

# 自動駕駛實務期末專案報告

參與比賽：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



組別：25

組長：藍翊庭

組員：周韋恩、呂軒名、連思涵

Private leaderboard：131.1052 / Rank 14

指導老師：李政德、陳奇業

## 壹、環境

- 作業系統：Ubuntu 18.04.6 LTS
- GPU 硬體加速：NVIDIA Tesla V100 32GB
- CUDA Version：12.1
- 語言：Python 3.10
- 套件：
  1. Torch：深度學習計算框架
  2. Torchvision：提供常用的資料集、模型和影像轉換工具
  3. OpenCV (cv2)：處理和轉換影像數據
  4. Dominate：用於創建和操作 HTML 文檔的庫
  5. Visdom：監控模型訓練過程，可視化各種參數
  6. Wandb (Weights & Biases)：實驗管理和結果的可視化
- 預訓練模型：無
- 額外資料集：無

## 貳、演算法與模型架構

**演算法設計**：在這次競賽中，我們採用了 Pix2Pix 這種基於條件生成對抗網絡 (Conditional Generative Adversarial Network, cGAN) 的框架。Pix2Pix 專門用於圖像到圖像的轉換任務，能夠基於輸入的一張參考影像(如黑白草圖)生成相應的目標影像(如真實場景照片)。

該算法的核心思想是使用對抗訓練，讓生成器 (Generator) 與判別器 (Discriminator) 相互對抗，從而訓練生成器生成越來越逼真的合成影像。

1. 生成器：目標是產生逼真的合成影像以欺騙判別器。
  - 採用了編碼器-解碼器 (Encoder-Decoder) 的 U-Net 架構
  - 編碼器負責從輸入影像中提取特徵
  - 解碼器將編碼器提取的特徵轉換為目標域的影像
  - U-Net 中的跳躍連接 (Skip Connections) 能夠幫助保留高分辨率的特徵，增強生成影像的細節和質量
2. 判別器：被訓練來區分真實影像與生成器產生的合成影像。
  - 使用了 PatchGAN 架構
  - PatchGAN 通過對影像的局部 patches 進行真實/假影像的二分類，間接評估整個影像的真實性
  - 這種設計能夠很好地應用於高分辨率影像生成任務

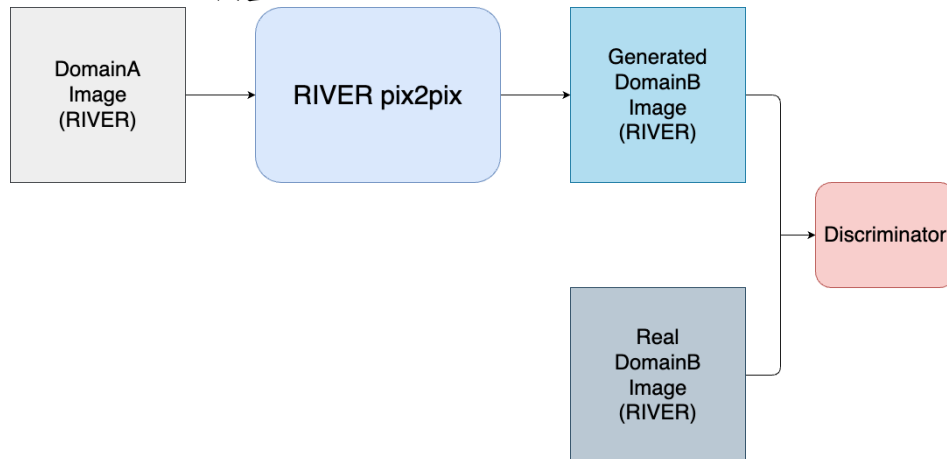
通過不斷的對抗訓練，生成器逐漸學會從參考影像生成高質量的目標影像，而判別器也變得更加敏銳。這種對抗性的訓練機制是算法的關鍵。

