AI CUP 2024 春季賽
以生成式 AI
建構無人機於自然環境偵察時所需之導航資訊
競賽 I — 影像資料生成競賽
競賽報告

隊伍：TEAM_5556
隊員：郭庭榛（隊長）
Private leaderboard：113.5708 / Rank 12

壹、環境

一、開發環境：Google Colaboratory
二、作業系統：Ubuntu 22.04.3 LTS
三、語言：Python 3.10.12
四、GPU
由於運算單元有限，訓練模型與測試模型時使用不同 GPU 資源：
1. 訓練模型時使用 NVIDIA A100-SXM4-40GB
2. 測試模型時使用 Tesla T4

五、套件

1.	dominate	2.9.1
2.	numpy	1.25.2
3.	Pillow	9.4.0
4.	pip	23.1.2
5.	torch	2.3.0+cu121
6.	torchvision	0.18.0+cu121
7.	scipy	1.11.4
8.	google.colab	1.0.0

六、預訓練模型：無
七、額外資料集：無

*註：由於使用 google colab，無法直接以本機 host 使用 visdom，故本次並無使用

貳、演算法與模型架構

本次競賽的目標為「利用二值圖生成無人機視野下的真實影像」，訓練資料集中的二值圖以一對一的方式對應真實影像，因此，我選用需要一對一對應影像作為訓練資料的 pix2pix 模型進行訓練。

Pix2pix 為 Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros (2017) 所提出的圖像轉換模型。此模型以 GAN（Generative Adversarial Networks）概念為基礎進行改良，目標是以條件式生成對抗網路，設計出通用於絕大多數圖像對圖像轉換應用的模型 。以下為模型說明：

一、基本架構——GAN（Generative Adversarial Network）生成對抗網路

GAN 主要由兩個互相競爭的神經網路構成：生成器（Generator）、辨別器（Discriminator）。以下為兩者之說明：

1. 生成器

藉由輸入至模型的隨機雜訊來生成假影像，其目標是盡可能生成近似真實的影像，使辨別器無法分辨真假。

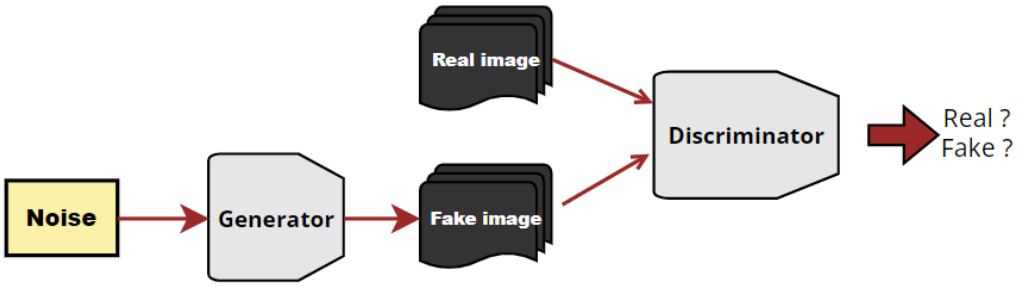
2. 辨別器

區分影像真偽，目標是在訓練過程中不斷加強辨識真假影像的能力。

生成器與辨別器透過訓練時兩者的對抗，使生成器能生成出更逼真的影像、辨別器有更精準的辨識能力。模型架構示意如圖 1。

圖 1

生成對抗網路模型架構示意圖



二、pix2pix 架構

1. cGAN（conditional GAN）條件生成對抗網路

由上述第一點可知，原始的 GAN 模型，生成器僅是透過輸入的隨機雜訊來生成圖片，然而 pix2pix 進行了改良，它是透過輸入圖片來生成真實影像，而此改良正是基於 cGAN 模型概念：增加輸入條件，必須輸入影像或其它規範的訊息，使生成器生成相關內容。

