**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\_5137

隊員：吳家萱（隊長）、許中萁、吳建中。

Private leaderboard： 0.722013 / Rank 17

**壹、環境**

作業系統：Windows11。

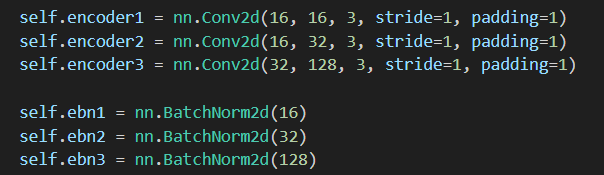
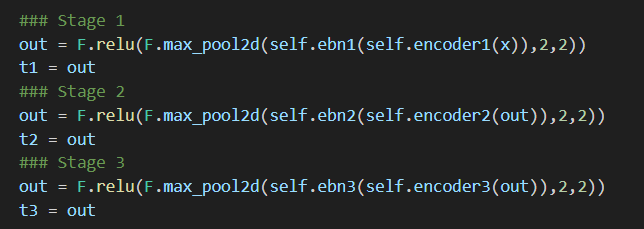
程式語言：Python 3.10.4。

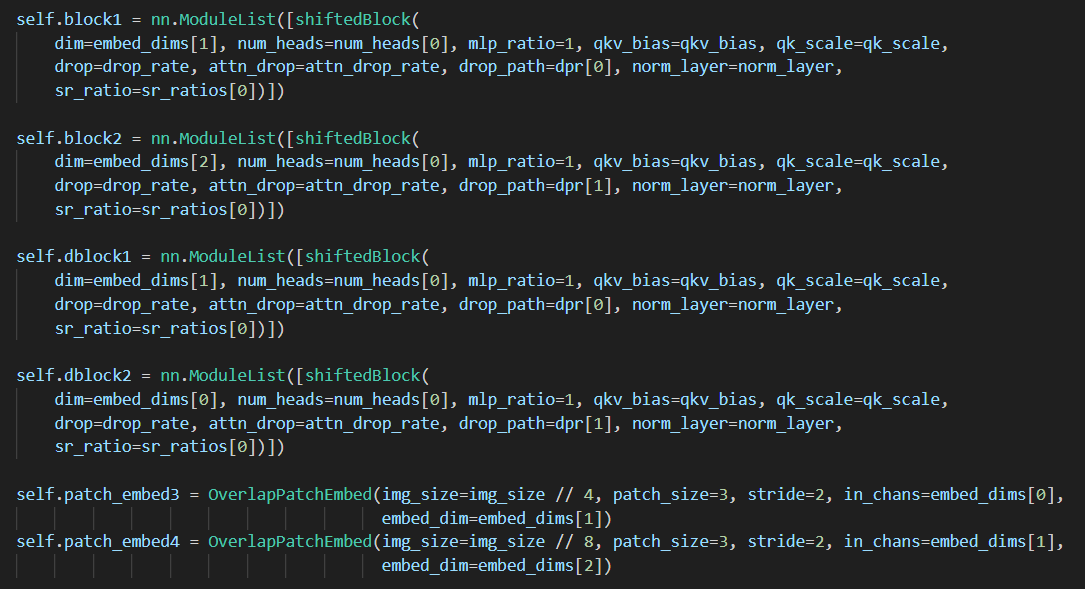
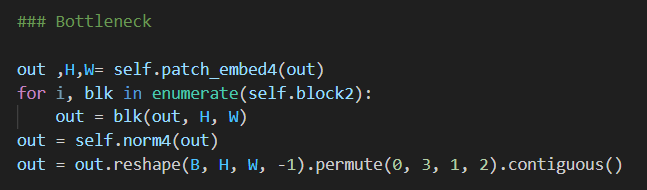
套件（函式庫）：PyTorch 2.3.0、numpy、cv2、os、torchvision、timm、matplotlib、math。

