窗戶框線不全，修補較不完整，而使用 Gated Convolution 和 U-net++後，可以看出樹幹及窗戶框線些許輪廓，最後再加上 Coarse to Fine 方法，會發現樹幹更粗更明顯，甚至連旁邊的台階也較清楚，而窗戶框線也較完整，因此，使用 Gated Convolution、U-net++和 Coarse to Fine 方法的輸出結果較接近 Ground Truth。

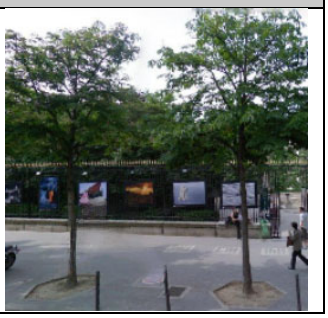
Mask	Ground Truth	Photoshop 內容感知填色方法
		
論文[5]架構(U-net)	GC+UN	GC+UN+C2F
		

圖 12 消融實驗輸出照片結果範例比較-1

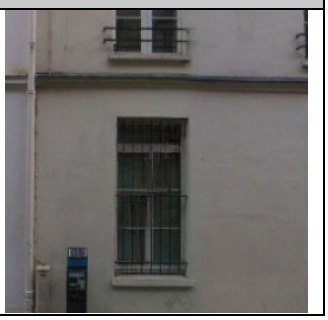
Mask	Ground Truth	Photoshop 內容感知填色方法
		
論文[5]架構(U-net)	GC+UN	GC+UN+C2F
		

圖 13 消融實驗輸出照片結果範例比較-2

5. 結論與未來展望

本專題利用以下三種技術改進參考論文：1) 利用門控卷積讓色差及邊界較不明顯; 2) 利用 Unet++提升修補完整性; 3) 利用 Coarse to Fine 使修補完整性更加細緻。與原論文相比，透過以上三種改善後，在 LOSS 和 MAE 數值上分別降低 74.9%以及 50.0%，達到 0.6243 及 0.0059，並且在 SSIM 與 PSNR 數值上也個別提升 2.4%和 14.6%，達到 0.9848 及 38.82。不僅如此，我們的專題效果比商用 Photoshop 內容感知填色修復更加有效，在結構性上修補得更好。

由於本專題目前只能修復於街景圖像，期望未來能增加訓練集，使其能擴充至更多種類的相片，並且正式上線，增加更多使用者。除此之外，也希望此專題可應用在商業攝影上。在移除物件方面，可利用較少時間得到更高效益及更好的成果。另外，若是修復損壞相片時，也只需掃描為電子檔後，進行線上修補，如此便可永久保存，不必再擔心損毀。

6. 誌謝

首先感謝指導老師林朝興教授在這一年內的教導，從學習基本神經網路知識開始，接著文獻研讀、選定題目，再到後來的正式製作以及最終準備專題展，朝興教授總是不厭其煩地陪我們走過製作專題的每個關鍵階段，衷心感謝朝興教授給予我們鼓勵、稱讚、及支持，讓我們數度獲得從失敗中站起來的勇氣，也才開始對於自己的能力有所自信，順利完成畢業專題系統研究與實作開發。

7. 參考文獻

[1] 清风拾影 . (n.d.). 老照片修复 . Retrieved September 12, 2020, from https://zhuanlan.zhihu.com/p/235421351

[2] 神奇的村林孝夫舊照片化學修復法 - 每日頭條 - KKNEWS.CC. (n.d.). Retrieved June 3, 2015, from https://kknews.cc/zh-tw/culture/jvnbjey.html

[3] 數字時代下的手工技藝,老照片修復技術還能不能傳承? 海納網. (n.d.). Retrieved May 14, 2021, from https://hainve.com/art/30841.html

[4] T. Ojala, M. Pietikainen, and D. Harwood, “A comparative study of ¨ texture measures with classification based on featured distributions,” Pattern Recogn., vol. 29, no. 1, pp. 51–59, 1996.

[5] H. Wu, J. Zhou and Y. Li, "Deep Generative Model for Image Inpainting With Local Binary Pattern Learning and Spatial Attention," in *IEEE Transactions on Multimedia*, vol. 24, pp. 4016-4027, 2022.

[6] A. Buades, B. Coll, and J. M. Morel, “A non-local algorithm for image denoising,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., vol. 2. IEEE, 2005, pp. 60–65.

[7] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu and T. Huang, "Free-Form Image Inpainting With Gated Convolution," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 4470-4479, doi: 10.1109/ICCV.2019.00457.

[8] Zhou et al., UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation. DOI: 10.1109/TMI.2019.2959609

[9] S. Qiao, H. Wang, C. Liu, W. Shen, and A. Yuille, “Micro-batch training with batch-channel normalization and weight standardization,” arXiv preprint arXiv:1903.10520, 2019. 10.48550/arXiv.1903.10520

[10] S. Qiao, H. Wang, C. Liu, W. Shen, and A. Yuille, “Micro-batch training with batch-channel normalization and weight standardization,” arXiv preprint arXiv:1903.10520, 2019. 10.48550/arXiv.1903.10520

[11] P. Isola, J. Y. Zhu, T. H. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recogn., 2017, pp. 1125–1134.

[12] Z. Y. Yan, X. M. Li, M. Li, W. M. Zuo, and S. G. Shan, “Shift-net: image inpainting via deep feature rearrangement,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 1– 17.

[13] J. Johnson, A. Alahi, and F. F. Li, “Perceptual losses for real-time style transfer and super-resolution,” in Proc. Eur. Conf. Comput. Vis. Springer, 2016, pp. 694– 711.

[14] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recogn., 2016, pp. 2414–2423.

[15] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. Efros, “What makes paris look like paris?” ACM Trans. Graph., vol. 31, no. 4, 2012.