## 參、模型主要改良

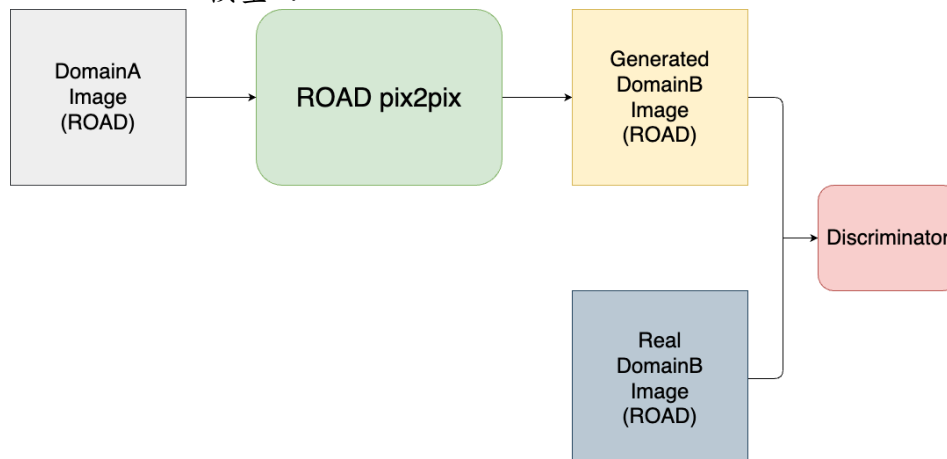
在本次競賽中，我們對基礎的 Pix2Pix 模型做了多方面的改良和創新：

- I. **資料前處理**：除了直接對原始資料集進行訓練，我們還對資料集進行了審視和篩選。具體來說，移除了一些模糊、品質極差的圖像, 避免這些低質量數據影響模型的訓練效果。
- II. **訓練策略**：與僅訓練一個通用的 Pix2Pix 模型不同，我們按照輸入草圖的類別(道路或河流)，分別訓練了兩個領域特定的模型 Road-Pix2Pix 和 River-Pix2Pix。這種分治策略能夠更好地捕捉不同領域的特徵模式，提高生成質量。
- III. **模型串聯**：我們將上述兩個領域特定模型串聯，構成一個複合模型。輸入草圖首先由類別識別模組判斷為道路或河流，再將其輸入對應的 Road-Pix2Pix 或 River-Pix2Pix 模型進行生成，實現了更靈活、準確的生成能力。以下是各個模型的架構圖。

**River-Pix2Pix 模型：**



**Road-Pix2Pix 模型：**



- IV. **超參數調優**：通過多次的實驗，我們找到了一組較為合適的超參數設置(如 Epoch、Batch Size 等)，此參數具體設置會在後面伍、訓練方式說明，使模型在生成質量和訓練效率之間取得了不錯的平衡。

除了上述的創新和優化措施，對於資料前處理的部分我們還有試過 Resize images 的部分，但成果都沒有比較好，反而在 FID Score 上的分數更不好，在之後肆、資料分析與處理過程的部分會詳細說明。

## 肆、資料分析與處理過程

## 資料集描述：

我們所使用的資料集具有以下結構和特點：

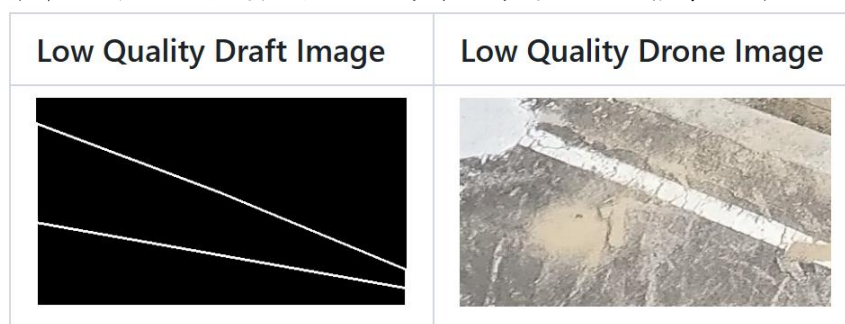
- I. 資料夾結構：
  - A. label\_img：包含黑白草圖，作為領域 A 的資料。
  - B. img：包含對應的實際空拍影像，作為領域 B 的資料。
- II. 數據總量：
  - A. 總共 4320 張影像。
- III. 分類：
  - A. 河流影像：
    1. 數量：2160 張。
  - B. 道路影像：
    1. 數量：2160 張。
- IV. 目標：
  - A. 利用這些資料進行影像轉換訓練，訓練模型將黑白草圖（label\_img 資料夾中的影像）準確轉換為逼真的空拍影像（img 資料夾中的影像）。

## 資料前處理：

在資料處理方面，我們進行了多步驟的資料前處理，以確保模型能夠學習到高質量的特徵。以下是詳細的資料前處理過程：

### I. 資料篩選(Data Filter)：

我們首先進行了資料篩選，刪除了模糊的影像。模糊影像可能會導致模型學到錯誤的模式，因此我們人工主觀判定哪些影像需要刪除，以保證資料集的質量。這個過程雖然費時，但我們認為這對於提升模型的準確性非常重要。以下是其中一個我們認為是模糊的影像示例：



- II. 資料集拆分：如上一節提到，我們訓練了兩個模型，一個訓練 ROAD datasets 而另一個用來訓練 RIVER datasets，所以我們需要將原資料集拆分成兩個不同的 datasets 以符合模型，透過觀察資料集的檔名來區分這兩種領域的 data。

