

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 II — 導航資料生成競賽

競賽報告

隊伍：TEAM_5171

隊員：張翔（隊長）、王睦閔

Private leaderboard：0.782478 / Rank 9

壹、環境

作業系統：Linux Ubuntu 16.04.4 LTS

GPU：Tesla P40

語言：Python 3.12.3

套件（函式庫）：

torch==2.3.0

torchvision==0.18.0

pillow==10.3.0

openCV-python==4.9.0.80

numPy==1.26.4

matplotlib==3.9.0

PySODMetrics(<https://github.com/lartpang/PySODMetrics>)

預訓練模型：VGG16，來自 [PyTorch 官方套件](#)

額外資料集：無使用額外資料集

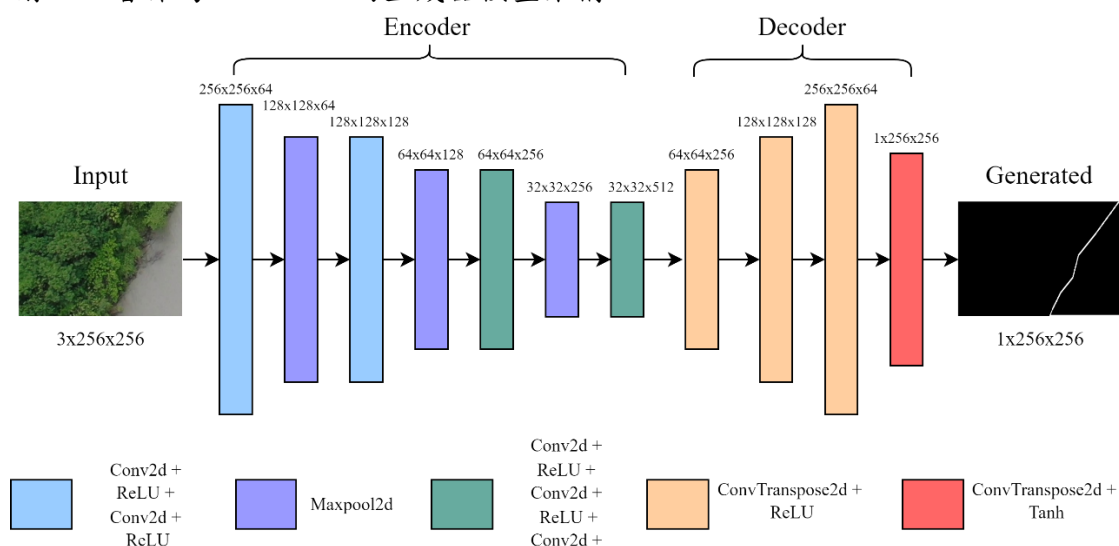
貳、演算法與模型架構

我們使用 GAN 模型架構來完成這次任務。GAN 是由生成器與判別器組合而成，我們使用兩個不同的 GAN 模型架構，其中，第一個 GAN 的生成器採用 Encoder-Decoder 架構，第二個 GAN 的生成器則採用 Unet 架構，而這兩個 GAN 的判別器皆採用 pix2pix 模型 [1] 中的 PatchGAN 架構。以下依序簡介我們兩個 GAN 模型架構。