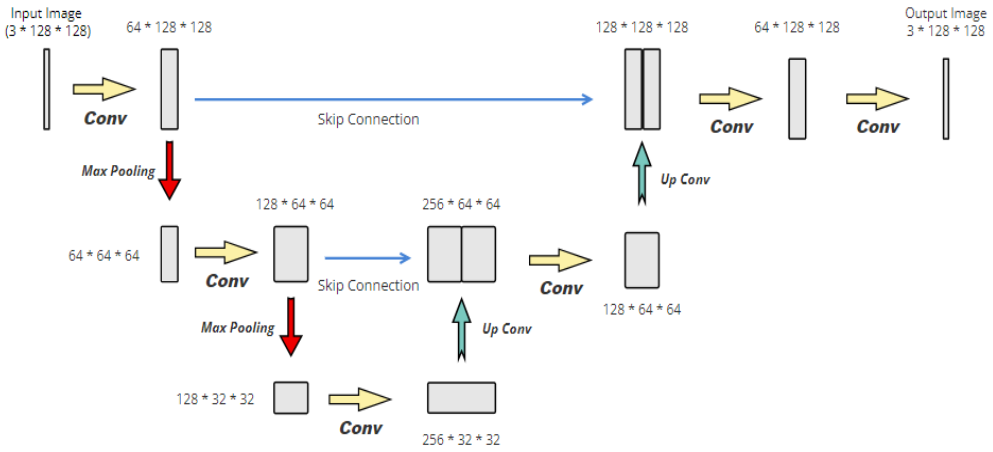
2. 生成器架構——U-Net

U-Net 架構是對原先生成器常見的編碼器－解碼器架構（Encoder-decoder）進行改良，並在這個架構基礎下，引入跳躍連階（skip connections），讓模型可以保留更多不同特徵（Ronneberger, Fischer, & Brox, 2015）。

編碼器將輸入的圖片，經由多個卷積層提取特徵，逐步增加特徵通道數，再經由最大池化層對圖片做下採樣處理，以達成降維目的。解碼器則是經由反卷積層進行上採樣，以恢復圖片大小，並利用卷積層逐步減少特徵通道數，最終生成目標圖片。至於跳躍連接，則是將編碼器中每一層的特徵圖，直接合併至對應的解碼器層，讓低階特徵能夠保留，不會在下採樣過程中喪失。（圖 2）

圖 2

U-Net 架構示意圖



3. 解碼器架構——PatchGAN

PatchGAN 辨別器架構，是針對影像上每一個 N * N 大小的 patch 進行真假判斷，而不像普遍的辨別器是直接整張影像拿來判斷，其目的是讓辨別器聚焦在高頻訊息 。

4. 損失函數

a. LcGAN（對抗損失）：

普遍使用 vanilla 模式，意及對抗損失是基於二元交叉熵，用於生成器與判別器的對抗訓練。生成器的目標是盡可能降低此項損失，表示騙過判別器的能力越高，而判別器的目標是提高此項損失，代表判別力越強。公式如下：

$$LcGAN(G,D) = E_{x,y}[logD(x,y)] + E_{x,z}\left[log\left(1 - D\left(x, G(x,z)\right)\right)\right]$$

b. L1 Loss

L1 Loss 可以計算生成圖片與真實圖片的誤差，其功用在於，保留更多真實圖片的細節，使生成圖片能與真實圖片更為接近。公式如下：

$$L_{L1}(G) = E_{x,y,z}[||y - G(x,z)||_1]$$

c. 總損失函數

$$G^* = arg\,\min_G \max_D\, LcGAN\,(G,D) + \lambda L_{L1}(G)\ \ (\lambda : L1\ Loss權重)$$

5. 優化器

通常使用 Adam 優化器，是一種結合動量法與 RMSProp 的自適應學習率優化算法，意即優化器能夠自行調整生成器與辨別器的學習率。它有兩個 beta 參數，beta1 是調整一階矩衰減率，pix2pix 預設為 0.5，而 beta2 調整二階矩衰減率，pix2pix 預設為 0.999。

參、技術模型原創性或改良成效

本次競賽，我主要是利用主辦單位提供之 baseline code（Zhu, J., Park, T., Wang, T., 2017）進行實作，但是有針對 U-Net 架構進行簡單改良。