而對於 testing datasets，我們一樣將他們通過檔名來區分並分成兩個 ROAD 跟 RIVER 的 data 來 reference 各自的模型。

### III. 資料大小調整(Resize images)：

考慮到模型對輸入圖像大小的要求，我們嘗試將原本的圖像大小統一調整為 256x256 像素。這樣的處理能夠確保模型在訓練時能夠更有效地學

習到影像特徵。然而，由於比賽時間即將結束，我們未能及時完成這一處理並提交評分。但後來我們進行了多次測試，發現統一調整影像大小為 256x256 像素後，模型的生成效果明顯提升。我們認為這主要是因為統一的影像大小能夠讓模型更好地學習整張圖特徵而不是每次隨機選取一部份來學習。

## 伍、訓練方式

我們採用的是分階段、循序漸進的訓練策略，具體分為以下幾個階段：

第一階段 - baseline model 訓練(200 epochs) 在這個階段，我們首先按照經典的 Pix2Pix 設定，訓練了一個通用的生成模型 baseline，將所有的訓練數據一股腦投入，不分領域類別。訓練參數為：

- n\_epochs = 400
- n\_epochs\_decay = 400
- Batch Size: 1

這一階段旨在為我們後續的改進策略打下基礎，初步了解數據的特徵和模型的表現。

第二階段 - 領域分治策略(200 epochs) 基於第一階段的基準模型，我們發現單一模型難以同時兼顧不同領域的特徵，生成效果並不理想。因此我們提出了領域分治的訓練策略：根據輸入影像是道路還是河流，分別獨立訓練兩個領域特定模型 Road-Pix2Pix 和 River-Pix2Pix。訓練參數為：

- n\_epochs = 400
- n\_epochs\_decay = 400
- Batch Size: 1

通過這種分治策略，兩個子模型能夠更專注於捕捉各自領域的特徵模式，生成質量顯著提升。

第三階段 - 模型融合與參數微調(800 epochs) 在獨立訓練了 Road-Pix2Pix 和 River-Pix2Pix 之後，我們將它們融合為一個複合生成模型。輸入影像首先經過分類判斷為道路或河流，再輸入對應的子模型生成輸出。

在融合後，我們進一步試調整訓練超參數，來進一步提升生成質量：

- n\_epochs = 600
- n\_epochs\_decay = 200
- Batch Size: 64

通過上調 Batch Size，我們能夠加快訓練過程，同時保持一個較大的學習率，避免陷入局部最優。然而實驗表明，較大的 Batch Size 對 GAN 模型的訓練並不十分友好，會降低生成質量。因此我們最終將 Batch Size 重新設置為 1，只保留了較長的 Epoch 數量。

第四階段 - 在訓練的最後階段，我們在多次的交叉訓練後，得到對於我們的模型的最好的參數設置，並發現 550 epoch 的 netG 有較好的效果，反而在 1000 epoch 時表現較差。訓練參數為：

- n\_epochs = 800
- n\_epochs\_decay = 200(最後取 550epoch 的 netG)
- Batch Size: 1



我們採用了循序漸進、由簡入深的訓練策略，從 baseline model 出發，不斷嘗試新的改進方法。訓練過程中，我們注重實驗和調參，想要找到一組最優的超參數組合，確保生成質量和訓練效率的平衡。

## 陸、結果呈現

生成的河流圖：



生成的道路圖：



## 柒、結果分析與結論

### 1. Epoch:

我們在 Training 時，考量到 train 越多層或許有較佳的效果，我們使用以下的 options，來進行兩個模型(Road\_pix2pix, River\_pix2pix)訓練。

- `n_epochs = 900`
- `n_epochs_decay = 100`

而在以往的 submission 中，發現 550 epoch 的 netG 有較好的效果，反而在 1000 epoch 時表現較差。

此結果暗示越多的 epoch 不一定有較佳的學習效果。

### 2. Data Filter:

我們在比賽接近尾聲時，刪除了較模糊的照片進行 training，並 training 1000 epoch。與之前的模型比較，刪除模糊照片後的模型有較好的表現。

### 3. Batch Size:

我們在比賽尾聲發現到了此 args，為了加快訓練速度，我們將預設的 `Batchsize=1` 改成 64，並發現效果非常差，在 Private Leaderboard 甚至分數

來到 200 多分。我們認為 GAN 模型在大 batchsize 的情況下 training 有較差的學習成果。

#### 4. 改善方法：

我們發現 pix2pix 作者與 NVIDIA 有合作，並提出了 pix2pixHD 架構 (<https://github.com/NVIDIA/pix2pixHD>)，但由於硬體限制，我們沒有充裕的時間可以做該實驗，但認為該 pix2pixHD 方法是有機會可以改善我們最佳的成績。

