第二十二屆旺宏科學獎
創意說明書

參賽編號:SA22-096
姓名:賴昱錡
作品名稱:利用 GAN 實現歷史航空影像之上色
參賽類別:資訊科
關鍵字:GAN、歷史航照、影像上色

目錄

壹、 前言

一、 研究動機

隨著台灣科技發展的進步，網路上可以輕易取得內政部所測製或中研院所徵集的一系列歷史航空照片，藉由數位典藏及GIS圖資發佈技術，讓二戰後期迄今所記錄臺灣歷史航照影像得以完整保存及活用。無論如何，歷年來所統計的資料均對人文科學的研究有著顯著的貢獻與影響。

但在比較過去及現在的地景變遷時，黑白的照片始終會帶來分析上的不方便，且網路上許多模型（如: CNN、U-Net）對舊航照的上色效果不甚理想（問題如: 雜訊過多、鮮豔度不足，請見附錄），因此此研究的最終目的，便是希望能建立一個上色模型（運用生成對抗網路所衍伸的 pix2pix、CycleGAN 等模型實作），並對這些方法進行深入的探討與比較，期許能讓灰階歷史航照的上色達到較佳的效果，為臺灣的地理資訊研究貢獻一些。

二、研究設備與器材

（一）硬體

1. 筆記型電腦:（CPU: Intel(R) Core(TM) i5-1135G7@2.40GHz，記憶體 8GB）
2. 向中研院人社中心申請之伺服器。

（二）軟體

1. Windows 11、Ubuntu 20.04（作業系統）
2. Python 3.10（程式語言）
3. Visual Studio Code、Google Colab（程式編輯器）
4. QGIS（地理圖資軟體）
5. Pytorch（機器學習框架）
6. matplotlib、pandas、cv2（Python 套件）

（三）訓練資料來源

中央研究院臺北、台南百年歷史地圖 WMTS 服務、Google Earth、國土測繪中心製作之正射影像圖集。

貳、 研究目的

一、利用生成對抗網路所衍伸的分支模型（pix2pix、CycleGAN、CUT 等）對舊航照進行上色。

二、以 pytorch 實作 pix2pix、CycleGAN、CUT、BicycleGAN 等模型，並探討以下變因對模型成效之影響:

（一）調整生成器與判別器之權重

（二）改變 pix2pix 生成器架構（有無 U-Net）

（三）改變 pix2pix 之訓練方式

（四）調整 CUT 模型的 GAN 損失模式

（五）是否採用 Identity Loss

（六）BicycleGAN 潛在編碼的加入時機

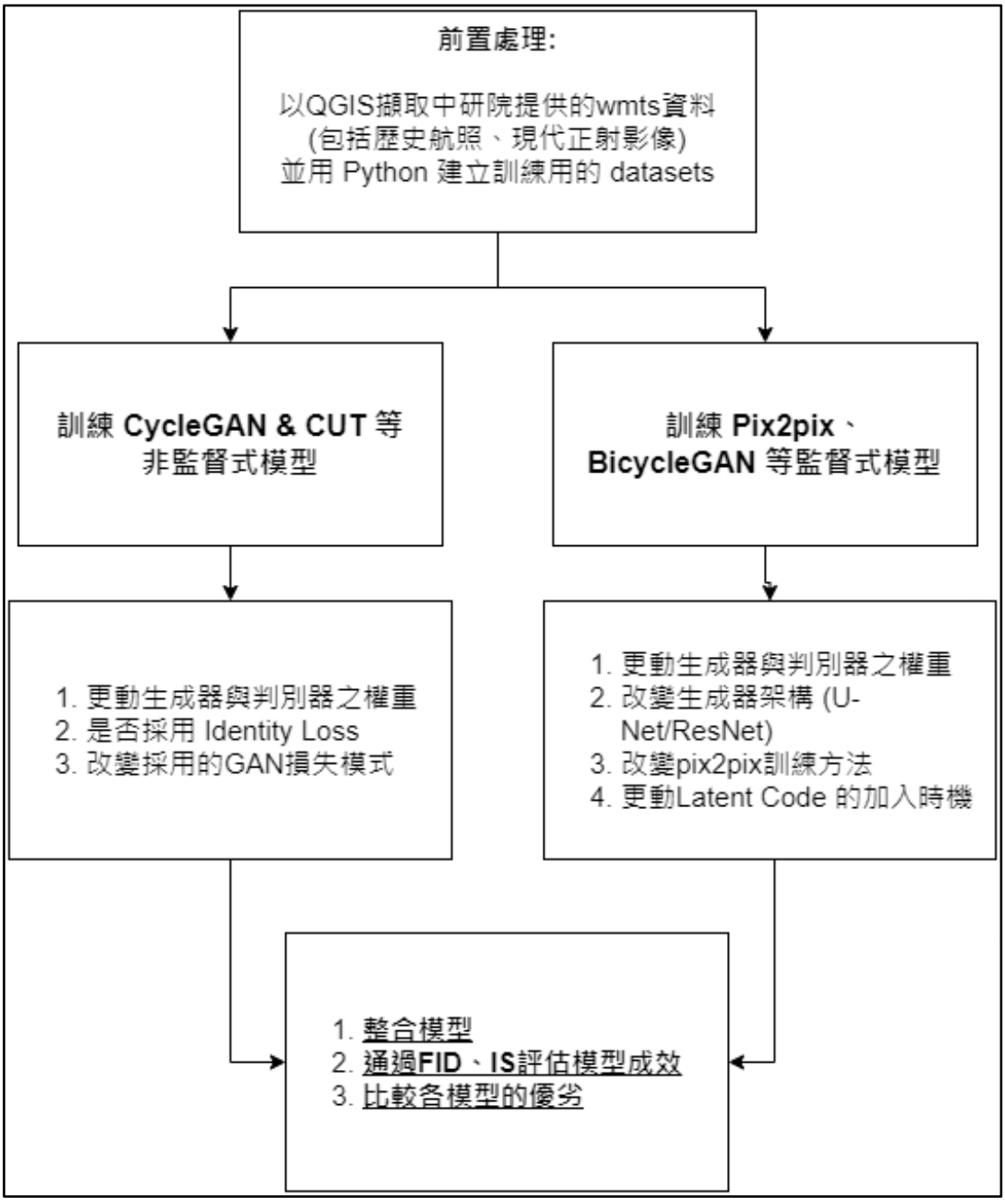
三、分析各個模型間上色成效的差異，並透過生成對抗模型的FID、IS等指標來評估生成影像的品質與多樣性。

