

目錄

壹、 前言	1
一、 研究動機	1
二、 研究目的	1
貳、 研究過程與方法	1
一、 文獻探討	1
(一) 物理侵蝕模型	1
(二) pix2pix 模型	3
(三) VAE (Variational Autoencoder)	4
二、 收集訓練模型所需之圖像資料	5
(一) 地形高度圖	5
(二) 衛星空照圖	6
三、 pix2pix 模型的訓練方式	7
四、 生成訓練資料集	7
(一) 生成貼圖模型訓練資料集	7
(二) 生成地形擬真模型訓練資料集	8
五、 本研究建構的模型結構—VAE-pix2pix	9
(一) 貼圖模型	9
(二) 地形擬真模型	10
六、 訓練貼圖模型	11
七、 訓練 VAE-pix2pix 模型	11
八、 訓練地形擬真模型	11
(一) 對 pix2pix 模型進行調整	11
(二) 訓練 pix2pix 結構之地形擬真模型	12
(三) 訓練本研究模型結構之地形擬真模型	12
參、 研究結果與討論	13
一、 評斷生成對抗網路模型的方法	13
(一) L1 Loss	13
(二) L2 Loss	13
(三) Perceptual Loss[11]	13
(四) FID (Fréchet Inception Distance)[2]	14

(五) SSIM index (structural similarity index)[10]	14
(六) 生成每張圖像所需的平均時間	15
二、貼圖模型之訓練結果	15
三、地形擬真模型之訓練結果	15
四、將 VAE-pix2pix 的 latent code 進行 PCA (主成分分析)	15
五、建構模型的 API 伺服器	18
六、Unity 用戶端	18
七、由用戶端調整 latent code 對於輸出風格的效果	18
八、透過改變 style encoder 的輸入改變輸出圖像的風格	20
九、討論物理侵蝕模型、pix2pix 模型及 VAE-pix2pix 之間的差異	20
肆、結論與應用	20
A VAE-pix2pix 的模型結構	25
B 高度圖及空照圖的具體收集範圍	25
一、中國橫斷山脈	25
二、喜馬拉雅山脈	26
三、祕魯安地斯山脈	27
四、阿根廷冰河	27
五、加拿大冰河	28

Abstract

In this study, we use NASA's SRTM 1 Arc-Second dataset to collect altitude maps from around the world, and we also use MapTiler API to collect corresponding satellite images. We use these collected images to train our VAE-pix2pix model, which is a Variational Autoencoder (VAE) combined with pix2pix (a Conditional Generative Adversarial Network). VAE-pix2pix can add details of the real mountain should have (including sharp ridges, mountain wall textures, continuous river networks, etc.) to the heightmap which is drawn by users. Our model can generate the corresponding satellite images as well. Compared with the original pix2pix model, our model can generate heightmap and satellite images that are more realistic. Furthermore, our model can also generate different styles of heightmap and satellite images by changing the value of the latent code, such as the color of the landform or the height of the snow line, etc., which increases the diversity of the images generated by the model. To make our model can be better used, we have developed a client on Unity, which can generate a mesh that allows users to directly use it when developing the game in Unity. In conclusion, our work has simplified the task of generating a realistic mountain model in-game or other fields as well.

摘要

本研究利用 NASA 的 SRTM 1 Arc-Second 資料集 [3] 來收集全球各地的地形高度圖 (heightmap)，我們也利用了 MapTiler 網站收集了相對應的衛星空照圖。利用這些收集的圖像，訓練我們自行研究的 VAE-pix2pix 模型。VAE-pix2pix 為 Variational Autoencoder (VAE) 及 pix2pix (為一個 Conditional Generative Adversarial Network) 結合的模型，能將人工繪製的高度圖加上真實山脈應有的細節 (包含尖銳的山脊、山壁上的紋路、連續的河流網路等……)，也能生成出相對應的擬真衛星空照圖。相較於原 pix2pix 模型，VAE-pix2pix 所生成的高度圖及空照圖會更接近於真實世界的山脈高度圖及空照圖，同時 VAE-pix2pix 模型也可以透過改變 latent code 的數值來生成出不同風格的高度圖及空照圖，如地貌的顏色或雪線的高度等，這些都增加模型生成圖像的多樣性。為了使我們建構的模型能廣泛的被應用，我們在 Unity 上開發了 Unity 客戶端，其生成的 mesh 可以讓使用者直接應用於遊戲的場景，簡化了生成擬真山脈模型的任務。

壹、前言

一、研究動機

隨著 3C 的普及，遊戲已經成為現代人打發時間、舒壓及社交的必需品；隨著科技技術的進步，對於遊戲畫質的要求也越高，而在製作各種遊戲時，常常會需要生成擬真的地形作為場景。

傳統上，遊戲的擬真的山脈地形是透過人工繪製。在將大致的架構畫出來後，還需花費不少時間捏出山脊和挖出河流等細節部分。而近年來，人工智能的演算法在圖像的生成上面有重大的突破，不論是憑空產生圖片或是將影像的風格提取出來，並轉換到另一張影像，都已經是可行的方法，因此本研究希望簡化人工繪製的過程，透過訓練生成對抗網路來達到生成擬真山脈地形的成果。

二、研究目的

本研究期望能簡化遊戲製作者在生成擬真的山脈地形的流程，同時確保生成擬真山脈的效果。研究參考遊戲地圖製作流程，是先產生高度圖(山脈、河流等地形特徵)再把衛星空照圖(植被、河流等地貌景觀)作為貼圖紋理貼到高度圖模型上。本研究使用兩個模型產生完整的擬真的山脈地形：

- 地形擬真模型能有效地將人工設計的山脈架構高度圖自動轉換成擬真山脈地形高度圖。
- 貼圖模型則依地形擬真模型生成出的擬真高度圖，生成相對應的空照圖，作為擬真高度圖 3D 渲染時的貼圖紋理。

貳、研究過程與方法

一、文獻探討

(一) 物理侵蝕模型

一般來說，若要提升遊戲中山脈地形的真實度，其中一種方法為使用侵蝕的物理模型，如論文 [6]，建構一個物理模型，模擬水流在地形上侵蝕與堆積，論文作者藉由此模型結合 GPU，實現改變地形樣貌的效果。

根據論文 [6] 的內所敘述的物理模型，本研究利用 PyTorch 實現論文 [6] 所述之物理侵蝕模型，之後會將其與經訓練的 VAE-pix2pix 模型進行真實度及實用性的比較。

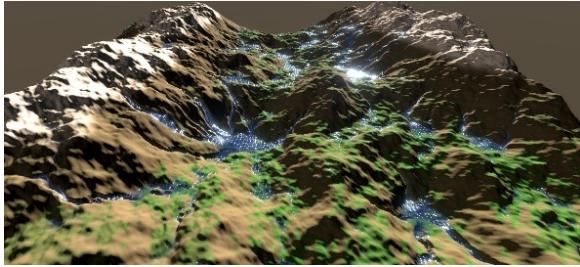
此物理侵蝕模型將地表分成正方形的網格，使用歐拉法求地表上每格的水深、含沙量和網格間的水流速，並根據水量和流速進行侵蝕和堆積，疊代多次後可得出侵蝕一段時間後的地表面和水面高度圖。一次疊代的步驟如下：

1. 在每個網格加上等量的水，模擬均勻的降雨
2. 更新流速 (加速度受坡度和阻力影響)
3. 根據流速讓水流到鄰近的格子，同時搬運等比例的砂土
4. 根據水量和流速進行侵蝕，增加水中含沙量，降低地面高度
5. 將水中一定比例的沙土堆積到地面
6. 移除每格一定比例的水，模擬蒸發

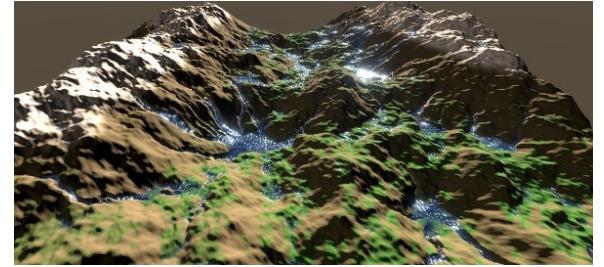
表 1: 侵蝕模型的參數

參數	說明	數值
KRain	單位時間、面積的雨量	0.05
A	水流速的乘數	1
KS	侵蝕速率	30
drag	水受到的阻力	0.01
deltaT	一次疊代的時間步長	0.01
KD	水中每單位時間沉澱的沙土 比例	0.002
KE	蒸發速率	0.0002