程式透過CUDA 11.8使用GPU訓練模型，無使用預訓練模型及額外資料集。

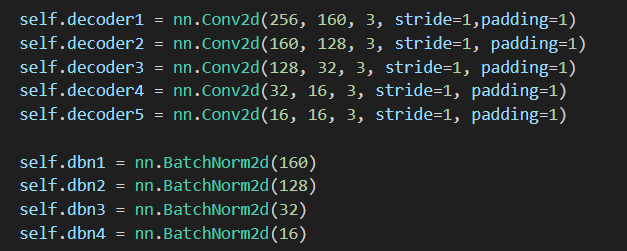
**貳、演算法與模型架構**

我們設計了一個改良UNeXt的模型架構，該模型結構為：Encoder、Bottleneck、Decoder，而我們額外在末尾增加了residual network來幫助模型訓練以更擬合本次比賽的dataset。

1. Encoder:  
   Encoder由三個捲積層組成，結構如下:    
   每次forward都會對每一層捲積層做BatchNormalization，透過max pooling將輸入的圖像分成多個子區域，對每個子區域輸出特徵最大值，ReLU激活函數。  
   
2. Bottleneck  
   使用Overlap Patch Embed將圖像特徵分割成重疊的patch，通過捲積進行嵌入（從而保留圖片邊緣的資訊，提高分割的精度），然後提供給shifted MLP結合卷積操作和MLP，將特徵圖沿著不同的方向進行平移，然後拼接，可以更有效地捕捉特徵。  
   Overlap Patch Embed結構如下：

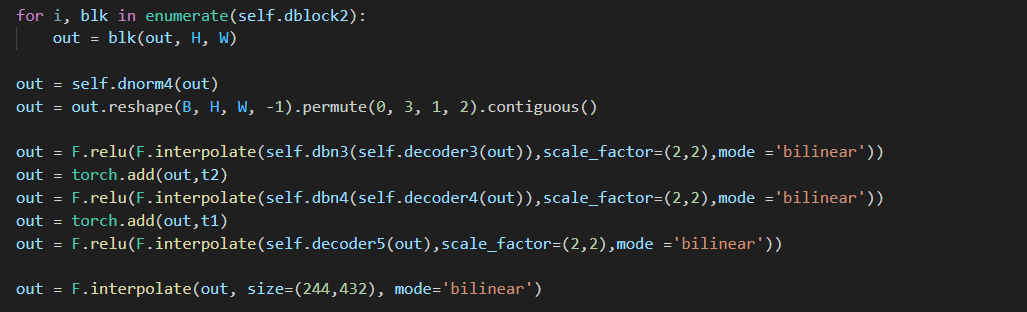
  
Bottleneck寫法如下，使用上述的self.patch\_embed4來做Overlap Patch Embed：

1. Decoder:

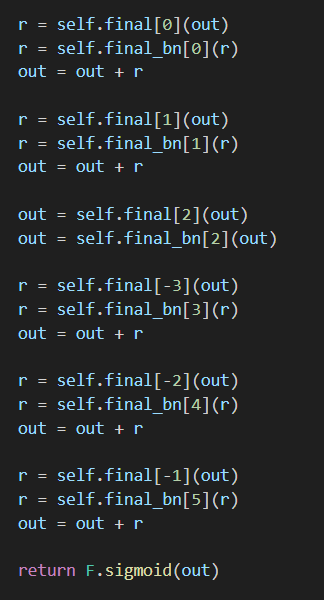
Decoder由多層卷積組成，逐層進行Up-sampling以恢復原特徵圖的尺寸，結構如下：  


每次forward都會對每一層捲積層做BatchNormalization，透過F.interpolate做上採樣，改變數據的大小，使用的是bilinear採樣法。

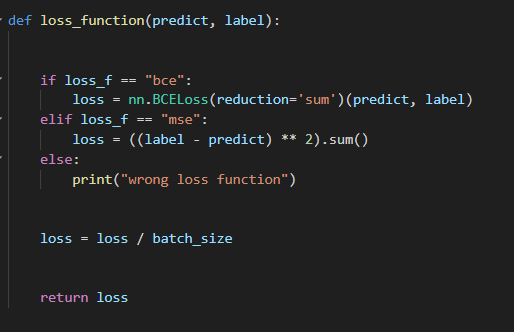




1. residual network  
   輸出前我們透過自己設計的residual network，最後使用sigmoid函數進行激活，輸出分割後的圖像，結構如下：