參、 研究過程與方法

一、研究架構

研究架構大致上可分為前置處理、模型訓練、調整參數、評估成效等四個部分，更詳細的內容請參考下圖。

註: 本研究程式碼均使用 Pytorch 作為其深度學習框架實作。



圖(一) 研究架構圖

二、前置處理作業

(一)運用 QGIS 及其外掛程式 Qtiles，以及中研院所提供之 WMTS，擷取臺北、臺南地區航照。如圖，透過 ESRI shapefile 的格式界定矩形範圍內的區域，使用 Qtiles 套件抓圖（zoom 值設定為 15~16）

圖(二) 透過 QTile 套件擷取特定範圍的 shapefile

(二)將航照透過 Python 的 Pillow 套件轉為灰階，存放於另一個資料夾。

(三)透過 shutil 套件，依據各模型訓練方式配置資料集。擷取民初歷史航照的方法與前述相同。

三、影像上色模型的訓練

註: 以下實驗，均以灰階處理航照與實際彩色航照(採集自中研院 wmts 中的正射影像圖集)進行訓練，測試的部分皆以灰階處理航照與歷史航照作為模型測試的輸入，訓練其輸出上色後的影像。而 CUT、CycleGAN 因其非監督式學習的方式，不具有成對的 datasets，與舊航照作為輸入的實驗相同，都不具有 Ground truth（真實圖像)可比較。於此階段中，歷史航照皆係指 1970~1980 中研院拍攝之臺北/台南地區航照。

(一) pix2pix & BicycleGAN

在 pix2pix 與 BicycleGAN 模型的訓練過程中，我們會輸入一對圖像，其中一張圖像是輸入，另一張則是目標輸出。模型的目標是使輸出圖像盡可能接近目標輸出圖像。我們將訓練模型所使用的圖像集分成三個部分，分別為訓練集(train)、驗證集(val)和測試集(test)。

訓練集用於訓練模型，驗證集則是在訓練過程中檢驗模型正確性的工具，而測試集則是在訓練完成後用來測試模型的性能，其中包括灰階處理的現代航照及 1970~1980 拍攝之臺北、台南地區航照(原始圖像即為灰階)。

圖(三) 256*512 的 pix2pix 訓練輸入圖像

而於此實驗當中，會將轉灰階的現代航照與現代航照合成為一張 512*256 的圖片(如上圖所示)放入模型訓練，train 中共 1550 張照片，test 中共 900 張照片，val 中共 900 張照片，共訓練 200 個 epoch，而模型的學習率(Learning rate)均設為 0.0002，在 100 個 epoch 後開始下降。

在訓練過程中會記錄生成器與判別器之對抗損失、L1 Loss 等損失，再透過 Python 的 matplotlib 繪製成損失曲線，觀察其變化趨勢。

(二) CycleGAN & CUT

在 CycleGAN 與 CUT 模型的訓練過程中，我們會輸入不成對的二組 256*256 像素的圖像，一組訓練集為 A 型態，另一組則為 B 型態，模型的目標是使 A 轉 B 和 B 轉 A 的任務越完美越好，進而為黑白影像，而此研究將 A 定為未上色的灰階影像，而 B 為有顏色的原影像 (Ground truth)。