1. GAN (EncoderDecoder-PatchGAN)：

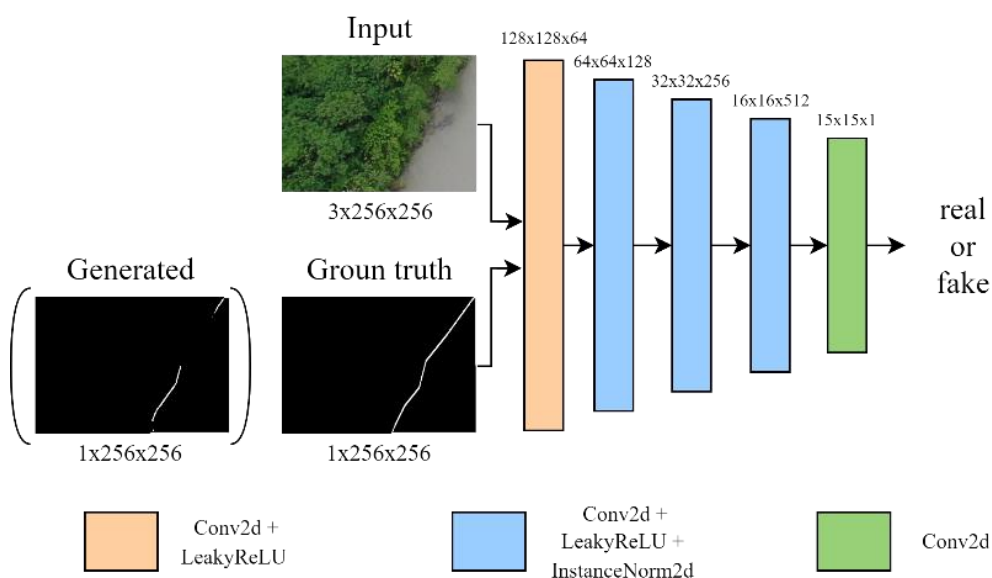
生成器：第一個 GAN 模型中，我們的生成器架構是基於 VGG16 架構進行延伸，並且包括一個編碼器和解碼器。編碼器包含了 VGG16 預模型的前 23 層，這些層主要由卷積層和最大池化層組成，負責提取輸入圖像的高層次特徵。解碼器由一系列反卷積層和激活函數 ReLU 組成。輸入的圖像首先經過編碼器提取特徵，然後將提取的特徵圖傳入解碼器進行解碼，最終輸出還原的圖像。通過編碼

和解碼的過程，能夠對輸入圖像進行有效的處理和還原。圖一為我們以 VGG16 前 23 層作為 encoder 的生成器模型架構。



圖一

判別器：我們的 PatchGAN 架構的判別器是由五個連續的卷積塊組成，前四個卷積塊中，每個卷積塊包括一個卷積層、一個 InstanceNorm2d，和一個激活函數 LeakyReLU，其中 InstanceNorm2d 可以提升訓練的穩定度。而最後一個卷積塊則是一個卷積層，負責輸出判別結果。圖二為我們的判別器模型架構。