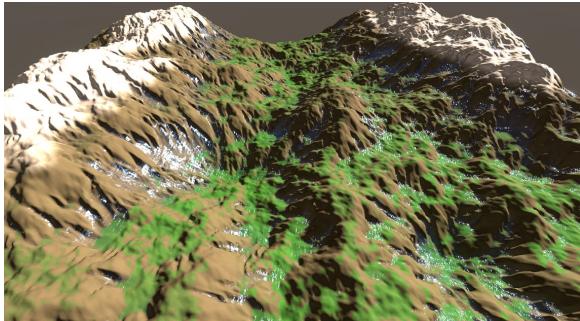
調整表 1 的變數能改變侵蝕後地形的風格。為了使用戶可以調整河流的連續性，本研究在論文 [6] 的基礎上加入 drag 參數。而當 drag 越高時，河流越不連續，執行結果圖 1 與圖 2。



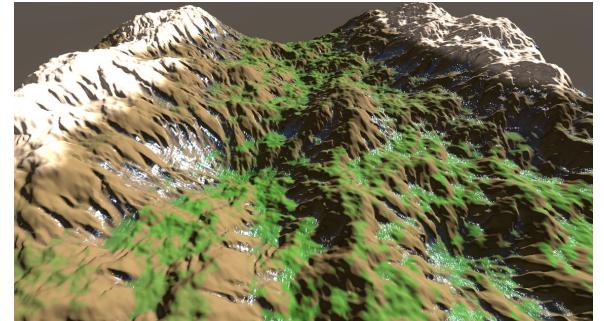
(a)



(b)

圖 1: drag 的值對河流連續性的影響。(a) $\text{drag} = 0$ 、(b) $\text{drag} = 0.2$ 

(a)



(b)

圖 2: 參數 A 的不同會影響所產生的地形風格。(a) $A = 0.1$ 、(b) $A = 1$

(二) pix2pix 模型

論文 [5] 則為 pix2pix，是一個 Conditional adversarial networks(又稱 Conditional GAN)，其訓練時將一對影像當作輸入，而模型的目標則是將第一張圖片轉換為第二張圖片。例如圖 3，模型的目標是將左圖做為輸入，輸出右邊的圖像。

pix2pix 由 generator 和 discriminator 兩個部分組成。generator 的結構為 U-Net，訓練時會嘗試把輸入圖像轉換為目標圖像。discriminator 則是一個分類器，訓練時會嘗試分辨哪些圖是 generator 生成的圖，哪些是真的目標圖像。兩者會同時訓練，generator 生成的圖越不容易被 discriminator 分辨出來，就代表 generator 表現得越好。利用這點來訓練 generator，就能讓它的輸出盡可能的真實，在此研究中，我們要利用的就是訓練好的 generator 來生成具有山脈細節的灰階高度圖。

如圖 4 中，discriminator 的任務是辨認出哪些圖片是由 generator 所生成(如左)，哪些是原始圖像(如右)。

此外，一般的 generator 都是使用 Encoder-decoder 結構，而 pix2pix 模型的 generator 則使用了特殊的 U-Net 結構，其為 Encoder-decoder 結構的改良，如圖五。其在不同層之間加上了

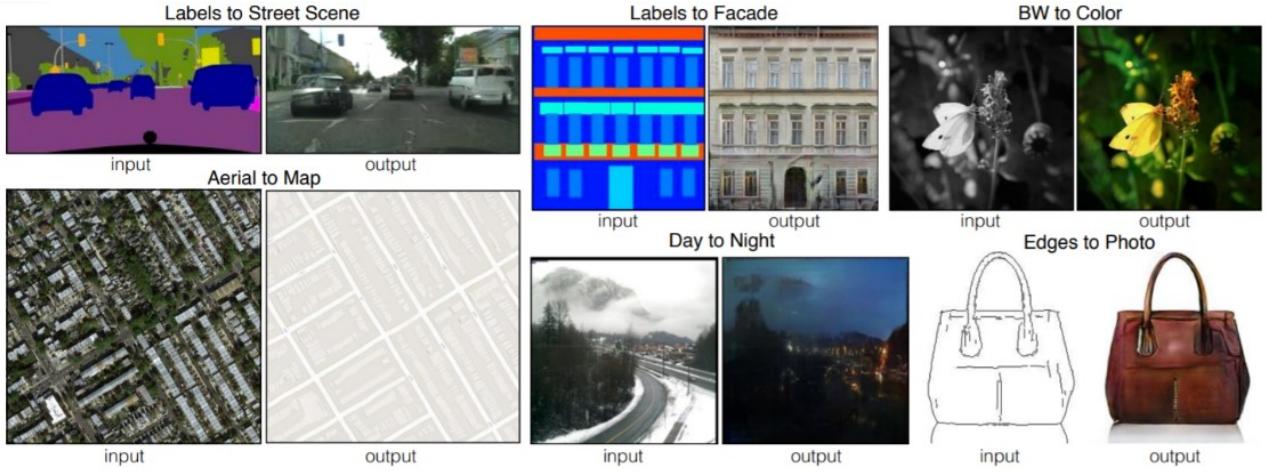


圖 3: pix2pix 可做到的圖像風格轉換 (取自 [5])

跳躍連接 (skip-connections)。而 U-Net 在圖像分割任務上表現十分良好。我們的研究將使用 pix2pix 作為基礎模型。

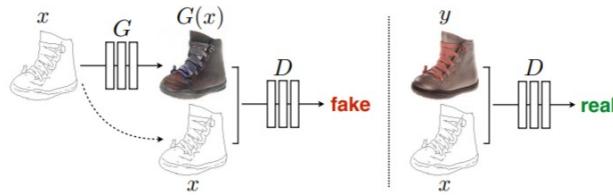


圖 4: 訓練 Conditional GAN 將鞋子的邊緣圖生成實際鞋子的圖像。(取自 [5])

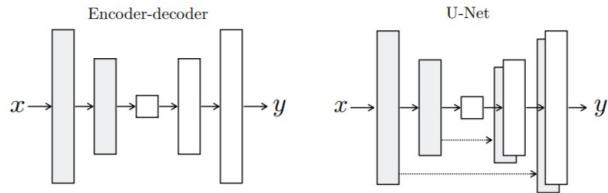


圖 5: Encoder-decoder 與 U-Net 結構比較 (取自 [5])

(三) VAE (Variational Autoencoder)

Autoencoder 模型為 Encoder-decoder 結構。運作時，輸入圖片 x 會被 encoder 編碼成 latent code，再由 decoder 依照 latent code 的資訊嘗試還原出 x 。Latent code 為整個模型結構的瓶頸，所以 encoder 的目標是把輸入 x 以最少資訊損失的方式壓縮成維數相對很小的 latent code，以供 decoder 使用。也就是說，encoder 做的是非線性降維，它會萃取輸入圖片的高階特徵。

而 VAE(Variational Autoencoder)[7] 類似 Autoencoder，但其中 encoder 的輸出為平均及標準差，這兩個參數代表著一個高斯(常態)分布，訓練時 latent code 會從該分布中隨機取出，傳給 decoder。且 latent code 的先驗分布會被 KL divergence 限制在標準高斯分布內，這些限制使 VAE 能學到更有意義的 latent space，也方便應用。

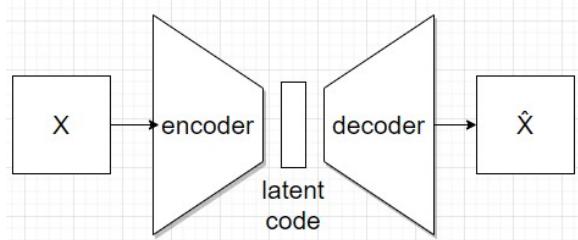


圖 6: VAE 的基本模型結構

生成擬真的人臉圖像即為 VAE 典型的應用，而 VAE 學習到的 latent space 中，各維度的意義可能是臉的方向、膚色、頭髮長度或眼睛大小。

我們的研究將以 VAE 與 pix2pix 結合，成為一種新的類神經網路架構，接著會利用這個類神經網路來訓練地形擬真模型和貼圖模型，最後會與基礎的 pix2pix 模型與物理侵蝕模型進行效果的比較。

二、收集訓練模型所需之圖像資料

本研究主要會收集五個地區的地形高度圖及空照圖。這個五個地區分別為橫斷山脈、喜馬拉雅山、祕魯安地斯山脈、阿根廷及加拿大的冰河地形。透過收集不同區域的地形，使 VAE-pix2pix 能學到不同地區的高度圖及空照圖的特徵。

(一) 地形高度圖

地形高度圖的資料來源為 NASA 的 SRTM 1 Arc-Second 資料集。此資料集將 1 經/緯度範圍的高度資料儲存為一張高度圖，每張高度圖的編號方式是按照其高度圖左下角的座標作為檔名，若此張高度圖的收集範圍為 $25^{\circ}\text{N}, 98^{\circ}\text{E}$ 、 $25^{\circ}\text{N}, 99^{\circ}\text{E}$ 、 $24^{\circ}\text{N}, 99^{\circ}\text{E}$ 、 $24^{\circ}\text{N}, 98^{\circ}\text{E}$ 四個座標點所圍成的範圍，則此張高度圖的檔名即為 N24E98，我們在附錄中會使用這種方式來表示各地區的收集範圍。