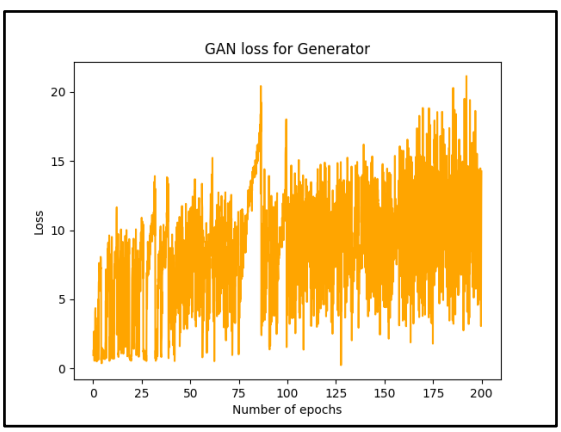
可以將資料集分為 4 個部分，分別為 trainA、trainB、testA、testB，train 資料夾用來訓練模型，各為 1550 張，而 test 資料夾各 500 張照片，用於測試、生成最終成果，其中包括灰階處理的現代航照及 1970~1980 拍攝之臺北、台南地區航照(原始圖像即為灰階)。共訓練 200 個 epoch，模型的學習率(Learning rate)均設為 0.0002，在 100 個 epoch 後開始下降。

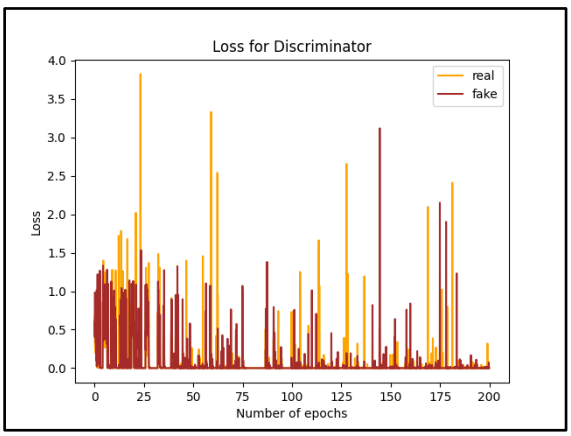
在訓練過程中除了記錄生成器與判別器之對抗損失、infoNCE Loss 等，也會記錄生成彩圖、灰階影像與原 datasets 的差異，即為 Cycle Consistency Loss，再透過 Python 的 matplotlib 繪製成損失曲線，觀察其變化。

四、探討模型細節調整對上色成效的作用

(一) 調整生成器與判別器之權重

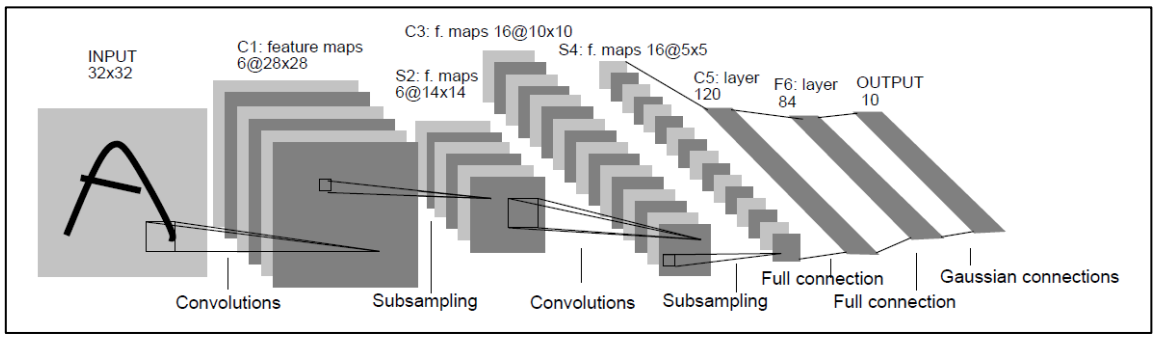
改變 Generator 和 Discriminator 上的 Filter 數量。透過改變這項因素，探討生成器與判別器之權重對損失趨勢的作用，在 GAN 訓練中 Generator 的 Adversarial Loss(對抗損失)常會出現上升的現象，而 Discriminator 的 Adversarial Loss 幾乎下降至 0 (如下圖所示)。是因為訓練到後期 Discriminator outperform Generator，尤其於 CycleGAN、pix2pix、CUT 等生成模型更容易發生。對於對抗損失上升的解決方法，主要集中於降低判別器的能力，如: 調降判別器的學習率(或升高生成器的學習率)、生成器比判別器訓練較多個 epochs、減少判別器卷積層上濾波器的數量等。





圖(四)以未經任何調整的 pix2pix 實作歷史航照上色的損失趨勢。
左為生成器對抗損失(大幅提升)、右為判別器對抗損失(下降至 0)。

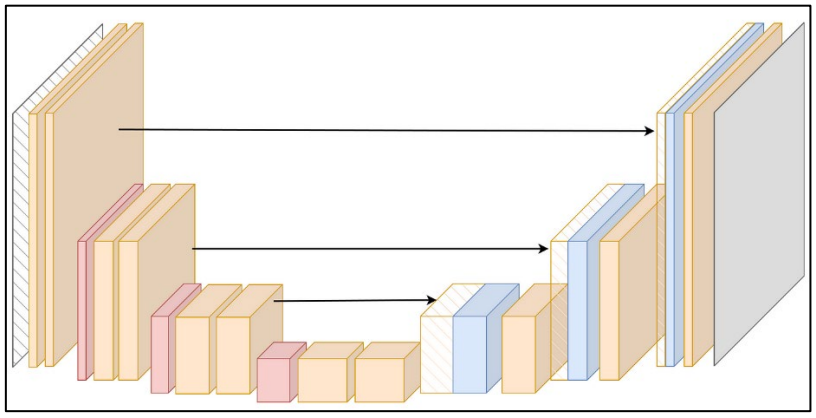
所以此研究中，考量實作上的難度&初步效果的評估後，目前更動了各模型中的 ngf（生成器末卷積層的 filter 數量)、ndf（判別器首卷積層的 filter 數量)、學習率等參數，再透過 FID、IS、對抗損失等指標來衡量這些在理論上能改變生成器、判別器權重的策略，對模型成效的影響。