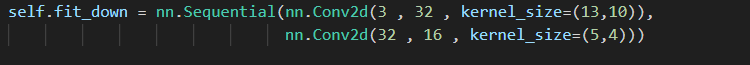


我們嘗試於模型中使用的loss為bce loss和mse loss，最終嘗試使用為mce loss，寫法如下所示：

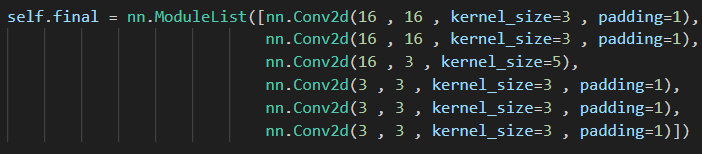


**參、技術模型原創性或改良成效**

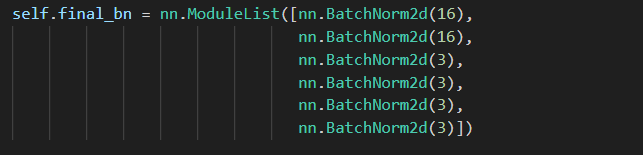
針對原240x428的圖片，UNeXt總共做了五次maxpooling，我們在forward的一開始時增加了一個self.fit\_down，使得240透過卷積運算轉換成224，而428轉換成416，使兩者質因數分解後皆有至少五個2為因數，如下所示：



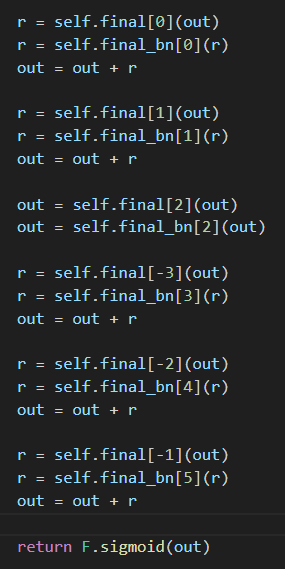
以及我們在原有的UneXt模型多設計了residual network，並用sigmoid激活函數，目的是為了讓模型更加擬合本次比賽的datast，結構如下：



我們對self.final做了五層捲積，將原始fit down的16 out channel降至3 channel，為了使padding後不改變channel大小，因此採kernel size=3, padding=1的結構，再將Decoder forward後的輸出大小(244, 432)透過nn.Conv2d(16 , 3 , kernel\_size=5)，使其大小剛好轉為(240, 428)。



我們的residual network透過self.final和self.final\_bn實現，結構如下：



在這次比賽過程中，我們有試驗過以下結構，配合residual network的self.final只有兩三層捲積，但效果不彰，public test dataset準確率僅約為0.68：