因為 SRTM 資料集採用特殊的 HGT 格式，不能使用一般的圖像軟體讀取，也使得生成訓練資料集的工作變得麻煩，所以我們利用 gmalthgtparser 來將 HGT 格式轉為可以直接以圖像軟體讀取的 PNG 格式。gmalthgtparser 為一個 Python module，可以讀取 HGT 檔案中特定

地點的高度值(單位為公尺)，讀取到高度值後，我們利用式 1 將高度值轉為 RGB 值。

$$(R, G, B) = \left(\left\lfloor \frac{\text{height}}{256^2} \right\rfloor, \left\lfloor \frac{\text{height \%}256^2}{256^1} \right\rfloor, \left\lfloor \frac{\text{height \%}256}{1} \right\rfloor \right) \quad (1)$$

對高度值進行轉換後，我們即可以將一張 HGT 檔案的高度圖轉換為方便易用的 PNG 高度圖，轉換後的高度圖如圖 7。

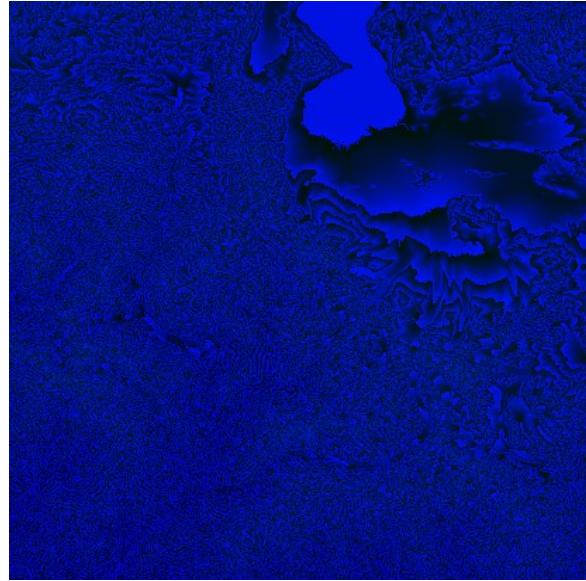


圖 7：由 hgt 檔案轉換的 PNG 圖檔

(二) 衛星空照圖

本研究中，地形高度圖需要與衛星空照圖相互對應，所以使用 MapTiler 網站所提供的 XYZ tiles map 來收集衛星空照圖。XYZ tiles map 是一種儲存地圖資料的方式，其方式為將大圖切割成許多張小圖，可以使地圖加載的速度變快，亦可節省網路資源。我們利用 MapTiler 所提供的衛星空照圖 tiles map 服務，收集橫斷山脈範圍內的多張衛星空照圖，再以 EPSG:4326 (WGS 84) 座標系統將各張小圖 (tiles) 組合成一張與地形高度圖互相對應的衛星空照圖。

因為衛星空照圖需與地形高度圖相互對應，所以空照圖的收集數量要與高度圖相同。表 2 為收集五個地區的高度圖及空照圖數量，具體的收集範圍列於附錄 A：

表 2: 五個地區的高度圖及空照圖收集總數 (單位：張)

地區	高度圖數量	空照圖數量
橫斷山脈	16	16
喜馬拉雅山	10	10
祕魯安地斯山	15	15
阿根廷冰河地形	9	9
加拿大冰河地形	5	5

三、 pix2pix 模型的訓練方式

pix2pix 模型之訓練方式是輸入一對照片，其中一張照片為輸入，另一張照片則為目標輸出。pix2pix 模型的目標即是將模型輸出盡可能接近目標輸出。在訓練過程中，我們將訓練圖像集分為三個部分，分別為 train、val 及 test，train 是用於訓練模型，val 則是在訓練過程中驗證模型的正確性，test 則是在訓練完成後，用於測試模型。

四、 生成訓練資料集

我們將收集的高度圖和空照圖分為 train、val 及 test 三個區域 (如圖 8)，並分別從這三個區域中切割出訓練資料集的圖像，這樣可以使三個部分的圖像不重複，以測試模型的精準度。

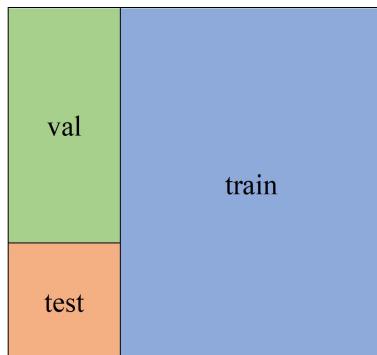


圖 8: train、val、test 在大圖的位置

(一) 生成貼圖模型訓練資料集

貼圖模型的目標是將輸入的空照圖生成相對應的高度圖，所以我們的訓練資料對即是空照圖及高度圖，其中空照圖為輸入 (input)，高度圖為目標輸出 (ground truth)。

將圖片分割成上述的三個區域後，我們隨機在高度圖與空照圖上切割出多個 256×256 大小的圖像，並將高度圖與空照圖拼接在一起，形成一個訓練資料對，如圖 9。然後將五個地區的空照圖與高度圖一起放在同一個資料集，透過我們的模型訓練，學習到這五個地區的高度圖及空照圖中不同的特徵。

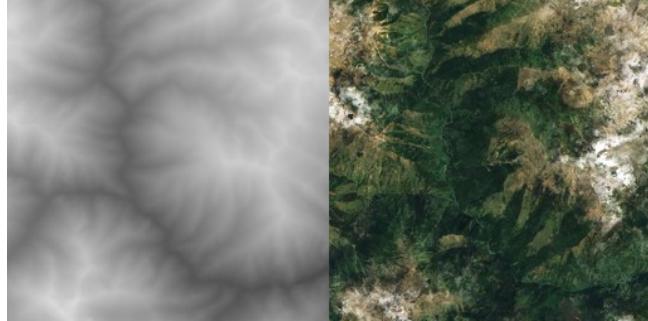


圖 9: 貼圖模型之訓練資料對

五個地區在 train、val、test 三個資料集中所佔的圖像數量皆相同，train 為 512 張、val 為 128 張、test 為 80 張。所以整個資料集的 train、val、test 張數分別為 2560 張、640 張、400 張。

(二) 生成地形擬真模型訓練資料集

訓練的資料對共有兩張圖像，分別做為輸入及目標輸出。目標輸出圖片的生成方式是從先前蒐集的地形高度圖上隨機取 256×256 大小的正方形。每次選取正方形的位置都是隨機的，以增進訓練資料的一般性。

至於輸入圖像，如果要用人工模仿每張目標輸出的方式畫出手繪圖，會花費很多時間。所以我們使用中值模糊的方式來快速生成輸入圖片。經過中值模糊處理後的圖片，在真實高度圖中的大山脊上的小河谷和大河谷上的小凸起物會被抹除，剩下的線條類似手繪圖的大致架構，符合模型輸入中山脈的大致架構這項條件。為了增進模型對於不同模糊程度的一般性，每張圖像的模糊程度是隨機的，也就是說中值模糊的 kernel size 是 1~9 隨機的奇數。

最後，我們將中值模糊過的真實高度圖和原始的真實高度圖組成模糊-清晰資料對(如圖 10)，用來訓練地形擬真模型。訓練資料集的 train、val、test 圖像數量與貼圖模型的資料集相同，如表 3。

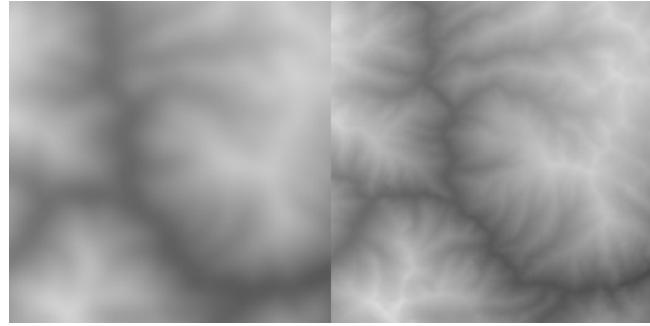


圖 10: 地形擬真模型之訓練資料對

表 3: 貼圖模型與地形擬真模型的訓練資料的 train、val、test 圖像數量 (單位：張)

模型的資料集	train	val	test	總計
貼圖模型				
地形擬真模型	2560	640	400	3600

五、本研究建構的模型結構—VAE-pix2pix

(一) 貼圖模型

本研究建構的模型為 VAE 及 pix2pix 的結合。pix2pix 中的 U-Net 有強大的生成能力，負責依照高度圖生成空照圖。但是經過測試，我們發現如果只單獨提供高度圖給 U-Net，它會無法反映出如雪線高度、山壁顏色等資訊。