圖(五) 卷積層可加入不同數量的 filter，進而影響輸出影像的特徵。

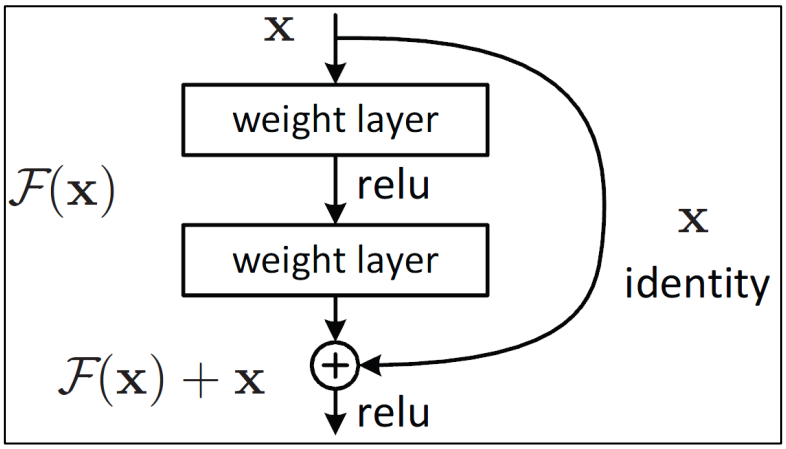
(二) 改變 pix2pix 生成器架構 (U-Net 的有無)

透過將 skip connection (跳躍連結)加入模型中之生成器，設該神經網路具有 n 層，在第 i 層和第 n-i 層之間都加入跳躍連結，且每一個連結都可以讓資訊直接流過，即為 U-Net 最大的模型架構特色，這樣萃取特徵的方式，也使得 U-Net 在許多圖像分割(Image segmentation)、圖像轉換的任務，對於多主體、細節相對多的圖片，還原的精確度頗高。此研究欲探討 U-Net 對模型成效的影響，故在這邊 pix2pix 也另外採用 ResNet 作為該模型的生成器，以所提及之 FID、IS 等評估指標試著比較兩者成效的差異。



圖(六) U-Net 之架構為在原 Autoencoder 上建立跳躍連結。

ResNet 和恆等映射(Identity mapping)有很大的不同，恆等映射的核心思想是如果映射函數為 H(x)，模型直接擬合訓練樣本 x，輸出為 H(x)。而在 ResNet 中，可以將模型的目標從 H(x)改為 F(x)=H(x)-x，模型的目標是每個殘差塊(Residual block)學到的「殘差變為 0」(等價於恆等映射 H(x)=x)，而不是直接學習恆等映射函數，架構如下圖所示。



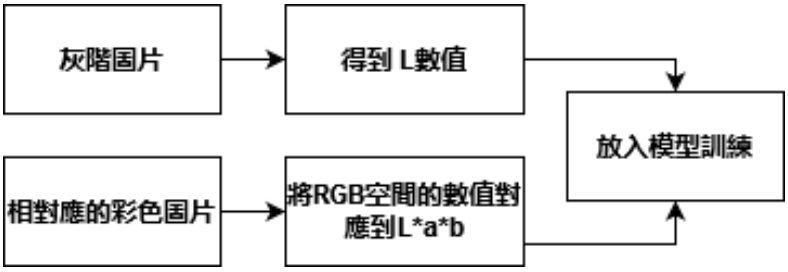
圖(七) ResNet 原理的簡介

因此，只要考量殘差和原樣本，就能推出生成的映射函數，在實現上，只需要直接將上層的輸入 x 與經過 non-linear 轉換的輸出 F(x) 相加（採用 shortcut connection），這樣不僅可以得到 F(x)+x，還不會增加額外的參數增加計算量。仍然可以使用 SGD 進行反向傳播訓練，常見的深度學習框架也可以簡單實現，在訓練深度加深時，通常可以得到較好的準確度與效率。。

(三) 調整 pix2pix 模型之訓練方式

對於 pix2pix 這種監督式學習模型而言，要達成影像上色之任務，可以由兩種方式去進行，其一為直接訓練模型針對灰階的圖像及 Ground truth 圖像之 RGB 數值進行映射，因為每張圖片是以像素為單位，而每個單位又具有三原色所對應的數值，一旦圖片的尺寸過大，便容易發生上色失準或耗費的運算資源龐大等問題，而另外一種影像上色常用的映射方式，是以 L*a*b 顏色空間的角度去處理圖片。