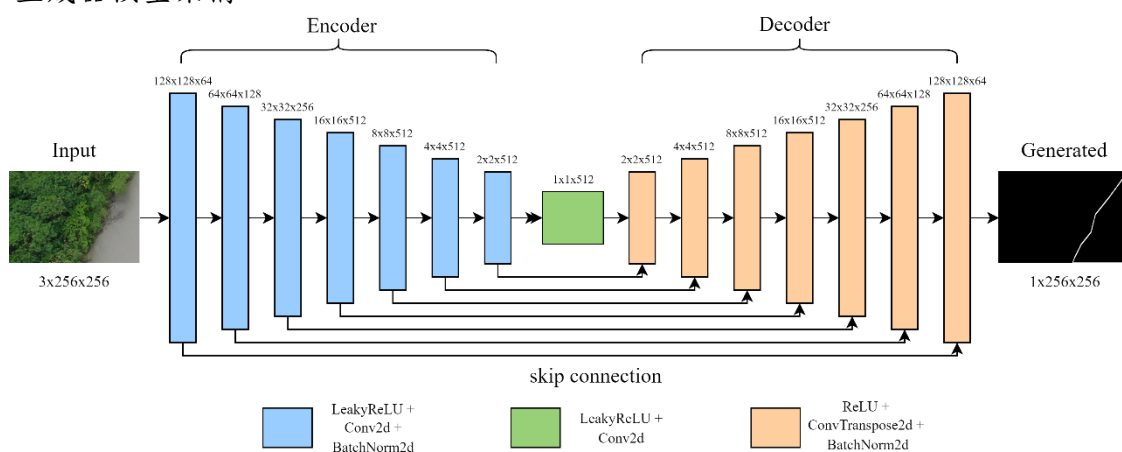


圖二

2. GAN (Unet-PatchGAN)：

生成器：第二個 GAN 模型的生成器參考了 pix2pix 模型的生成器架構-Unet。編碼器通過一系列的卷積層逐漸將輸入圖像壓縮為低維特徵圖。最初的輸入是 3 通道圖像，經過第一個卷積層變為 64 通道，接著通過多個 EncoderBlock 進一步提取特徵，並逐步增加通道數直至 512。這些 EncoderBlock 包含激活函數 LeakyReLU 和卷積層，部分還包含 BatchNorm 進行正規化，以幫助穩定

訓練過程。在解碼器部分，模型通過一系列的反卷積操作逐漸將特徵圖恢復到與輸入圖像相同的尺寸。在這個過程中，每個 DecoderBlock 將特徵圖放大，同時減少通道數。部分 DecoderBlock 包含 Dropout 層，以防止過擬合。解碼過程中，使用了 skip connection，即將編碼器對應層的輸出與解碼器的輸出拼接，這樣有助於保留更多原始圖像的細節訊息，提高生成圖像的品質。判別器則採用與第一個 GAN (EncoderDecoder-PatchGAN) 相同的架構。圖三為我們的 Unet 生成器模型架構。



圖三

參、技術模型原創性或改良成效

融合 VGG16 特徵提取器：我們使用預訓練的 VGG16 模型作為編碼器，利用 VGG16 在 ImageNet 上訓練所得的豐富特徵，有助於提升生成圖像的品質。解碼器使用了多層轉置卷積層來逐步恢復圖像空間分辨率。這種方法不同於傳統的簡單上采樣，能夠更好地保持圖像的細節和結構。

採用 ensemble 技術：通過組合兩個以上模型的輸出來提升生成圖像表現。由於我們設計了兩個不同的 GAN 模型，一個是 EncoderDecoder-PatchGAN，另一個是 Unet-PatchGAN，我們將兩個模型的生成圖像的每個像素值以不同的比例作加權平均，把加權平均後的圖像作為最後的結果。我們發現 ensemble 後的圖像比單一使用 EncoderDecoder-PatchGAN 或 Unet-PatchGAN 模型的生成圖像表現更好，其 Mean F-measure 分數有顯著地提升。

肆、資料分析與處理過程

在提供的訓練資料集 (35_Competition 2_Training dataset_V3.zip) 中包含了 2160 張河流圖和 2160 張道路圖，總計 4320 張圖像。為了進行模型的訓練、驗證和測試，我們按照 8:1:1 的比例先將訓練資料集切分為訓練資料 (3456 張)、驗證資料 (432 張) 和測試資料 (432 張)，並確保訓練資料中的河流圖和道路圖各佔一半以保持資料平衡。

我們認為 3456 張圖像可能無法讓模型充分學習到各種特徵，因此我們採用了資料擴增的方法，對所有訓練資料個別進行水平翻轉以及垂直翻轉，並將擴增後的圖像加入至訓練資料中。執行以上兩個資料擴增方法後，新增了 3456 張水平翻轉圖像以及 3456 張垂直翻轉圖像。

此外，由於提供的訓練資料中，有各種天氣狀況下的圖像，其亮度表現有些微的不同，因此我們根據這個特點對訓練資料進行亮度調整，使用 ColorJitter 套件，參數 brightness 設定為 0.5，表示在 0.5 的範圍內對圖像的亮度進行隨機調整，新增 3456 張亮度調整後的圖像。

經過以上三種資料擴增方法後，我們的訓練資料總數增加至 13864 張，其中包含原圖像、水平翻轉圖像、垂直翻轉圖像、亮度調整圖像，各 3456 張圖像。

另外，為了符合模型的輸入格式，我們將圖像長寬 resize 至 256x256 大小，並以平均值 $[0.5, 0.5, 0.5]$ ，標準差 $[0.5, 0.5, 0.5]$ 將每個通道的像素值縮放到 $[-1, 1]$ 的範圍內，以提高模型的訓練穩定性和收斂速度。