為了解決這樣的問題，我們建構了 VAE-pix2pix 模型，在 pix2pix 的 U-Net 前增加了 style encoder (為一個 VAE)，讓 style encoder 負責從目標空照圖提取出這些風格資訊，再由 U-Net 以正確的風格生成空照圖。從 VAE 的角度來看，此模型相當於利用 U-Net 作為 Decoder，並在瓶頸處額外輸入高度圖作為空間資訊的 VAE。我們將建構的模型稱之為 VAE-pix2pix 模型。

為了不限制能處理的圖片大小，本研究的 style encoder 不會像典型的 VAE 一樣產生向量作為 latent code，而是會保留空間維度。

因為 style encoder 的輸入中含有目標空照圖 (ground truth) 的資訊，所以作為瓶頸的 latent code 必須足夠窄，以防止 U-Net 輕易的將目標空照圖直接輸出。如果輸入圖的長寬為 256×256 ，則 VAE 會將 latent code 縮小到一個有 16 channels，長寬為 4×4 的 tensor。貼圖模型的結構具體如圖 11。

此外，模型中的 style encoder 中有使用 Batch Normalization。本研究將 batch size 設為

30，所以 Batch Normalization 不致於使 style encoder 輸出的 latent code 不穩定。

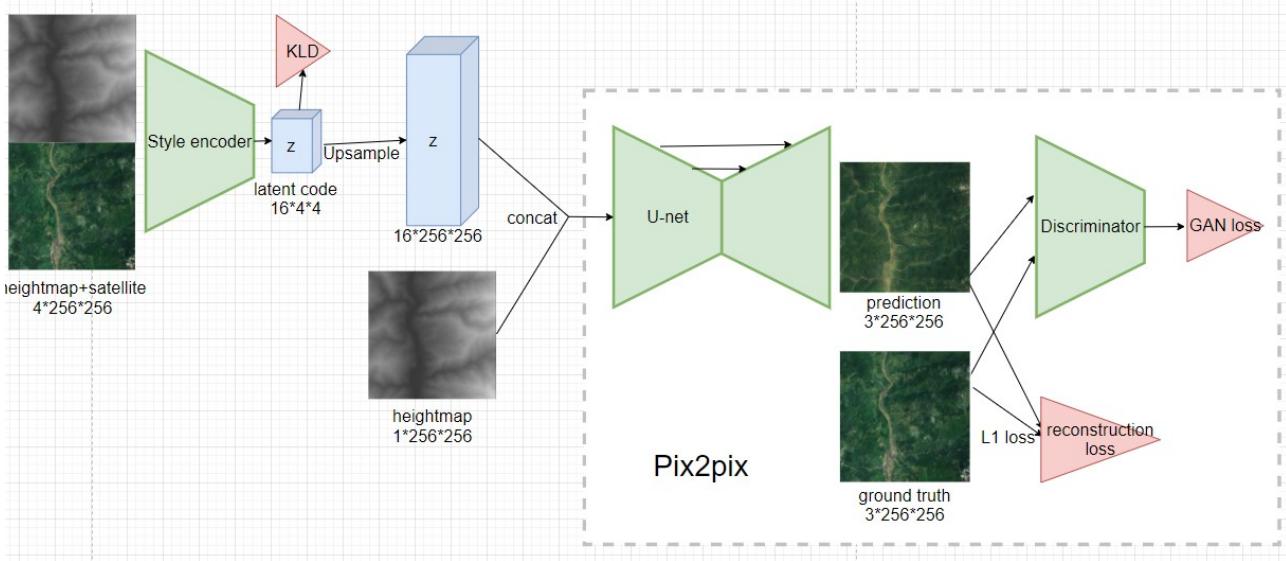


圖 11: 貼圖模型的 VAE-pix2pix 模型結構

(二) 地形擬真模型

地形擬真模型結構和貼圖模型結構相同，差別只在地形擬真模型訓練時，是要以模糊高度圖做為輸入，藉以生成出較多細節的高度圖，而貼圖模型則是以清晰高度圖 (ground truth) 做為輸入。地形擬真模型的結構具體如圖 12。

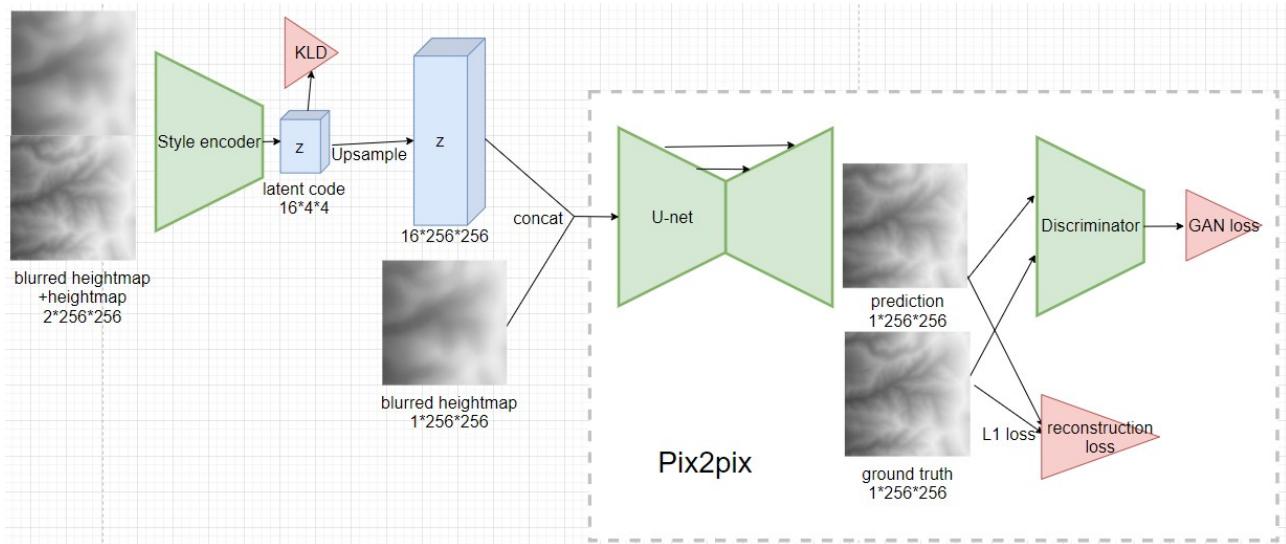


圖 12: 貼圖模型的 VAE-pix2pix 模型結構

因本研究的模型結構是在 pix2pix 模型的基礎上額外加上 VAE，所以 pix2pix 的部分仍然沿用論文作者的程式碼，進行訓練及測試時也是使用 pix2pix 原作者所提供的程式進行訓

練及測試。生成完貼圖模型的訓練資料集後，我們先訓練最基本的 pix2pix 模型。我們利用 pix2pix 模型所提供的 train.py 程式碼進行訓練，訓練指令如下：

六、 訓練貼圖模型

生成完貼圖模型的訓練資料集後，我們先訓練最基本的 pix2pix 模型。我們利用 pix2pix 模型所提供的 train.py 程式碼進行訓練，訓練指令如下：

```
python train.py --model pix2pix --name pix2pix_sat_all --dataroot datasets/  
height29_10km_pix2pix_sat_all --direction AtoB --input_nc 1 --output_nc 3
```

七、 訓練 VAE-pix2pix 模型

在五、中提到我們是在 pix2pix 模型的架構下加入自行建構的 VAE，故訓練指令與上述指令的差異只在 model 參數的不同，下述指令的 vae_pix2pix 即為本研究的模型結構：

```
python train.py --model vae_pix2pix --name vae_sat_all --dataroot datasets/  
height29_10km_pix2pix_sat_all --direction AtoB --input_nc 1 --output_nc 3
```

兩個貼圖模型的訓練結果會於 **研究結果與討論** 中進行說明。

八、 訓練地形擬真模型

在訓練原 pix2pix 模型與本研究建構的模型前，我們先對兩個模型內的 pix2pix 進行以下調整：

(一) 對 pix2pix 模型進行調整

pix2pix 模型原本是針對「一般圖片的風格轉換」這項工作設計的，這裡指的是像素值有固定上下界且通常有 RGB 三數值的圖片。這和我們一「生成擬真山脈」性質上有些許不同，所以我們調整 pix2pix 模型使其符合需求(如圖 13)：

1. 高度圖只有一個數值，所以設定模型的輸入 (input_nc) 及輸出 (output_nc) 維度皆為 1。
2. 高度圖的像素值沒有上下界的限制。為了讓輸出不因線性映射造成結果過大的失真，讓模型能均勻調整產生細節，我們把 generator 輸出層的激活函數 (activation function) tanh 去除。