L*a*b 顏色空間分為 L、a*、b*三個顏色通道，其中 L 描述圖像之亮度值(即為一般的灰階圖片)，*a 和*b 兩項數值分別描述綠-紅色與藍-黃色的數值，而本研究所採用的另一種映射手法，便是由 L 通道的數值，預測*a 與*b 兩圖層之狀況，因此這種映射訓練的方式如下所示，透過模型預測所得的 L*a*b 數據，即可轉換、還原為 RGB 的圖片。



圖(八) 以 L*a*b 映射方式的 pix2pix 訓練流程

(四) GAN 損失模式於對比式學習所帶來的影響 (Vanilla GAN / LSGAN / WGAN-GP)

CUT 模型主要採用 CycleGAN 的架構及對比學習的訓練模式，而此研究欲探討加入不同的損失進行評估，是否會影響到模型生成的效能，這邊主要針對 Vanilla GAN、LSGAN（最小平方 GAN，Least Squares GAN)、WGAN-GP（Wasserstein GAN with Gradient Penalty）三種生成對抗模型中常見的損失模式進行探討，並評估

其生成的效能。

Vanilla GAN 是最簡單且最早被提出的生成對抗網路模型，它涉及兩個網絡：生成器（Generator）和判別器（Discriminator）。生成器產生假數據，判別器評估這些數據是否真實，而生成器根據判別器的評估進行改進，除了一般的對抗損失外，LSGAN 及 WGAN-GP 皆採用了不一樣的損失模式。

$$\min_G \max_D V(D,G)$$

$$V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z))]$$

LSGAN 由 Vanilla GAN 所衍生，也是 CUT 原始論文所採用的 GAN 損失模式，其與 Vanilla GAN 的主要區別在於，它使用 Least Squared Loss 作為判別器的損失函數。該函數的優點在於，它在處理數據分佈不均的問題時優於 Vanilla GAN，也就是透過使用平方損失來代替 Cross-Entropy（交叉熵），以改進判別器的判斷結果，使結果更平滑，可以數學形式表達如下:

$$\min_D V_{\mathrm{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}(\boldsymbol{x})}\left[(D(\boldsymbol{x})-b)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z}))-a)^2\right]$$

$$\min_G V_{\mathrm{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z}))-c)^2\right],$$

WGAN-GP 其與其他生成對抗網絡的主要區別在於，它使用 Wasserstein 距離作為生成器與判別器之間的評估標準。它還使用了梯度懲罰(Gradient Penalty)作為正則項，以避免判別器在經過預測後輕易抵達飽和狀態，進而改善生成器之性能，可以數學形式表達如下:

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}\left[(\|\nabla_{\hat{\boldsymbol{x}}}D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]}_{\text{Our gradient penalty}}.$$

（五）有無採用 Identity Loss 帶來的影響

CUT 模型的總損失公式可以寫成以下的數學形式，其中 λx 與 λy 為控制兩者比重的係數，λy 同時也代表著 Identity Loss。正常的 CUT 採用 Identity Loss（參數設為 λx=1， λy=1），若未採用 Identity Loss，該 CUT 架構稱作 FastCUT（參數設為 λx=10， λy=0），在原論文中，損失函數的數學形式可以表達如下:

$$\mathcal{L}_{\mathrm{GAN}}(G,D,X,Y) + \lambda_X \mathcal{L}_{\mathrm{PatchNCE}}(G,H,X) + \lambda_Y \mathcal{L}_{\mathrm{PatchNCE}}(G,H,Y).$$

Identity loss 在 CycleGAN 中的作用是保留輸入和輸出顏色組成的一致性，如果不添加該 Loss 的話，生成器可能會自主地修改圖像的色調，使得整體的顏色發生改變，如下圖所示。此階段欲透過比較 CUT 及 FastCUT 生成影像的表現優劣，以了解 Identity Loss 對於對比式的非監督式學習之影響。

原 CUT、FastCUT 模型針對歷史航照上色表現的差異

（六）BicycleGAN 潛在編碼加入位置帶來的影響

BicycleGAN 為 pix2pix 模型的一種變體，而在前述的研究過程中，可以發現 pix2pix 未經權重、生成器架構等調整，預設的模型於歷史航照的上色效果不甚理想，因此，本研究除了測試原 BicycleGAN 模型在上色中的表現外，也期許透過一些調整來讓這類的監督式學習達到較佳的效果。

經過訓練，BicycleGAN 可以找到潛在編碼 z(latent code)與目標圖像 B 之間的關係，因此生成器可以在給定不同的 z 之時學會生成具有多種不同特徵的目標圖像，過程中，我們嘗試改變 z 輸入給 Generator 的位置(首段、中段…等位置)，欲探討潛在編碼在何時加入生成器網路能讓歷史航照上色達到較好的效果。

五、生成模型指標的採用

生成模型產生的是高維的複雜結構數據，不同於判別模型，較難以簡單的指標來評估模型的好壞。因此在此研究中採用兩種近年研究中常用於評估生成模型的指標（僅判別圖像）：IS（Inception Score）和 FID（Frechet Inception Distance score）。