原始程式碼中的生成器 U-Net 架構，僅有 unet_128 與 unet_256。根據原作程式碼註解得知，unet_128 適用於輸入圖片大小 128x128，而 unet_256 適用於輸入圖片大小 256x256。前者下採樣 7 次，後者下採樣 8 次，會讓圖片在架構瓶頸處縮減至大小 1x1。由於最終前處理完的資料集圖片大小為 512x512，雖然也可以使用 unet_256 或 unet_128 當作生成器架構，但這樣生成器深度不夠使圖片縮減至 1x1，表示圖片雖然變大了，但是採樣次數不變，可能無法最有效提取特徵，因此我參考原始碼的處理方式，新增了 unet_512 架構，使輸入圖片經過九次下採樣卷積層提取特徵，最終在瓶頸處大小變為 1x1，符合原始程式碼概念。

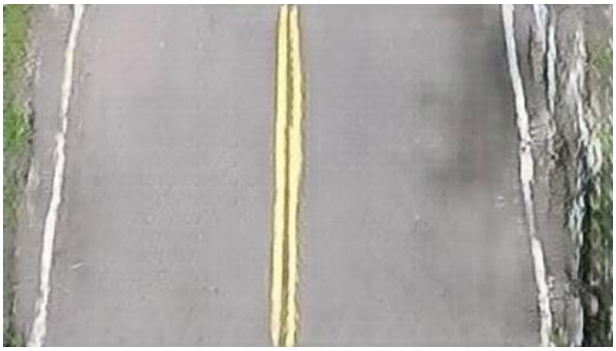
經由訓練成果觀察，在適當的參數條件下，我覺得我改良的架構在道路圖上較有正面效果（請見圖 3、圖 4、圖 5，圖片為 PRI_RO_1000391）。由圖 3、圖 4 可看出，使用 unet_512 作為生成器的生成圖片，它的特徵較為清晰，影像生成完整度也較優良， 尤其是中線與邊線的生成更為完整。至於河流圖，經由多次嘗試，我覺得此次改良相對於道路圖來說，並沒有太明顯的效果，因為我使用生成以下兩張圖片的模型參數去訓練河流圖生成模型，觀察生成的圖片，沒有太大差異。

圖 3

使用 unet_256 生成器架構訓練生成圖

圖 4

使用 unet_512 生成器架構訓練生成圖



肆、資料分析與處理過程

一、資料集說明

1. 圖片種類

本次競賽的目標為「使用河流及道路的邊界與中軸線二值圖，生成對應的真實視野影是像圖片」，故資料集中共有兩種類別的圖片，其一為 label image（二值圖），其二為無人機視野下的真實圖片。

2. 圖片大小

統一為 428 像素* 240 像素

3. 圖片內容分類

無論是河流還是道路圖，圖片皆由天氣、拍攝角度、河流或道路佔影像比例的不同構成不同內容。

河流圖共分為 18 類，由三種天氣（晴天、陰天、雨天後）、兩種視角（向下 90 度、向前下 15 度），以及三種河流佔影像比例（佔 20%、50%、80%）互相搭配而成。

道路圖共分為 12 類，由三種天氣（晴天、陰天、雨天後）、兩種視角（向下 90 度、向前下 15 度），以及四種道路佔影像比例（佔 30%、50%、60%、75%）互相搭配而成。

4. 資料集分配

a. 訓練資料集

共含有 4320 張 label images（二值圖）與 4320 張對應之無人實際拍攝的真實圖片。其中，河流圖與道路圖各有 2160 組一對一 label image 與真實圖片。

b. 公開測試資料集

共含有 720 張 label images。其中，河流與道路之 label image 各有 360 張。

c. 私有測試資料集

共含有 720 張 label images。其中，河流與道路之 label image 各有 360 張。

二、資料前處理

依據本次競賽的生成目標與資料集內容所知，須利用訓練資料集的一對一對應圖片，進行監督式學習，故呼應第貳點報告內容，我使用 pix2pix 架構進行模型訓練，因此在訓練前先做了對應的前處理。以下說明處理過程與理由：