3. 將山脈加上細節和一般的風格轉換有很大的差別：前者的輸出會沿用輸入的像素「值」作為架構(不只沿用形狀架構)，後者輸入和輸出的像素值則不必然有直接關係。所以我們在 generator 的最外層(也就是輸入和輸出層)加上額外的 skip connection 讓此層的輸出直接與模型輸入相加，成為最終的模型輸出(原本的 generator 第二層以下才有 skip connection)。因此，模型只需要學習輸入和目標輸出的差，也就是哪裡要增高、哪裡要降低，而不用學習如何重建整個高度圖。

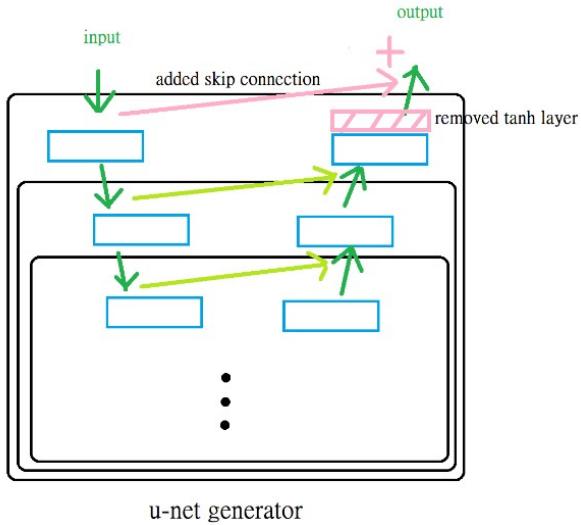


圖 13: Generator(U-Net) 的修改

(二) 訓練 pix2pix 結構之地形擬真模型

與貼圖模型相同，我們也利用 pix2pix 所提供的訓練程式碼來訓練模型，具體指令如下：

```
python train.py --model pix2pix --name pix2pix_hei_all_med1-29 --dataroot
datasets/height29_10km_pix2pix_hei_all_med1-29 --direction AtoB --input_nc 1 --
output_nc 1
```

(三) 訓練本研究模型結構之地形擬真模型

而訓練本研究模型結構之地形擬真模型的具體指令如下：

```
python train.py --model vae_pix2pix --name vae_hei_all --dataroot datasets/
height29_10km_pix2pix_sat_all --direction AtoB --input_nc 1 --output_nc 3
```

兩種地形擬真模型的訓練結果會於 **研究結果與討論** 中進行探討。

叁、研究結果與討論

一、評斷生成對抗網路模型的方法

一般來說，很難找到一個好的方法來評斷生成對抗網路的表現，因為它不像一般的分類器能利用分類的精確度來評斷神經網路的優劣。

在本研究中，我們將會利用比較輸出與目標輸出的 L1 Loss、L2 Loss、Perceptual Loss、FID、SSIM Index (structural similarity index) 及生成每張圖像所需之平均時間來探討物理侵蝕模型、基礎的 pix2pix 模型與本研究模型的差異

(一) L1 Loss

L1 Loss 的計算方式是將兩張大小相同圖像所相對應的像素值相減後取絕對值，再相加在一起，最後取平均。具體公式如式 2：

$$\text{L1 Loss}(X, Y) = \frac{1}{h \times w} \left(\sum_{i=1}^h \sum_{j=1}^w |X_{i,j} - Y_{i,j}| \right) \quad (2)$$

其中， h, w 分別為圖片的高度及寬度。可以發現 L1 Loss 的數值越小，代表兩張圖越相近。藉由 L1 Loss 我們可以看出兩張圖的相似程度。

(二) L2 Loss

L2 Loss 與 L1 Loss 的計算方式相似，只是將絕對值替換成平方，具體計算公式如式 3：

$$\text{L2 Loss}(X, Y) = \frac{1}{h \times w} \left(\sum_{i=1}^h \sum_{j=1}^w (X_{i,j} - Y_{i,j})^2 \right) \quad (3)$$

與 L1 Loss 相似，L2 Loss 同樣是數值越小，代表兩張圖越相近，我們也可以透過 L2 Loss 來看出兩張圖的相似程度。但是 L2 Loss 對於偏離越多的值，會造成平方效果的改變。

(三) Perceptual Loss[11]

在計算 Perceptual Loss 時，會利用到一個已經訓練好 (pre-trained) 的 VGG16 模型。Perceptual Loss 的計算方式是計算兩張圖像在 VGG16 各層 activation 的 L1 Loss，最後再將各層計算出的 L1 Loss 相加。從 Perceptual Loss 可以看出兩張圖的風格是否相似，且 Perceptual Loss 同樣是數值越小，代表兩張圖越相近。

(四) FID (Fréchet Inception Distance)[2]

在計算 FID 時，會利用到一個已經訓練好 (pre-trained) 的 inception network v3 神經網路來提取兩張圖片的特徵 (feature)。圖片的特徵 (為一個 2048 維的高階特徵) 主要可以從 inception network 輸出層的前一層提取到。對於目標輸出，我們可以假設這個 2048 維向量是服從高斯分布。那由神經網路輸出的特徵應該也要服從高斯分布。所以我們知道生成對抗網路的目標是使這兩個分布的距離盡量接近。

而計算這兩個分布的距離等同於求目標輸出和輸出的 2048 維特徵的距離。數學上，如果想要計算兩個分布的距離，我們可以使用 Fréchet distance 來進行計算。

在計算上，我們會假設這兩個分布是服從高斯分布，且我們知道若一個隨機變數服從於高斯分布，則這個隨機變數可以使用高斯分布的標準差與平均表示，只要兩個分布的標準差和平均皆相同，則兩個分布相同。標準差和平均就是用來計算 FID。但因為這裡我們要計算的是多維的向量，所以我們會使用平均和共變異數 (covariance) 矩陣來計算兩個分布的距離。而平均的維度是 2048 綴，而共變異數矩陣就是一個 2048×2048 綴的矩陣。有了以上的定義後，我們就可以使用式 4 來計算輸出與目標輸出的 FID。

$$\text{FID}(X, Y) = \|\mu_X - \mu_Y\|_2^2 + \text{Tr} \left(\Sigma_X + \Sigma_Y - 2 (\Sigma_X \Sigma_Y)^{\frac{1}{2}} \right) \quad (4)$$

(五) SSIM index (structural similarity index)[10]

SSIM 指標是一種用來評斷兩張圖像相似程度的方法，相較於其他種方法 SSIM 指標能更好的符合人眼對圖像品質的判斷。SSIM 指標主要透過比較兩張圖片的亮度、對比度及結構 (structure) 來評斷兩張圖片的相似程度，具體計算方式如式 5：

$$\begin{aligned} \text{SSIM}(\mathbf{X}, \mathbf{Y}) &= [l(\mathbf{X}, \mathbf{Y})]^{\alpha} [c(\mathbf{X}, \mathbf{Y})]^{\beta} [s(\mathbf{X}, \mathbf{Y})]^{\gamma}, \\ l(\mathbf{X}, \mathbf{Y}) &= \frac{2\mu_x\mu_y+c_1}{\mu_x^2+\mu_y^2+c_1}, c(\mathbf{X}, \mathbf{Y}) = \frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_2}, s(\mathbf{X}, \mathbf{Y}) = \frac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+c_3} \end{aligned} \quad (5)$$

$l(X, Y)$ 是用來比較 X, Y 兩張圖的亮度， $c(X, Y)$ 則是用來比較對比度，而 $s(X, Y)$ 用來比較兩張圖的結構。 μ_x, μ_y 代表兩張圖像素值的平均， σ_x, σ_y 代表兩張圖像素值的標準差， σ_{xy} 為兩張圖的共變異數 (covariance)， C_1, C_2, C_3 是三個常數，以避免出現分母 0 的情況。另外，在本研究中，我們設定 $\alpha = \beta = \gamma = 1$ 。

根據以上的公式，我們可以發現 SSIM index 滿足對稱性 ($\text{SSIM}(X, Y) = \text{SSIM}(Y, X)$)、有界性 ($-1 \leq \text{SSIM}(X, Y) \leq 1$) 及極限值唯一 ($\text{SSIM}(X, Y) = 1 \iff X = Y$)

(六) 生成每張圖像所需的平均時間

我們利用 PyTorch 內建的 `torch.cuda.Event()` 來計算生成圖像所需的時間。計算完生成總時間後，就可以算出生成單張圖像所需的平均時間。在本研究中，我們是利用一張 Nvidia GTX 1080 Ti 12 GB 的顯示卡來進行圖像的生成。