（一）IS（Inception score）

IS 是基於 Google 的預訓練網絡 Inception Net-V3 所設計的指標。Inception Net-V3 為一卷積網絡模型，輸入為圖片張量，輸出為 1000 維向量。輸出向量的每個維度的值對應圖片屬於某類的概率，因此整個向量可以看做一個概率分佈。以下為 IS 的兩個定義，IS 考慮以下兩個方面來評估生成器的質量：

1. 對於單一的生成圖像，Inception 輸出的概率分佈熵值應盡量小。越小說明生成圖像越有可能屬於某個類別，圖像質量就越高。可以數學表示如下:

$$
\begin{aligned}
& E_{x \sim p_G}(H(p(y|x))) \\
=& \sum_{x \in G} P(x) H(p(y|x)) \\
=& \sum_{x \in G} P(x) \sum_{i=1}^{1000} P(y_i|x) \log \frac{1}{P(y_i|x)}
\end{aligned}
$$

2. 對於生成器生成的一批圖像，Inception 輸出的平均概率分佈熵值應盡量大。也就是說，因為生成器應保證生成圖像的多樣性，因此一批圖像在 Inception 的輸出應盡量平均地“遍歷”所有 1000 維標籤。可以數學表示如下。

$$
\begin{aligned}
&H(E_{x\sim p_G}(p(y|x))) \\
=&H\left(\sum_{x\in G} P(x)P(y|x)\right) \\
=&H(p(y)) \\
=&\sum_{i=1}^{1000} P(y_i)\log\frac{1}{P(y_i)} \\
=&\sum_{i=1}^{1000}\sum_{x\in G} P(y_i,x)\log\frac{1}{P(y_i)} \\
=&\sum_{x\in G} P(x)\sum_{i=1}^{1000} P(y_i|x)\log\frac{1}{P(y_i)}
\end{aligned}
$$

將 2 個定義合而為一，可以得出以下的公式（如下圖所示），其中 KL(p(y|x)||p(y))為兩個分布的 KL 散度(相對熵)，再取指數即可得到 IS score，根據定義，IS 值越高，生成圖片的質量與穩定度越高。但考量到 IS 在評估時出現的部分問題，如:參數的微調會對 IS 影響頗大、無法反映模型是否出現 overfitting 等問題，因此本研究也參採了由 IS 進行改善的 FID score(Frechet Inception Distance score)。

$$
\begin{aligned}
&\sum_{x\in G} P(x)\sum_{i=1}^{1000} P(y_i|x)\log\frac{P(y_i|x)}{P(y_i)} \\
=&E_{x\sim p_G} KL(p(y|x)||p(y))
\end{aligned}
$$

$$
\mathrm{IS} = \exp E_{x\sim p_G} KL(p(y|x)||p(y))
$$

（二）FID (Frechet Inception Distance score，FID)

FID 的設計源自 Inception Net-V3，卻不使用該網路的原輸出，而是刪去了模型的輸出層，因此輸出層變為 Inception Net-V3 的最後一個池化層。該層的輸出為 2048 維向量，因此每張圖像都將被預測為 2048 個特徵。

對於常見的分佈（例如高斯分佈），只要確定了分佈類型和均值和方差，則該分佈就已確定。我們假設生成的圖像和真實圖像也遵循類似的分佈，如果它們之間的均值和方差比較接近，我們就有理由認為生成的圖像是比較真實的。但是直接計算圖像的均值和方差是不可取的，因為協方差矩陣規模太大（像素數*像素數）。因此，我們首先通過 Inception Net-V3 映射為 2048 維的特徵向量，再求特徵向量的均值和協方差矩陣進行比較。可以數學形式表達如下:

$$
\mathrm{FID}(x,g) = \|\mu_x - \mu_g\| + \mathrm{Tr}\left(\Sigma_x + \Sigma_g - 2\sqrt{\Sigma_x\Sigma_g}\right)
$$

這條算式中 μx, Σx 是真實圖像的 2048 維特徵向量集合的均值和協方差矩陣；而 μg, Σg 是生成圖像的 2048 維特徵向量集合的均值和協方差矩陣，Tr 代表一矩陣的跡。較低的 FID 分數代表生成的分佈更接近真實圖片分佈，若測試用的真實圖片清晰且多樣，也就意味著生成圖像品質高、多樣性佳。

(三)評估的方式與流程:

1. FID score 的計算需要兩個 datasets 去進行比較，而在此研究中，pix2pix。CUT 模型僅會生成上色後之圖像，因此只需比較其與正確 Ground truth 的相似度即可，而在跑 CycleGAN 的模型時，會生成兩組圖像，一組為由現代航照所生成的灰階影像，另一組為由灰階處理影像所生成的上色航照，分別與實際灰階處理影像、實際現代航照進行比較，取生得之兩 FID 數值平均。
2. IS（Inception Score）只需要輸入一組 dataset 即可計算出來，在此研究中，由於目標為生成仿真的古航照上色影像，故僅針對模型生成之上色圖像進行評估。