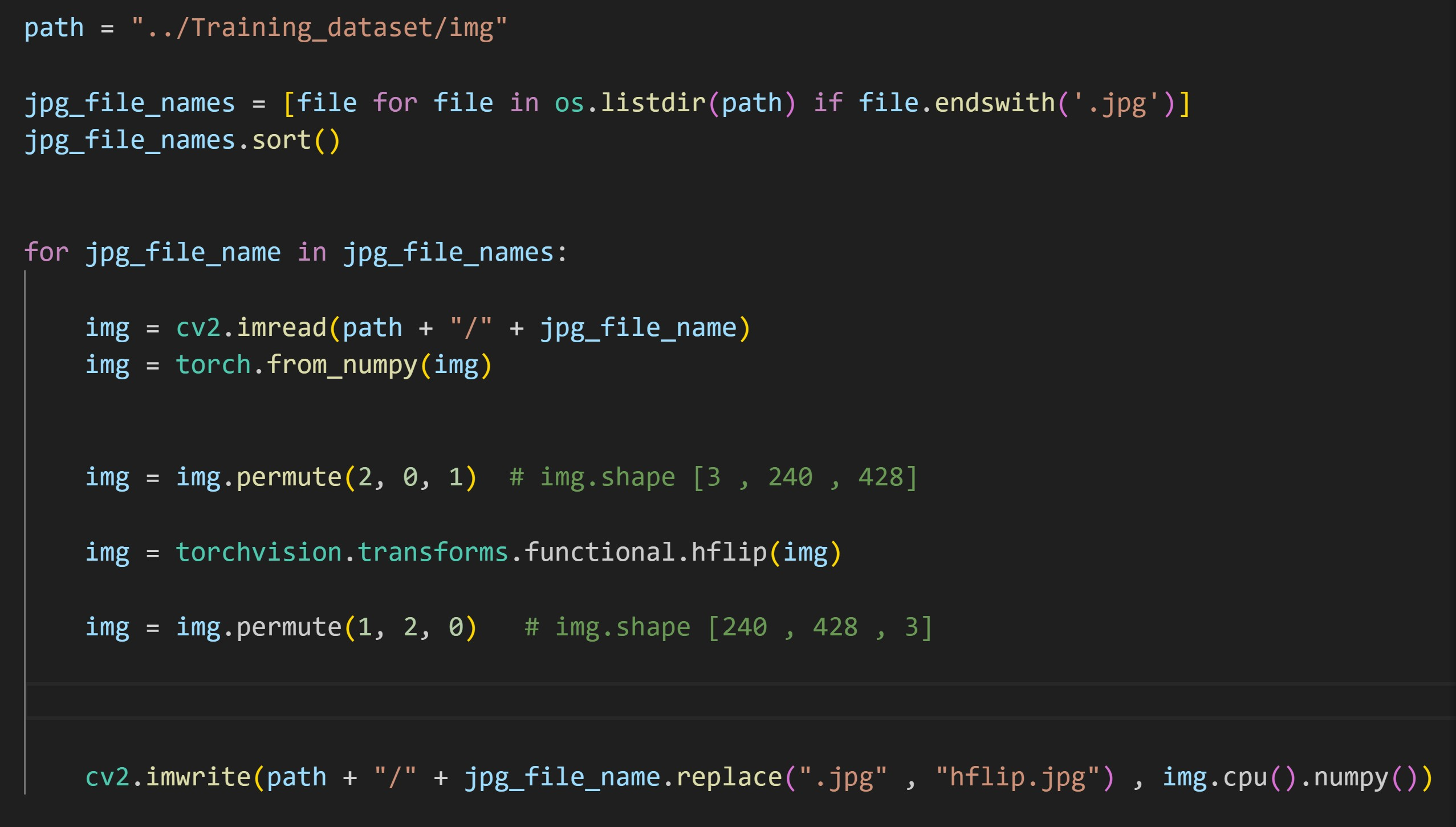


**肆、資料分析與處理過程**

我們有對訓練資料集做水平翻轉擴增，使用我們撰寫的data\_flip.py，如下：  
img.permute()為對tensor進行維度換位，轉換後的img.shape為 [3, 240, 428]

對img做水平翻轉，使用torchvision.transforms.functional.hflip(img)

水平翻轉完再將其用img.permute()轉換為原始圖片維度，即 [240, 428, 3]。



在Training dataset讀取時，轉換成numpy型態，接著用unsqueeze做擴維，變成[1, 240, 428, 3]，因為跑了batch size次 torch.cat((data, img), dim=0)，所以最後的data.shape變成[batch\_size, 240, 428, 3]，最後做data.permute(0, 3, 1, 2)使其最後data.shape變成[batch\_size, 3, 240, 428]。