二、貼圖模型之訓練結果

表 4 為原 pix2pix 結構與 VAE-pix2pix 結構的貼圖模型的測試結果，其圖像皆為來自 test 資料集之圖像，也就是說模型並沒有在訓練過程中“看過”這些圖像。藉由觀察這些圖像，我們可以更好的評斷模型的學習程度。表 5 則為兩個模型在 test 資料集的平均指標值。

從表 5 可以發現我們建構的 VAE-pix2pix 架構相較於原 pix2pix 架構生成出的衛星空照圖較接近於目標輸出，不僅山脈的顏色更接近於目標輸出，且輸出的空照圖相當符合輸入高度圖的結構。這點也反映在各指標上，L1 Loss、L2 Loss、Perceptual Loss 及 SSIM 都表現較好。

三、地形擬真模型之訓練結果

與貼圖模型相同，地形擬真模型的測試圖像都是來自 test 資料集。測試完成後，測試結果如表 6，我們會將模型輸出與目標輸出計算出各指標，如表 7，藉此來判斷每個模型的表現。

從表 7，我們可以發現 VAE-pix2pix 生成的高度圖較原 pix2pix 所生成的圖像來說更接近於目標輸出。由各指標也可以看到 VAE-pix2pix 要優於 pix2pix。相較於物理侵蝕模型，我們的方法約比其生成的速度快 20 倍。

四、將 VAE-pix2pix 的 latent code 進行 PCA (主成分分析)

在應用時，地形擬真模型和貼圖模型中的 U-Net 各需要輸入 16 維的 latent code 作為指定的生成風格。為了方便應用，我們將兩種模型在資料集中的所有圖片產生的 latent code 合併後進行 PCA，這樣在使用時，使用者只需要輸入較少維的值(例如在 Unity 客戶端中為 4 維)，再用 PCA 得到的轉換矩陣將它轉換到 latent space，即可輸入 U-Net。

表 4: 原 pix2pix 結構及 VAE-pix2pix 結構的貼圖模型的測試結果

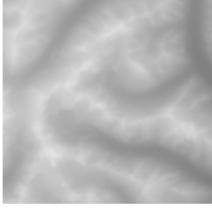
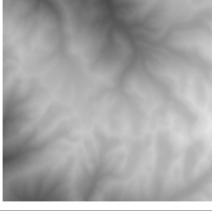
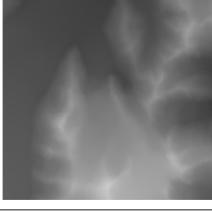
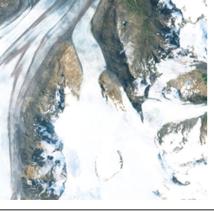
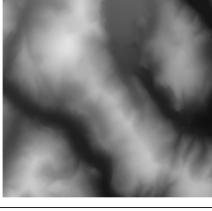
地區	輸入	模型輸出		目標輸出
		原 pix2pix	VAE-pix2pix	
橫斷山脈				
喜馬拉雅山				
祕魯安地斯山				
阿根廷冰河地形				
加拿大冰河地形				

表 5: 原 pix2pix 及 VAE-pix2pix 在 test 資料集的平均指標值

模型	L1 Loss	L2 Loss	Perceptual Loss	FID	SSIM	平均生成時間 (毫秒)
原 pix2pix	44.17	3879.938	3.4326	85.536	0.195	7.007
VAE-pix2pix	18.466	798.316	2.8969	96.251	0.37	10.052

表 6: 物理侵蝕模型、原 pix2pix 結構及 VAE-pix2pix 結構的地形擬真模型的測試結果

地 區	輸入	模型輸出			目標輸出
		物理侵蝕模型	原 pix2pix	VAE-pix2pix	
橫 斷 山 脈					
喜 馬 拉 雅 山					
祕 魯 安 地 斯 山					
阿 根 廷 冰 河 地 形					
加 拿 大 冰 河 地 形					

表 7: 物理侵蝕模型、原 pix2pix 及 VAE-pix2pix 在 test 資料集的平均指標值

模型	L1 Loss	L2 Loss	Perceptual Loss	FID	SSIM	平均生成時間 (毫秒)
物理侵蝕模型	9.735	157.747	0.7734	144.036	0.897	1770
原 pix2pix	2.837	196.875	0.8327	100.7692	0.931	7.863
VAE-pix2pix	1.036	9.413	0.424	33.742	0.982	10.247

五、建構模型的 API 伺服器

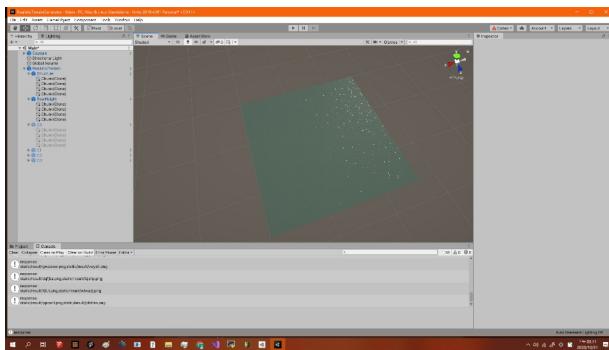
為了避免每次需要應用各模型處理高度圖時都要重新載入模型，我們利用 Python 的 Flask 套件建立了一個 API 伺服器，並將其建置於工作站上。用戶可以直接上傳高度圖，並在伺服器上用訓練好的各模型進行處理。這個 API 可以快速的在模型之間切換，且支援本研究中所有訓練的模型。

六、Unity 用戶端

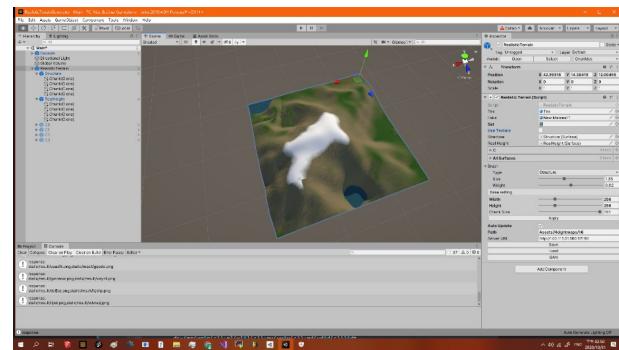
我們自行開發了一個 Unity 客戶端，使用戶可以在 Unity Editor 的編輯模式中直接對地形進行操作。使用者可以匯入一張高度圖或直接在地形上繪製 3D 的模型。每當畫完一筆畫後，Unity 會直接對 API 伺服器送出請求，並在一秒內更新出擬真的山脈地形。也可以將貼圖模型所生成的擬真衛星空照圖貼在 3D 模型上，具體功能如圖 14a、圖 14b、圖 14c、圖 14d。

七、由用戶端調整 latent code 對於輸出風格的效果

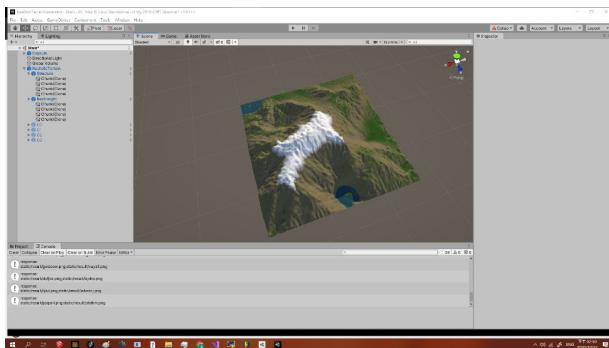
圖 15 展示了以 512 x 512 的橫斷山脈模糊高度圖為輸入高度圖，並在空間上使用不同的 latent code，造成模型輸出在空間上的風格差異。整張圖 latent code 的設置為：第一主成分由左邊界的-0.6 線性增加至右邊界的 0.6，第二主成分由下邊界的-0.6 線性增加至上邊界的 0.6，更後面的主成分皆設為 0。