1. 等比例圖片放大

本次競賽，我以主辦單位提供之 baseline code 為基本架構進行改良嘗試。在原先的程式碼中，pix2pix 生成器架構為 U-net 256 或 U-net 128，但是本次資料集的圖片大小為 428*240，需要先進行放大或縮小的處理，才可以符合模型架構的輸入要件。在原始程式碼中，雖然可以直接利用 Resize and Crop 或是 Scale width and Crop 進行圖片大小處理，但是資料集中，河流與道路的周圍環境，會因為河流與道路的樣貌不同而有所不同，若是隨機剪裁時，未剪到 label image 有邊線的部分，模型無法判斷該圖片周圍環境，以致未能有效訓練。另外，若是

直接以不等比例放大長與寬，我認為會讓真實場景的細節失真度變高，所以我並無使用原始程式碼提供之處理方式，而是選擇以長寬等比例縮放圖片。

接者，依據日常經驗來看，若放大圖片，能讓圖片的細節更為肉眼可見，但同時圖片噪點會更多；若縮小圖片，相對於放大圖片，可能會造成部分圖片細節喪失，但噪點較少。觀察競賽資料集後發現，無論是河流還是道路，在角度、光線、河流與道路佔比上，都有其多元性，並非單一化的構造，因此我希望能保留圖片更多的細節，我決定將資料集圖片放大至長為 512 像素、寬依長的放大倍率放大（約為 287.1 像素），以保留更多不同的細節。

2. 圖片填充

基於前項說明，我將圖片長寬等比例放大，長邊為 512 像素，因此參考 baseline code，我新增了 U-net 架構選項——U-net 512。針對此架構，輸入圖片的大小須為 512*512，所以，我選擇填補圖片以吻合模型輸入條件。

填充方式如下：將大小 512*287.1 的圖片置於正中間，上半部與下半部空白部分別約為 512*112.45，將這兩部分空白處皆填上白色，這樣圖片大小就會全數符合 512*512。選擇白色填補是因為本資料集的 labal image 僅由黑白兩色構成，又因邊界線與中軸線皆為白色，所以在不增加其餘色彩的情況下，以白色作為填補色，恰好當作圖片整體邊框。

3. 合併訓練資料集 label image 與真實圖片

為符合程式碼圖片處理方式，在完成前兩項處理後，使用 baseline code 中 *combine_A_and_B.py* 檔案進行 label image 與真實影像圖片水平合併。

伍、訓練方式

本次競賽，我首先嘗試對兩種不同場景的影像，使用近似 baseline code 原始設定的相同權重進行訓練，發現河流與道路圖在相同權重條件下，河流圖訓練時的收斂速度會比道路圖還要快，因此我後來採取不同的模型權重設定，分別對兩種場景進行訓練。

最終繳交的成果，河流圖與道路圖在訓練時，生成器皆是使用 U-Net_512 架構，其最後一層卷積層的 filters 數量維持原始設定為 64；而辨別器皆是使用 basic（70*70 PatchGAN）架構，其第一層卷積層的 filters 數量也維持原始設定為 64；正規化則使用 Batch Normalization。至於兩種場景不同的權重設定，我是在每次訓練過程中，觀察損失函數與生成的假圖片，並針對不同狀況進行參數微調。主要調整的參數是 lambda_L1、lr（學習率）、epoch 數量。以上三種參數調整的原則為：

1. lambda_L1

此參數是調整 L1 Loss 的權重。L1 Loss 的功用是盡可能讓生成的圖片與

真實圖片樣貌接近，但是同時會讓影像變得較模糊，所以若訓練過程中，圖片越來越模糊，我會將參數微調降低，若圖片細節損失越來越嚴重，則會微調升高。

2. lr（學習率）

學習率大有助於模型初期快速收斂，但是若沒有適當調整，到中後期反而會震盪導致無法收斂至最佳模型，反之，學習率小有助於模型精細學習，但是速度會減慢，且容易過擬合。所以依據以上原則，若我觀察模型損失函數震盪過嚴重，則會將學習率微調降低，但若觀察損失函數都沒有降低，則會將學習率微調升高。