中間以data=data/255做資料預處理的標準化。

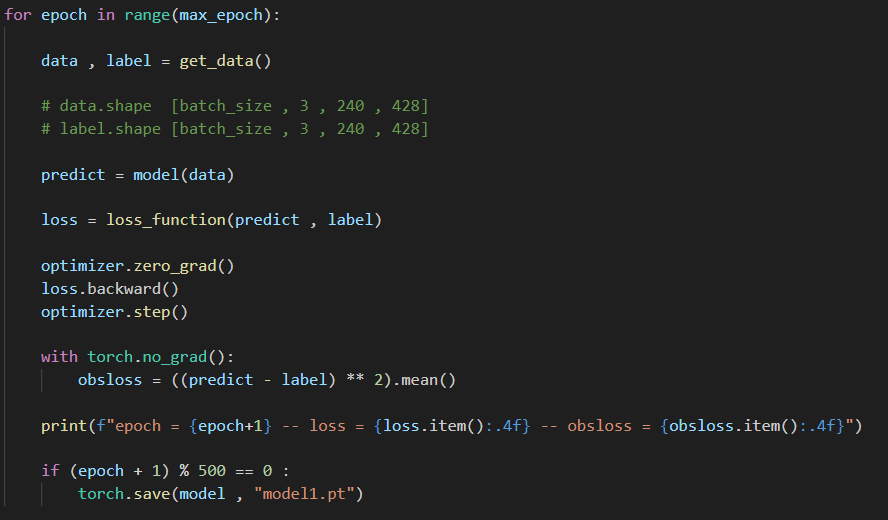






**伍、訓練方式**

模型訓練方法與過程步驟如下：

1. 數據加載  
   使用我們自己撰寫的get\_data.py來讀取dataset的img和label，在下載的過程中會同時shuffle，讓每次模型訓練都取隨機排列的資料，可以避免模型overfitting。
2. 模型超參數配置  
   使用我們自己撰寫的hyper\_parameter.py來設計超參數，包含epoch、batch size、learning rate、loss函數（我們使用了兩種，分別為bce loss和mse loss，最終使用mse loss），我們的optimizer在訓練的前期和後期使用torch.optim.Adam，中期使用torch.optim.Adam.SGD。
3. 模型設置  
   使用我們的改良UNeXt模型架構，結構如上文演算法與模型架構中描述。
4. 損失函數  
   我們使用mse loss當作我們的損失函數。
5. 模型訓練梯度更新  
   在main.py中，我們訓練模型的方法如下，每次計算loss後清除梯度optimizer.zero\_grad()，然後進行反向傳播loss.backward()，並更新模型參數optimizer.step()，我們選擇每500個世代保存一次模型torch.save(model, "model2.pt")。  
   
6. 測試資料  
   使用我們自己撰寫的predict\_public.py和predict\_private.py，將儲存的模型model2.pt讀取並用於public test dataset和private test dataset，生成出道路導航路線圖，以predict\_public.py為例，程式寫法如下。  
   

**陸、結果分析與結論**

我們最終修改的模型成效如下所示，可以觀察到，當為直線或明顯可觀察之道路/河道時，辨識生成效果較為良好：

|  |  |
| --- | --- |
|  |  |
| PUB\_RO.jpg | PUB\_RO.png |

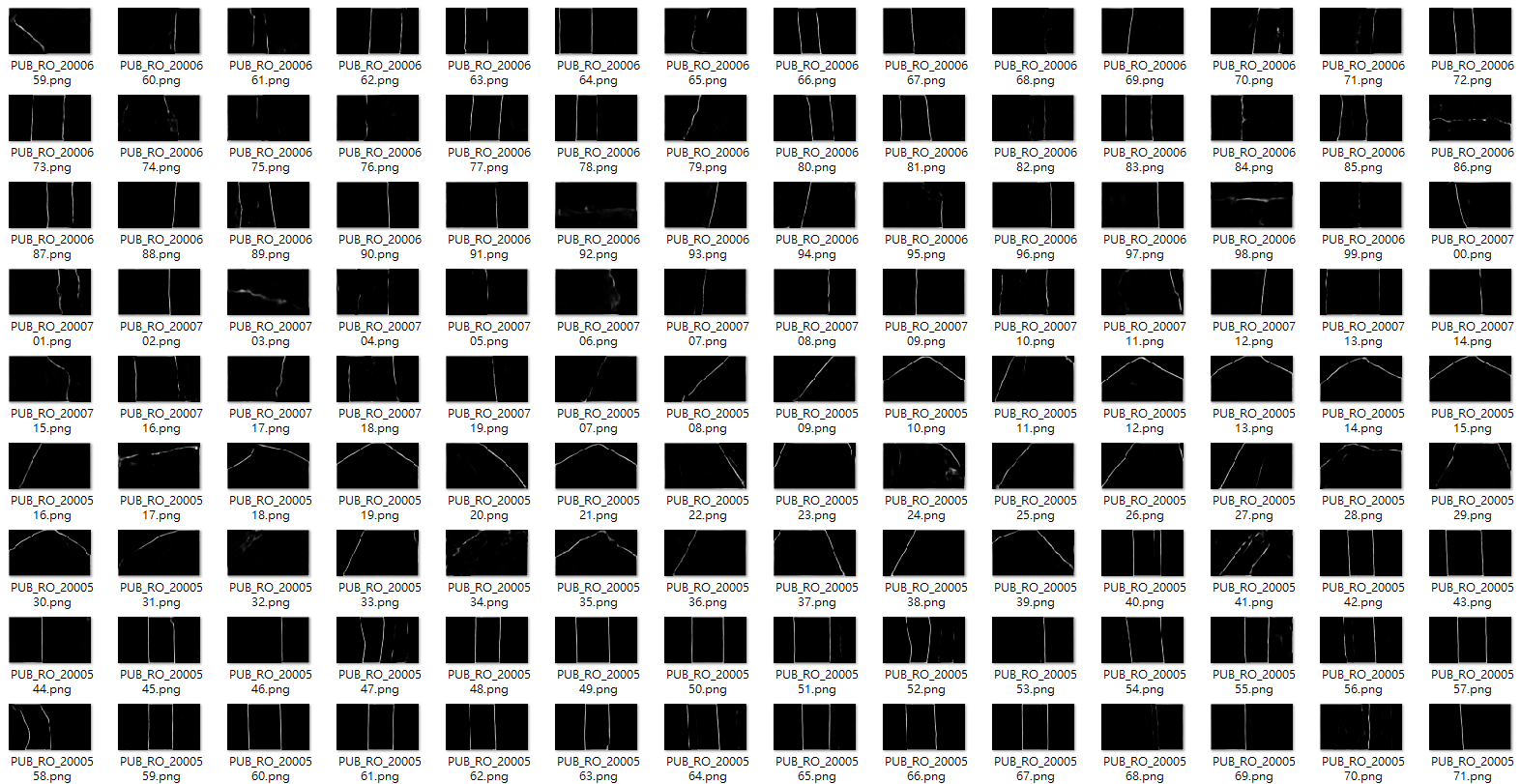
然而，當道路邊緣被遮擋而無法輕易辨識時，我們模型的辨識生成效果便不彰，如下所示：