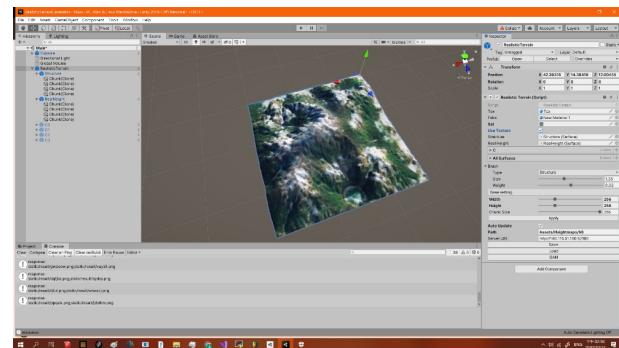
(a) Unity 客戶端的初始畫面



(b) 用戶手繪之大致地形



(c) 經過地形擬真模型處理後的擬真地形



(d) 將貼圖模型的輸出貼在擬真地形上的效果

圖 14: Unity 客戶端功能

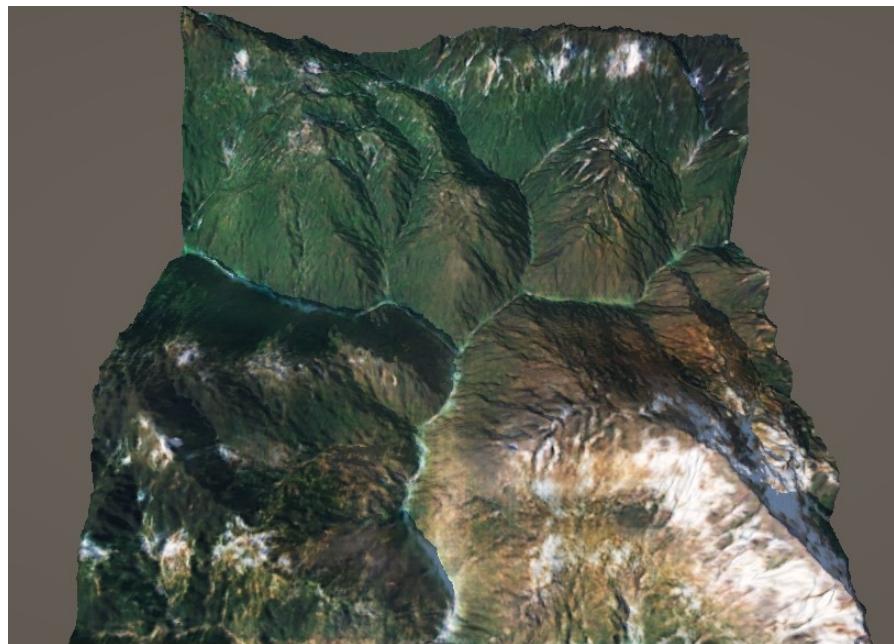


圖 15: 在空間上展示不同風格的輸出

八、透過改變 style encoder 的輸入改變輸出圖像的風格

為了測試以上的討論，我們將一張高度圖及多張與此張高度圖不互相對應的空照圖一起輸入到 VAE-pix2pix 的 style encoder 中，驗證能改變輸出圖像的風格。表 8 為測試的結果。可以發現 VAE-pix2pix 模型可以很好的將輸入空照圖的特徵提取出來，且也可以保留高度圖上河流及山脊等特徵。

九、討論物理侵蝕模型、pix2pix 模型及 VAE-pix2pix 之間的差異

在此節中，我們會比較物理侵蝕模型、原 pix2pix 模型及本研究建構的 VAE-pix2pix 模型之間的差異，比較如表 9。

肆、結論與應用

在本研究，我們利用 NASA 的 SRTM 1 Arc-Second 及 MapTiler 網站收集了全球五個地區的高度圖及相對應的空照圖。利用這些收集的圖像，訓練了自行建構的 VAE-pix2pix 模型。VAE-pix2pix 為 Variational Autoencoder (VAE) 及 pix2pix 結合的模型，可以將人工繪製的高度圖自動加上真實山脈應有的細節 (包含尖銳的山脊、山壁上的紋路、連續的河流網路等……)，也能生成相對應的擬真衛星空照圖。

經過實測，相較於原 pix2pix 模型，VAE-pix2pix 所生成的高度圖及空照圖會更接近於真實世界的山脈高度圖，且 VAE-pix2pix 模型也可以透過改變其 latent code 的數值來生成出不同風格的高度圖及衛星空照圖，如地貌的顏色或雪線的高度等，這些都能增加模型生成圖像的多樣性，讓應用更為廣泛。與物理侵蝕模型進行比較，不僅生成速度遠快於物理侵蝕模型，生成品質也更為擬真，這些優於傳統模型的地方。

為了使模型的使用更加簡單，不用在終端機上打入許多複雜的指令，我們將模型的使用包裝成 Unity 客戶端，Unity 客戶端可以在圖形使用者介面完成在模糊高度圖加上山脈細節的工作，並能直接在畫面上顯示 3D 模型，也可以將生成出的衛星空照圖貼在擬真地形的 3D 模型中，使空照圖及高度圖能更好的呈現。

綜合以上，本研究的 VAE-pix2pix 模型可以生成出更為擬真的高度圖及空照圖。而我們開發的 Unity 客戶端，可以使我們的模型直接應用於遊戲的開發中，也使得原先需要分成兩個步驟的生成擬真的山脈地形與生成相對應的空照圖整合為一個步驟。這些都會讓遊戲開發生成擬真山脈模型的任務變得十分容易。

表 8: 將同一張高度圖與不同的空照圖作為 style encoder 的輸入，並比較其輸出

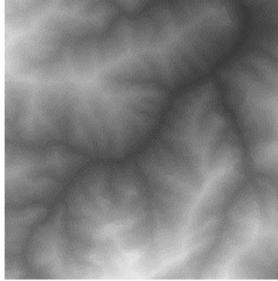
輸入的高度圖	輸入的空照圖(style)	VAE-pix2pix 的輸出
		
		
		
		
		

表 9: 物理侵蝕模型、原 pix2pix 模型及 VAE-pix2pix 的比較

模型	功能	生成速度	生成品質	是否可以改變生成圖像的風格
物理侵蝕模型	將大致高度圖經過侵蝕後變得較為擬真	慢 (約 1.7 秒)	最差	否
原 pix2pix 模型	將大致高度圖變得更為擬真、根據高度圖生成相對應的衛星空照圖	最快 (約 7 毫秒)	其次	否
VAE-pix2pix 模型	將大致高度圖變得更為擬真、根據高度圖生成相對應的衛星空照圖	快 (約 10 毫秒)	最好	是

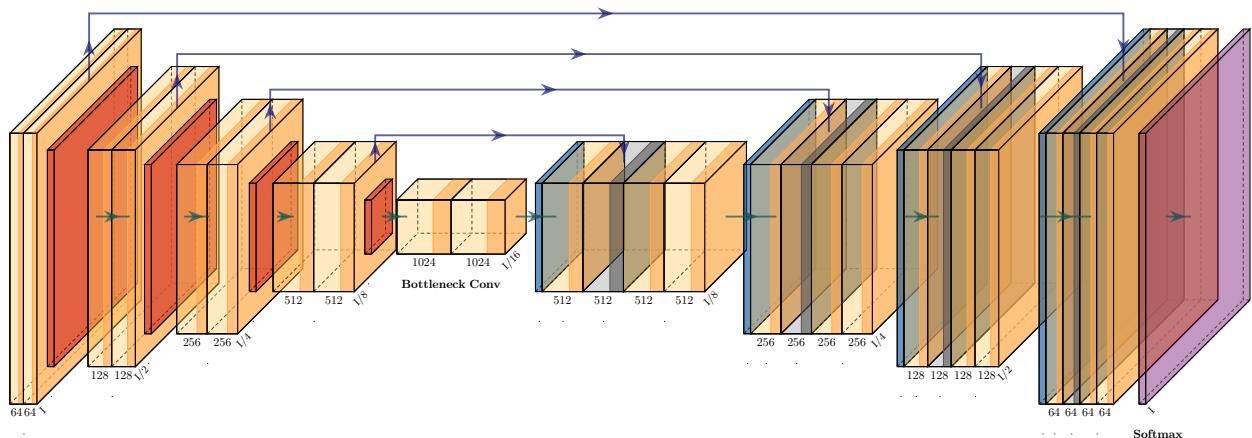
參考文獻

- [1] 程品奕、李杰穎.“利用生成對抗網路生成擬真的山脈地形”. In: 中華民國第 60 屆中小學科學展覽會 (2020). URL: <https://www.ntsec.edu.tw/Science-Content.aspx?a=6821&fld=&key=&isd=1&icop=10&p=1&sid=16557>.
- [2] D.C Dowson and B.V Landau. “The Fréchet distance between multivariate normal distributions”. In: *Journal of Multivariate Analysis* 12.3 (1982), pp. 450–455. ISSN: 0047-259X. doi: [https://doi.org/10.1016/0047-259X\(82\)90077-X](https://doi.org/10.1016/0047-259X(82)90077-X). URL: <http://www.sciencedirect.com/science/article/pii/0047259X8290077X>.
- [3] Earth Resources Observation And Science (EROS) Center. *Shuttle Radar Topography Mission (SRTM) 1 Arc-Second Global*. 2017. doi: [10.5066/F7PR7TFT](https://doi.org/10.5066/F7PR7TFT). URL: https://www.usgs.gov/centers/eros/science/usgs-eros-archive-digital-elevation-shuttle-radar-topography-mission-srtm-1-arc?qt-science_center_objects=0#qt-science_center_objects.
- [4] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.
- [5] Phillip Isola et al. “Image-to-image translation with conditional adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.
- [6] Balázs Jákó and Balázs Tóth. “Fast Hydraulic and Thermal Erosion on GPU.” In: *Eurographics (Short Papers)*. 2011, pp. 57–60.
- [7] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).
- [8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.
- [9] Maximilian Seitzer. *pytorch-fid: FID Score for PyTorch*. <https://github.com/mseitzer/pytorch-fid>. Version 0.1.1. Aug. 2020.
- [10] Zhou Wang et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

- [11] Richard Zhang et al. “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”. In: *CVPR*. 2018.