3. epoch 數量

此參數調整主要以我擁有的計算資源，以及模型訓練的收斂速度為考量。由於我擁有的計算資源不夠豐富，加上訓練資料集並不龐大，因此會控制 epoch 數在 200 個上下，再依據生成圖片狀況進行微調。

接下來針對繳交模型進行訓練過程說明：

1. 河流圖模型訓練

最終繳交成果是我訓練的第 9 個版本，此模型訓練共分為三個階段。第一階段，我調整訓練率為 0.0001、lambda_L1 為 110，並運行 100 個 epoch。觀察生成狀況，我認為收斂不達預期，所以第二階段我再以相同的條件運行 30 個 epoch，但觀察損失函數，我認為收斂效果有限，且圖片較為模糊，所以進行第三階段微調。第三階段微調，我將 lambda_L1 降低成 100，而訓練率同樣設定初始為 0.0001，但是會進行線性降低。最終我使用第三階段微調的 90 個 epoch 生成模型進行圖片生成。

2. 道路圖模型訓練

最終繳交成果是我訓練的第 12 個版本，此模型訓練分為兩階段。第一階段，由於在前幾次訓練過程中，我覺得道路圖的細節比起河流圖更難訓練，所以我調整 lambda_L1 為 115，訓練率嘗試 0.00015，並設定 50 個 epoch 後開始線性調整。第一階段運行 100 個 epoch 過後，我觀察圖片模糊程度偏高，於是第二階段微調 lambda_L1 為 100，又運行了 70 個 epoch 後結束訓練。

陸、結果分析與結論

此次競賽是我第一次自己研讀論文及進行深度學習實作，因此這次的過程與成果，我覺得有諸多需要詳加檢討的問題，以下將說明訓練結果以及待改進之處：

一、訓練結果分析

1. 河流圖與道路圖在相同參數條件下的生成品質是有出入的，在適當訓練權重下，河流圖的收斂速度會比道路圖快，推測是因為本次資料集中，河流圖的場景複雜度較低，基本上是由草和河流組成，而道路圖除了草叢、道路本身外，還有道路線條特徵需要學習。當然，河流圖

周遭的田野環境也具有複雜度，但是我在多次訓練中，始終沒有能生成更真實的田野結構（請見圖 5），這部分是需要改進的地方。

2. 模型訓練時，學習率若設定較大，如 0.0002，應當運行較少 epoch 就進行線性衰減，否則容易過擬合。圖 6-1 與圖 7-1 是兩個不同模型所生成的圖片，圖 6 初始訓練率為 0.0002，圖 7 則為 0.0001，可看出圖 6 收斂很快，在 epoch 數不多的情況下就生成接近真實的圖片場景分布，圖 7 則較為模糊且缺少細節。但是經過多個 epoch 訓練後進行測試，生成圖 6 的模型完全無法生成正常的圖片（請見圖 8），反倒是生成圖 7 的模型表現較好（請見圖 9）。

二、檢討與改進方向

1. 輸入通道數

二值圖僅由黑與白構成，並無其它色彩，下次可以嘗試以單通道方式輸入模型架構進行訓練。

2. 預處理方式改進

此次訓練填充大量無意義之白色區塊，我個人認為在進行提取特徵時，會浪費很多資源提取不必要的特徵，再加上我是放大圖片再進行填充，更耗費學習資源。

改進方法為嘗試合理剪裁（例如：考慮中軸線所在位置進行剪裁），使圖片每次只訓練部分特徵，能節省訓練資源，並且聚焦在小區域的特徵上。

3. 分割資料及方式改進

這次分割訓練資料集與驗證資料集時，我是使用隨機方法，但是本次競賽的資料集，其實不管是河流圖還是道路圖，都有不同的光線、角度、河流或道路佔比，意即，若我使用隨機分割的方法，當沒辦法把不同類型的圖片平均分割為訓練集，就會發生特定種類的圖片樣本不夠多，很可能因此在生成該類圖片時嚴重失真。