伍、訓練方式

我們的訓練資料 batch size 設定為 64，兩個 GAN 模型的生成器及判別器的優化器皆為 Adam，學習率設定為 0.0002，beta 參數分別設定為 0.5 以及 0.99。定義了兩個損失函式，BCE Loss 以及 L1 Loss。並將 epoch 設定為 1000。

首先先訓練判別器，輸入原始圖像和目標圖像，以 BCE Loss 計算其真實損失；生成器生成假圖像，再將原始圖像和假圖像輸入判別器，計算其虛假損失。這兩部分損失的平均值作為判別器的總損失。對於判別器的損失進行反向傳播和權重更新。

接著訓練生成器，生成器生成假圖像，並與原始圖像輸入至判別器，計算其真實性的損失，同時還計算生成圖像與目標圖像之間的 L1 Loss，控制生成圖像與目標圖像的相似度。並將這兩部分損失相加作為生成器的總損失，對於生成器的損失進行反向傳播和權重更新。特別的是，我們每 5 個 epoch 才會進行一次生成器模型權重的更新，這樣做可以避免模式崩潰 (mode collapse)。

每個 epoch 結束後，會計算並記錄生成器和判別器的平均訓練損失。並且每隔 5 個 epoch 會保存當前生成器的模型權重，同時生成驗證資料圖像來觀察訓練效果。訓練完成後，挑選在測試資料上表現較好 (Mean F-measure 分數較高) 的模型權重。我們挑選了四個不同 epoch 的 Unet-PatchGAN 生成器，與一個 EncoderDecoder-PatchGAN 生成器作為後續做 ensemble 的模型，以提升生成品質。

陸、結果分析與結論

我們選擇了訓練至 465、645、680 以及 775 個 epoch 的 Unet-PatchGAN 生成器模型，以及一個訓練至 230 個 epoch 的 EncoderDecoder-PatchGAN 生成器模型。此外，我們的模型是生成單通道灰階圖，但我們觀察到目標圖像是二值化的圖像，因此我們對模型的生成結果利用大津演算法 (Otsu's algorithm) 進行二值化，接著做形態學運算 (Morphological operation) 的 opening 進行細節上的修補。以下為 ensemble 前後，各模型在測試資料上的分數及後處理過後的生成結果。

Unet-PatchGAN (epoch 465)：

before binarizing: 0.6748

after binarizing: 0.7092

after binarizing and opening : 0.7233

result :



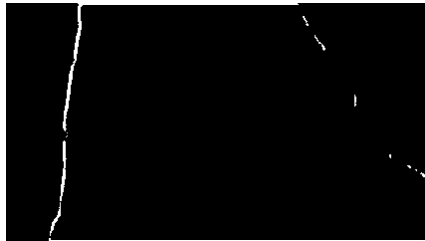
Unet-PatchGAN (epoch 645) :

before binarizing : 0.6706

after binarizing : 0.7035

after binarizing and opening : 0.7189

result :



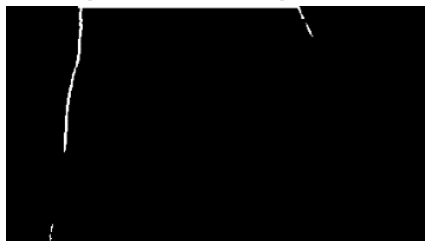
Unet-PatchGAN (epoch 680) :

before binarizing : 0.6817

after binarizing : 0.7244

after binarizing and opening : 0.7360

result :



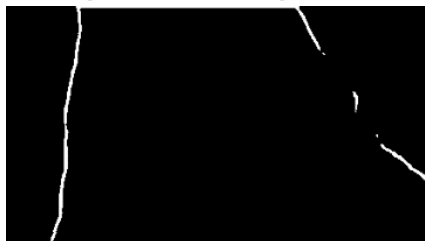
Unet-PatchGAN (epoch 775) :

before binarizing : 0.6736

after binarizing : 0.7017

after binarizing and opening : 0.7186

result :