附錄

壹、VAE-pix2pix 的模型結構

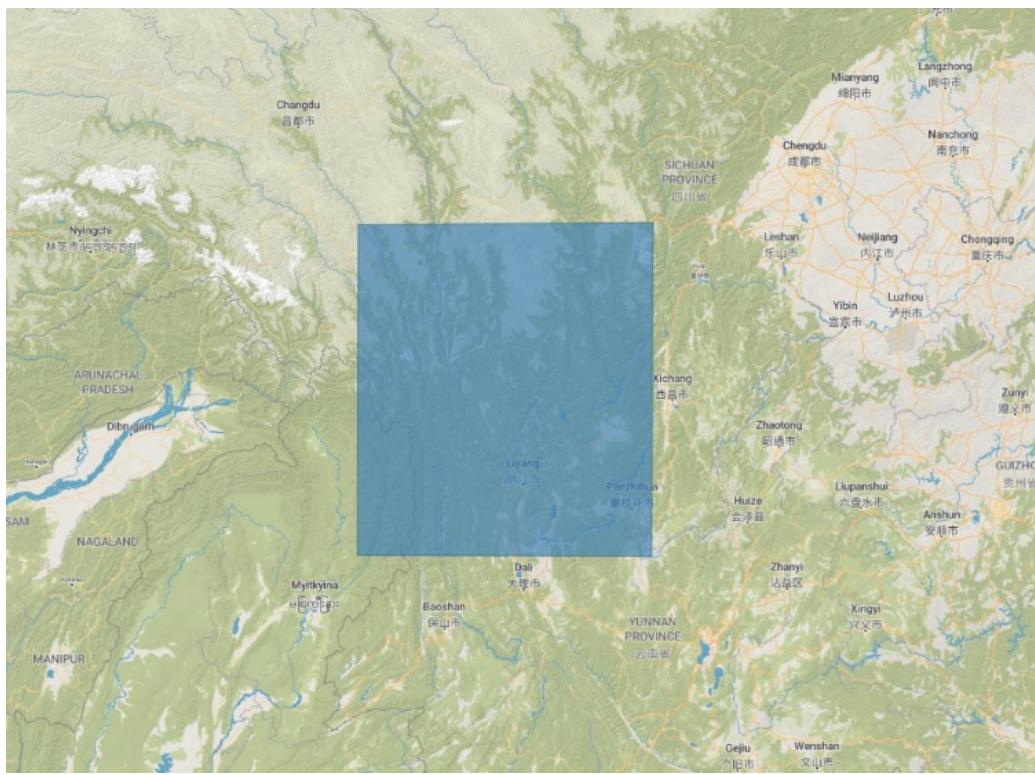


貳、高度圖及空照圖的具體收集範圍

如內文所提，我們會利用高度圖左下角的經緯度座標表示一張高度/空照圖，如一張高度圖包含 25°N, 98°E、25°N, 99°E、24°N, 99°E、24°N, 98°E 四個座標點所圍成的範圍，則此張高度圖的檔名即為 N24E98。除了會在下面列出各個座標點，我們也有將具體的收集範圍在地圖上框出。

一、中國橫斷山脈

- | | | |
|------------|-------------|-------------|
| 1. N27E099 | 7. N29E101 | 13. N28E099 |
| 2. N27E098 | 8. N29E100 | 14. N28E098 |
| 3. N26E101 | 9. N29E099 | |
| 4. N26E100 | 10. N29E098 | 15. N27E101 |
| 5. N26E099 | 11. N28E101 | |
| 6. N26E098 | 12. N28E100 | 16. N27E100 |



二、喜馬拉雅山脈

- | | | |
|------------|------------|-------------|
| 1. N29E081 | 5. N28E084 | 9. N28E086 |
| 2. N29E082 | 6. N28E085 | 10. N27E087 |
| 3. N29E083 | 7. N27E085 | |
| 4. N28E083 | 8. N27E086 | |



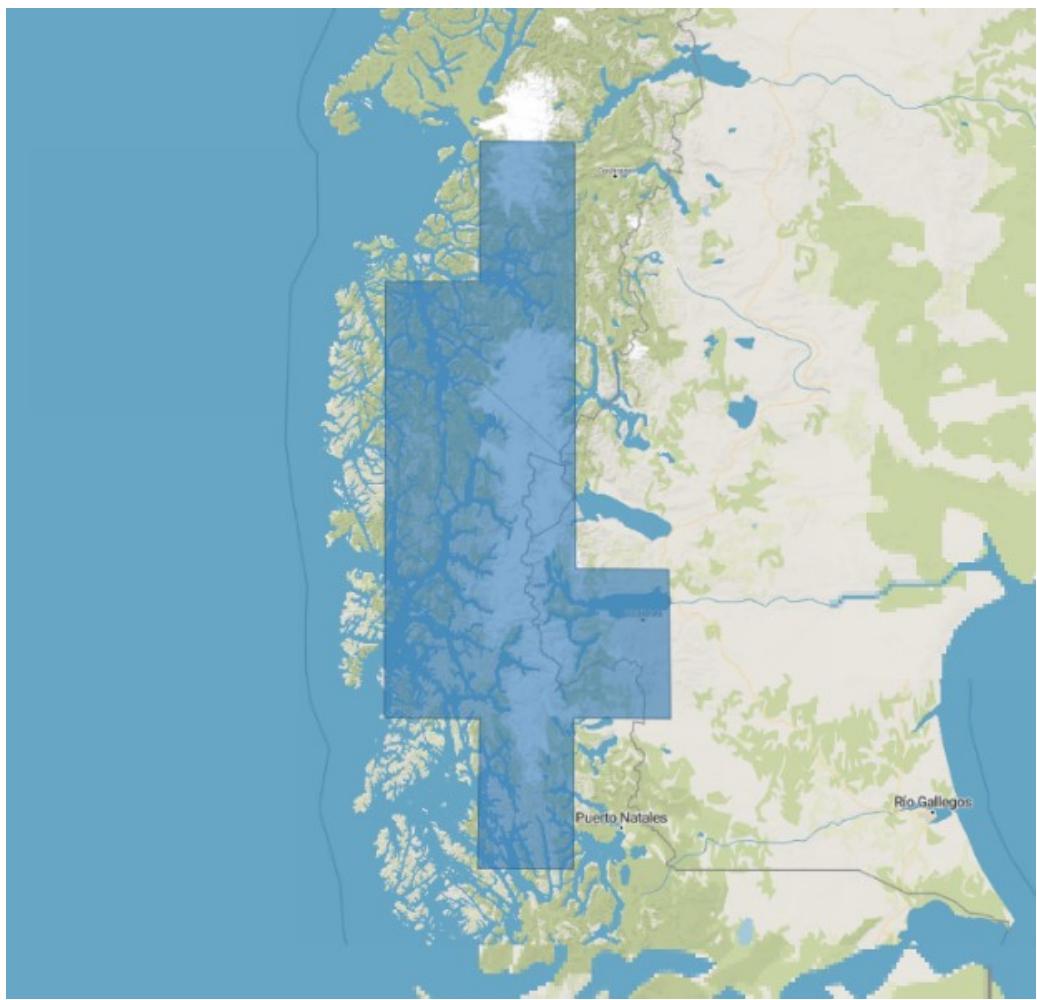
三、 祕魯安地斯山脈

- | | | |
|------------|-------------|-------------|
| 1. S07W079 | 6. S10W077 | 11. S14W076 |
| 2. S08W079 | 7. S11W077 | 12. S14W075 |
| 3. S08W078 | 8. S12W077 | 13. S15W075 |
| 4. S09W078 | 9. S12W076 | 14. S15W074 |
| 5. S10W078 | 10. S13W076 | 15. S16W073 |



四、 阿根廷冰河

- | | | |
|------------|------------|------------|
| 1. S48W074 | 4. S50W074 | 7. S51W074 |
| 2. S49W074 | 5. S50W075 | 8. S51W075 |
| 3. S49W075 | 6. S51W073 | 9. S52W074 |



五、 加拿大冰河

- 1. N58W134
- 2. N57W133
- 3. N56W132
- 4. N56W131
- 5. N55W131