肆、 參考資料

[1] Taesung Park, Alexei A. Efros, Richard Zhang, Jun-Yan Zhu. **Contrastive Learning for Unpaired Image-to-Image Translation. In ECCV 2020.**

[2] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros. **Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In CVPR 2020.**

[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros. **Image-to-Image Translation with Conditional Adversarial Network. In CVPR 2017.**

[4] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, Eli Shechtman. **Toward Multimodal Image-to-Image Translation. In NIPS 2017.**

[5] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Sepp Hochreiter. **GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In NIPS 2017.**

*此篇報告書所使用之圖片，均擷取自各模型的原論文。

伍、 附錄

一、Inception Score 評估之程式碼

```
def inception_score(batch_size=args.batch_size, resize=True, splits=10):
    # Set up dtype
    device = torch.device(args.device)  # you can change the index of cuda
    # Load inception model
    inception_model = inception_v3(pretrained=True, transform_input=False).to(device)
    inception_model.eval()
    up = nn.Upsample(size=(299, 299), mode='bilinear', align_corners=False).to(device)
    def get_pred(x):
        if resize:
            x = up(x)
        x = inception_model(x)
        return F.softmax(x, dim=1).data.cpu().numpy()
```

```
    # Get predictions using pre-trained inception_v3 model
    print('Computing predictions using inception v3 model')
    files = readDir()
    N = len(files)
    preds = np.zeros((N, 1000))
    if batch_size > N:
        print(('Warning: batch size is bigger than the data size. '
               'Setting batch size to data size'))
    for i in tqdm(range(0, N, batch_size)):
        start = i
        end = i + batch_size
        images = np.array([imread(str(f)).astype(np.float32)
                           for f in files[start:end]])
        # Reshape to (n_images, 3, height, width)
        images = images.transpose((0, 3, 1, 2))
        images /= 255
        batch = torch.from_numpy(images).type(torch.FloatTensor)
        batch = batch.to(device)
        y = get_pred(batch)
        print(y.shape)
        preds[i:i + batch_size] = get_pred(batch)
    assert batch_size > 0
    assert N > batch_size
    # Now compute the mean KL Divergence
    print('Computing KL Divergence')
    split_scores = []
    for k in range(splits):
        part = preds[k * (N // splits): (k + 1) * (N // splits), :]  # split the whole
data into several parts
        py = np.mean(part, axis=0)  # marginal probability
        scores = []
        for i in range(part.shape[0]):
            pyx = part[i, :]  # conditional probability
            scores.append(entropy(pyx, py))  # compute divergence
        split_scores.append(np.exp(scores))
    return np.max(split_scores), np.mean(split_scores)
if __name__ == '__main__':
    MAX, IS = inception_score(splits=10)
    print('MAX IS is %.4f' % MAX)
```

```
    print('The average IS is %.4f' % IS)
```

除了 IS 評估的程式碼外，研究使用的程式碼將放置於 Github 平台上:
https://github.com/laiyuchi/ckmsc39th_research_GAN_colorization

二、網路可用之上色模型效果

這些模型針對歷史航照之上色，多出現飽和度不足、建築物輪廓模糊、上色不精確等現象，因此本研究才以 CNN 以外的影像生成模型作為研究的主體與方向。

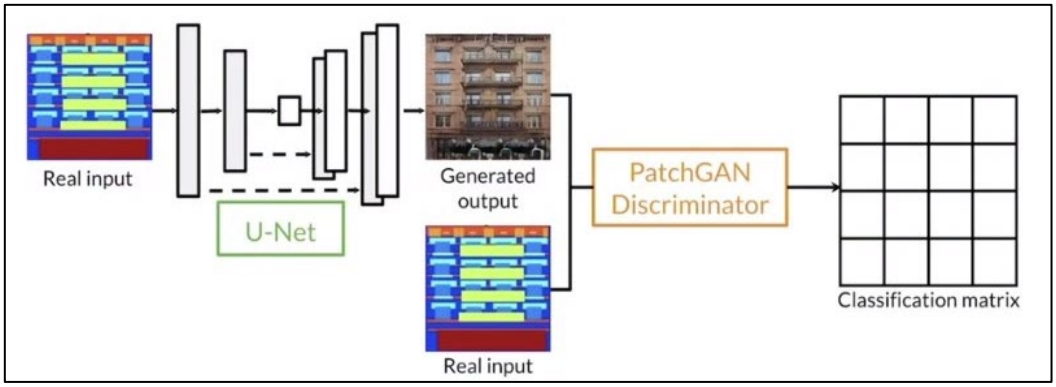
(1) Palette, a tool for colorizing photos