## 捌、未來工作

### 1. 資料集擴充：

- 收集更多類型的影像數據，包括不同天氣、季節和地理環境下的影像，以增強模型的泛化能力。
- 將資料集擴展到其他類別的影像，如城市、馬路等，進一步提升模型的應用範圍。

### 2. 模型改進：

- 探索並實驗更先進的生成對抗網絡架構，如 Pix2PixHD、CycleGAN 等，以進一步提升生成影像的質量。
- 考慮引入更多的輔助損失函數，如感知損失和風格損失，以提升生成影像的視覺品質。

### 3. 訓練策略優化：

- 實施混合精度訓練，以提升訓練效率，減少計算資源消耗。
- 探索自適應學習率調整方法，如學習率遞減和循環學習率，以提高模型收斂速度和穩定性。

### 4. 生成影像評估：

- 引入更多的評估指標，如結構相似性指數 (SSIM)、峰值信噪比 (PSNR) 等，進行更全面的評估。
- 結合主觀評估和客觀評估，確保生成影像在視覺效果和技術指標上均達到高水平。

### 5. 應用場景探索：

- 研究生成影像在實際應用中的可行性，如無人機導航、環境監測等。
- 探索模型在其他圖像翻譯任務中的應用潛力，如醫療影像分析、藝術風格轉換等。
- 自動駕駛：研究生成影像在自動駕駛中的應用，如用於訓練和測試自動駕駛車輛的感知系統，幫助車輛在不同環境下進行路徑規劃和決策。

透過這些未來工作，我們希望能夠進一步提升模型的性能和應用價值。

## 玖、學習收穫

這次參與生成式 AI 競賽讓我們獲得了豐富的學習經驗和寶貴的收穫。在過程中，我們面臨了多種挑戰，例如資料前處理階段需要篩選出高質量的訓練數據，這是一個非常費時且繁瑣的過程。我們人工判斷並刪除了許多模糊的影像，以確保模型能夠學習到有用的特徵。此外，在調整超參數和訓練策略時，我們發現較大的 Batch Size 對 GAN 模型訓練效果不佳。經過多次試驗和調整，我們最終確定了最佳的訓練參數，使得模型在生成質量和訓練效率之間達到平衡。

這次競賽也讓我們深刻體會到團隊合作的重要性。每個成員在不同環節中的專業知識和技能互補，使我們能夠順利完成各項任務。特別是在模型調整和優化過程中，通過團隊的共同努力和反覆測試，我們成功提升了模型的性能和穩定性。

整體來說，這次競賽不僅提高了我們在生成對抗網絡方面的技術能力，還增強了我們的問題解決能力和團隊合作精神。未來，我們將繼續探索生成式 AI 技術，應用於更多實際場景，為技術的發展和應用貢獻更多力量。

## 拾、程式碼

Github 連結：<https://github.com/Sherry2580/AI-cup-2024-spring>

按照 readme 的 Setup 中的步驟運行檔案中的所有 cells（已包含 requirement 在其中），如果想要達到跟 Private Leaderboard 上最好成績一樣的成果，運行 [preprocess\\_dataset.ipynb](#) 後直接運行 [test\\_model.ipynb](#) 即可，將會提供當時訓練出最好結果的 pre-trained model.

此外，這次比賽有與其他認識的隊伍 (TEAM\_5333) 進行討論與交換意見，code 有些許雷同但不盡相同，僅作為合作用途，合作隊伍的 Github Repo 如下：

LittleFish-Coder：<https://github.com/LittleFish-Coder/gen-ai-uav>

## 念、使用的外部資源與參考文獻

參考的文獻如下：

● Medium - 超解析度 GAN:

<https://chenabei0421.medium.com/%E8%B6%85%E8%A7%A3%E6%9E%90%E5%BA%A6gan-photo-realistic-single-image-super-resolution-using-a-gan-c792689ebe77>

● Medium - Conditional GAN:

[https://tomohiroliu22.medium.com/深度學習\\_paper\\_系列-08-conditional-gan-ea75376580a6](https://tomohiroliu22.medium.com/深度學習_paper_系列-08-conditional-gan-ea75376580a6)

● Medium - Pix2Pix：<https://ithelp.ithome.com.tw/articles/10326287?sc=rss.iron>

● CSDN - GAN Loss: <https://blog.csdn.net/morgan777/article/details/107267075>



- Stack Overflow:

<https://stackoverflow.com/questions/273946/how-do-i-resize-an-image-using-pil-and-maintain-its-aspect-ratio>

在本次 AI-cup 比賽使用的程式碼皆沒有使用如 chatgpt 等生成式語言模型，也沒有使用生成式語言模型來做資料擴增，但我們在撰寫 code 的時候有使用 Microsoft copilot 作為輔助，在撰寫書面報告時有時用 chatgpt 輔助我們完成。