改進方法為人工先篩選一遍，將類似圖片標記為同一類，再由每一類取一樣的比例進入驗證集，保留相同比例在訓練集，可以減少資料不平衡的事情發生

4. 適當的邊緣處理方式

多次訓練後，我發現生成的結果普遍會在邊緣出現一塊一塊的構造（如圖 10），使生成圖片一眼看就知道是假的，所以針對圖片邊緣處理進行加強是我改進的方向。

圖 5

河流生成圖片田野失真示意圖

*註：紅色圈框選處為較嚴重失真的區域，肉眼可見模型生成的周邊道路是扭曲的。

圖 6-1

初始學習率 0.0002 模型於訓練時生成圖片

*註：初始學習率 0.0002、生成器架構 unet_512，其餘為預設條件，訓練第 50 個 epoch。

圖 6-2

圖 6-1 之真實圖片

圖 7-1

初始學習率 0.0001 模型於訓練時生成圖片

*註：初始學習率 0.0001、生成器架構 unet_512、lambda_L1 = 110 運行 50 個 epoch。

圖 7-2

圖 7-1 之真實圖片

圖 8

圖 6 河流模型架構訓練 195epoch 生成圖片

*註：初始學習率 0.0002、生成器架構 unet_512，其餘為預設條件（100epoch 以後會線性衰減學習率），訓練 195 個 epoch。

圖 9

圖 7 河流模型架構訓練 195epoch 生成圖片

*註：初始學習率 0.0001 運行 130epoch，線性衰減 65epoch、生成器架構 unet_512、lambda_L1=110 運行 130epoch，lambda_L1=100 運行 65epoch；共運行 195epoch。

圖 10

邊緣結構失真示意圖

*註：紅色框選處為此次競賽生成圖片時，時常生成的格狀構造

柒、程式碼

Github 連結：https://github.com/Tingouoww/aicup2024.git

大型檔案雲端連結：https://drive.google.com/drive/folders/1-jQx8Z9FtfxWSRHC5nv1TgTz36S0h308?usp=sharing

捌、使用的外部資源與參考文獻

一、參考文獻

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 234-241). Springer, Cham.

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1125-1134).

Zhu, J., Park, T., Wang, T.(2017). CycleGAN and pix2pix in PyTorch. GitHub repository. Retrieved from https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

邱泓翔（2019）。運用 GAN 實現字體風格轉換。2019 年臺灣國際科學展覽會

優勝作品專輯。
Allen, T. (2019). Adagrad、RMSprop、Momentum and Adam – 特殊的學習率調整方式. Retrieved from https://hackmd.io/@allen108108/H1l4zqtp4

報告作者聯絡資料表

隊伍名稱	TEAM_5556	Private Leaderboard 成績	113.5708	Private Leaderboard 名次	12
身分 (隊長/隊員)	姓名 (中英皆需填寫) (英文寫法為名, 姓，例：Xiao－Ming, Wu，名須加連字號，姓前須加逗號)	學校＋系所中文全稱 (請填寫完整全名，勿縮寫)	學校＋系所英文中文全稱 (請填寫完整全名，勿縮寫)	電話	E-mail
隊長	郭庭榛 Ting － Chen, Kuo	國立成功大學工程科學系	National Cheng Kung University Department of Engineering Science	0984-111-620	tingouo20@gmail.com
隊員 1					
隊員 2					
隊員 3					
隊員 4					
指導教授資料					
每隊伍至多可填寫兩名	指導教授 中文姓名	指導教授 英文姓名 (英文寫法為名, 姓，例：Xiao－Ming, Wu，名須加連字號，姓前須加逗號)	任職學校＋系所 中文全稱 (請填寫完整全名，勿縮寫)	任職學校＋系所 英文全稱 (請填寫完整全名，勿縮寫)	E-mail
教授 1					
教授 2					