(2) DeOldify

三、各模型架構與損失函數的計算

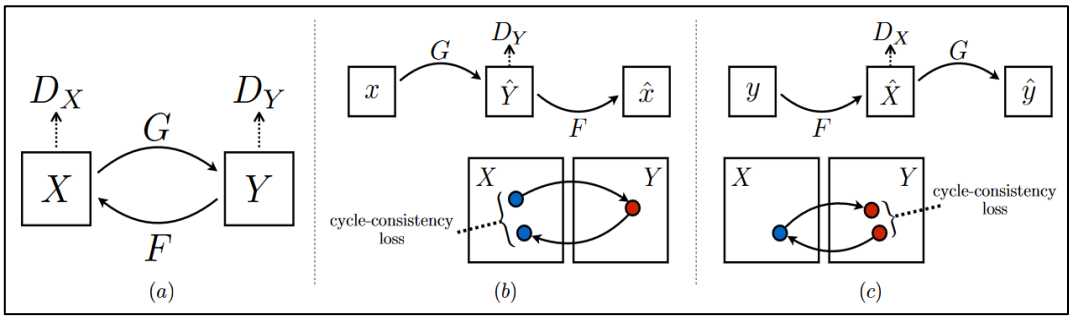
(一) 模型架構: 針對本研究採用之模型進行簡短地解說。

1. Pix2pix: 預設模型由 U-Net 架構之生成器與 PatchGAN 模式的判別器結合而成進行監督式學習。

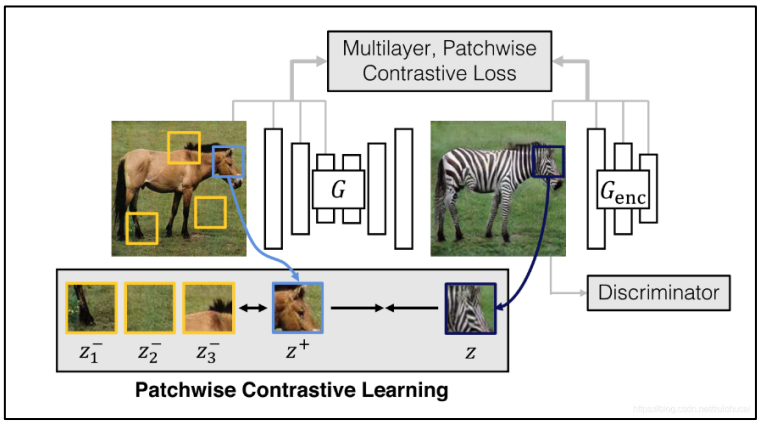


2. CycleGAN: 由兩組生成器、判別器組成，模型透過循環生成、判別的過程，達

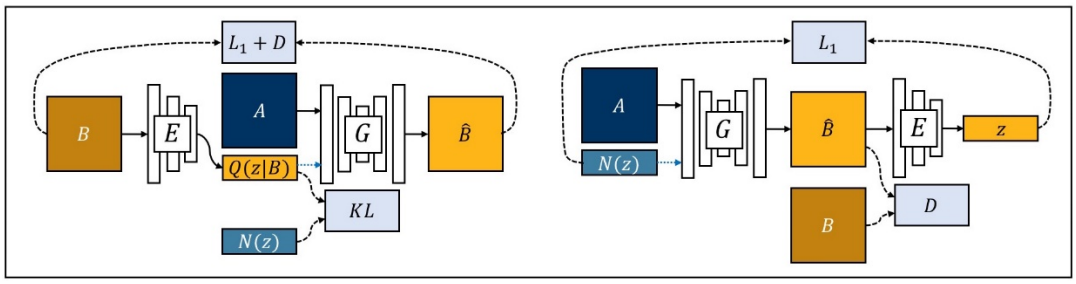
到影像生成的非監督式學習。



3. CUT (Contrastive Unpaired Translation): 為以 CycleGAN 為基底的模型，且使用多層圖像塊的對比損失，最大化相對應的多層圖像塊之間的共同資訊，這樣將生成器和 Encoder 相結合，取得對應輸入圖像的生成圖像。



4. BicycleGAN:該模型延續 pix2pix 具有的特性(skip-connection 等)，由 cVAE-GAN、cLR-GAN 組合而成，兩模型架構與特性詳見此連結。



（二）損失函數的定義與計算（公式細節請見參考資料附註之論文）

1. Pix2pix、BicycleGAN

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x,G(x,z))],$$

▲ Adversarial Loss（對抗損失）

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1].$$

▲ L1 損失

2. CycleGAN、BicycleGAN

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))],$$

▲ Adversarial Loss（對抗損失）

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].$$

▲ Cycle-consistency Loss（循環一致性損失）

$$\ell(\boldsymbol{v}, \boldsymbol{v}^+, \boldsymbol{v}^-) = -\log\left[\frac{\exp(\boldsymbol{v} \cdot \boldsymbol{v}^+/\tau)}{\exp(\boldsymbol{v} \cdot \boldsymbol{v}^+/\tau) + \sum_{n=1}^{N} \exp(\boldsymbol{v} \cdot \boldsymbol{v}_n^-/\tau)}\right].$$

▲ infoNCE Loss（資訊對比損失）

$$L_{Identity}(G, F) = \mathbb{E}_{y \sim p_{data}(y)}[\|G(y) - y\|_1] + \mathbb{E}_{x \sim p_{data}(x)}[\|F(x) - x\|_1]$$

▲ Identity Loss

四、致謝

感謝中央大學蔡宗翰教授、中研院人社中心廖炫淯研究副技師、人社中心陳哲安學長指導，以及一路上協助研究的每位老師。