|  |  |
| --- | --- |
|  |  |
| PUB\_RI\_2000076.jpg | PUB\_RI\_2000076.png |
|  |  |
| PUB\_RO\_2000423.jpg | PUB\_RO\_2000423.png |

未來我們可能透過其他更新的論文中的模型技術，來加強我們的模型學習，嘿是對UneXt再做更多的調整修改（譬如修改不同層數的residual network等…etc）。

我們在比賽進行過程中，有嘗試過不添加residual network的版本，然而，在public test準確率上僅有0.68，所以我們認為，嘗試不同層數的residual network有助於模型擬合本次比賽的資料集。

0.68的生成大致如下，顯然在某些道路邊緣較無法良好的進行辨識：



**柒、程式碼**

Github連結：<https://github.com/lavinia0724/AI-CUP-Navigational-Data-Generation-with-Generative-AI>

**捌、使用的外部資源與參考文獻**

Jeya Maria Jose Valanarasu, Vishal M. Patel, 2022, *UNeXt: MLP-based Rapid Medical Image Segmentation Network.* MICCAI

**報告作者聯絡資料表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 隊伍名稱 | TEAM\_5137 | Private Leaderboard 成績 | 0.722013 | Private Leaderboard 名次 | Rank 17 |
| 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 | 吳家萱  Chia－Hsuan,  Wu | 天主教輔仁大學  資訊工程學系 | Fu Jen Catholic University Department of Computer Science and Information Engineering | 0916921480 | lavi＠nisra.net |
| 隊員1 | 許中萁  Chung－Chi,  Hsu | 天主教輔仁大學  資訊工程學系 | Fu Jen Catholic University Department of Computer Science and Information Engineering | 0966702318 | f0978473376@gmail.com |
| 隊員2 | 吳建中  Jian－Zhong,  Wu | 天主教輔仁大學  資訊工程學系 | Fu Jen Catholic University Department of Computer Science and Information Engineering | 0932101121 | andywu10523@gmail.com |
| 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 | 黃貞瑛 | Jen－Ing, Hwang | 天主教輔仁大學  資訊工程學系 | Fu Jen Catholic University Department of Computer Science and Information Engineering | jihwang  @csie.fju.edu.tw |
| 教授 2 |  |  |  |  |  |