EncoderDecoder-PatchGAN (epoch 230) :

before binarizing : 0.6554

after binarizing : 0.7061

after binarizing and opening : 0.7250

result :



ensembled (weight=0.12, 0.12, 0.33, 0.1, 0.33 respectively) :

before binarizing : 0.6824

after binarizing : 0.7484

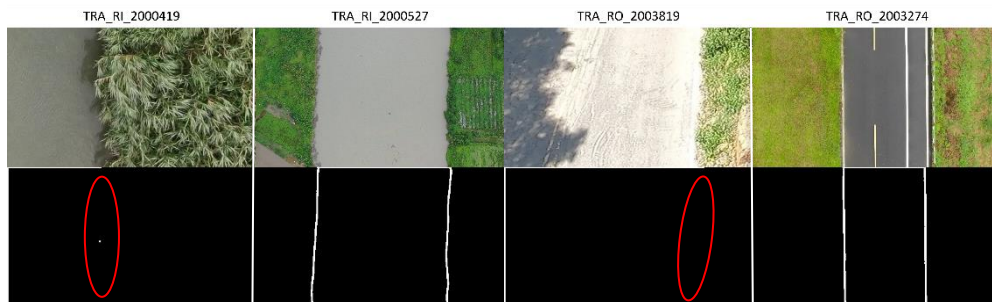
after binarizing and opening : 0.7558

result :



可以觀察到，對生成圖像接續做了二值化及 opening 後，分數皆有明顯提升，且對各模型的生成圖像進行 ensemble 後再做二值化及 opening，也有得到更高的分數。

此外，我們發現我們的模型在沒有明顯邊緣且有樹叢的地方經常判斷不出是否是邊界，導致生成圖在該部分的線段沒有連接起來（紅色圈起處），如圖四所示。



圖四

針對這個問題，我們認為可以從兩個部分做改進。首先第一個部分，可以針對資料集原圖像做更多的前處理，例如可以利用邊緣偵測 (edge detection) 勾勒出原圖像的邊緣，提升模型的判斷能力。再來第二個部分，我們這次的模型都是以卷積層作為編碼器及解碼器，我們認為可以改為利用 Transformer 做為 encoder 或 decoder 架構，例如使用 Vision Transformer 作為編碼器架構。因為利用卷積層進行特徵提取的範圍受限於卷積核大小，對長程依賴 (Long-range dependence) 的捕捉能力較弱，而若使用 Transformer 的自注意力機制來替代傳統的卷積，較能夠捕捉長程依賴，能夠增強模型的泛化能力，在多樣化資料上表現更為穩定。

柒、程式碼

Github 連結：https://github.com/chsiang426/AICUP_GenerativeAI_II_2024

捌、使用的外部資源與參考文獻

[1] P. Isola, J. -Y. Zhu, T. Zhou and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

報告作者聯絡資料表

隊伍名稱	TEAM_5171	Private Leaderboard 成績	0.782478	Private Leaderboard 名次	9
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	張翔 Hsiang, Chang	國立政治大學 資訊科學系	National Chengchi University Department of Computer Science	0932-733-786	chsiang2002@gmail.com
隊員1	王睦閔 Mu-Hong, Wang	國立政治大學 資訊科學系	National Chengchi University Department of Computer Science	0900-201-216	109703029@g.nccu.edu.tw
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					

★註1：請確認上述資料與AI CUP報名系統中填寫之內容相同。自2023年起，獎狀製作將依據報名系統中填寫內容為準，有特殊狀況需修正者，請主動於報告繳交期限內來信moe.ai.ncu@gmail.com。報告繳交截止時間後將**不予修改**。

★註2：上傳程式碼檔案與報告至程式碼託管平台或是雲端等網站，將相關連結寄信至：jamesouo@g.nccu.edu.tw，並同時副本至：t_brain@trendmicro.com與moe.ai.ncu@gmail.com，缺一不可。