DIP Final Project Report

0310754黃煜閔

0650736邱奕中

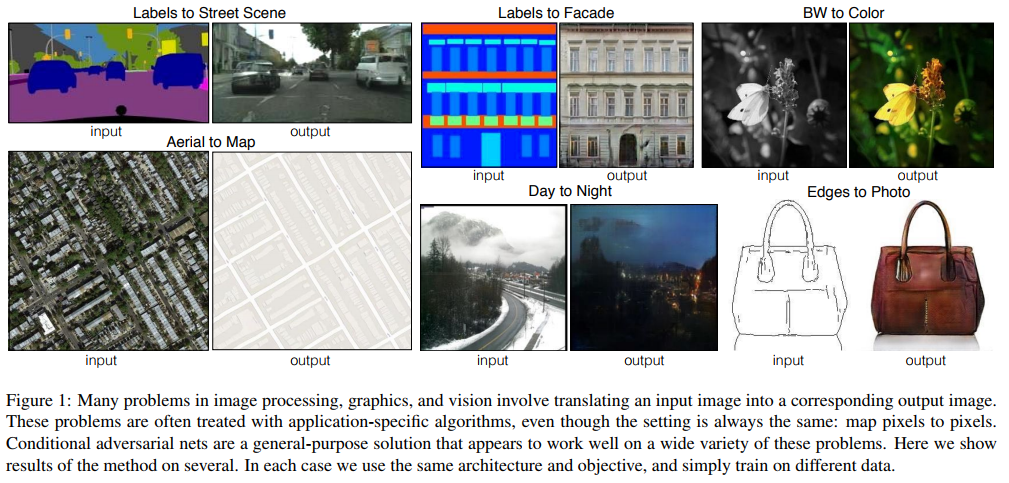
這次的final project，我們要做的事情是把空照圖轉成黑白的，把房子的地方標白色，其他地方標黑色，然後跟ground truth比相似度，包括pixel的接近度以及房子數量。我們可以從幾個不同的角度切入這個問題。首先，從傳統影像處理的角度來看，這是一個segmentation(加上classification)的問題；從machine learning的角度來看，這是一個非常標準的supervised learning問題，有input，有output，還有ground truth可以拿來訓練；最後，從另一個角度來看，這是一個domain transfer的問題，因此我們做了三種不同類型方法的比較。

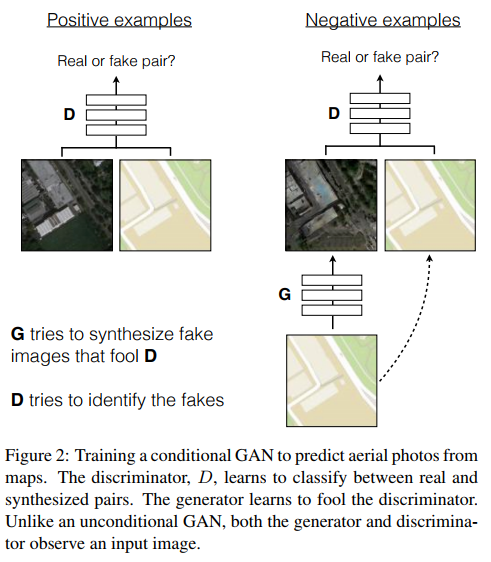
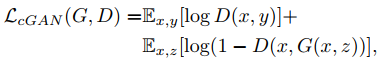
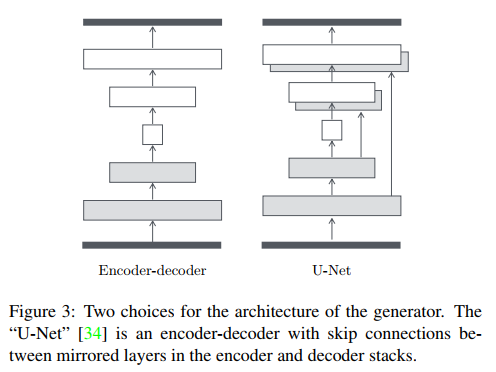
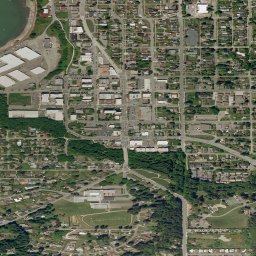
首先，一般影像處理的方法，在相關task上面，近幾年都輸machine learning的方法很多，但一般影像處理的方法還是有幾個優點，第一，他不需要training，training是很耗時間跟資源的，尤其是這次的data set，是5000\*5000的圖片，如果把圖片做downsample可以增加效率，但這次data量少其實也不太好train，因此如果時間急迫，沒時間train，或是電腦GPU太差，就可以考慮用傳統的方法，但條件允許的情況下，machine learning的方法performance一般都比較好。

接著就是supervised learning的方法，我覺得這搞不好是最有機會有好的performance的方法，基本上應該是基於CNN做改進，因為我們有ground truth，所以可以用監督式學習的方法train，一般來說，監督式學習的表現都比較好，但我們想試試比較新的GAN模型，所以沒有採用。

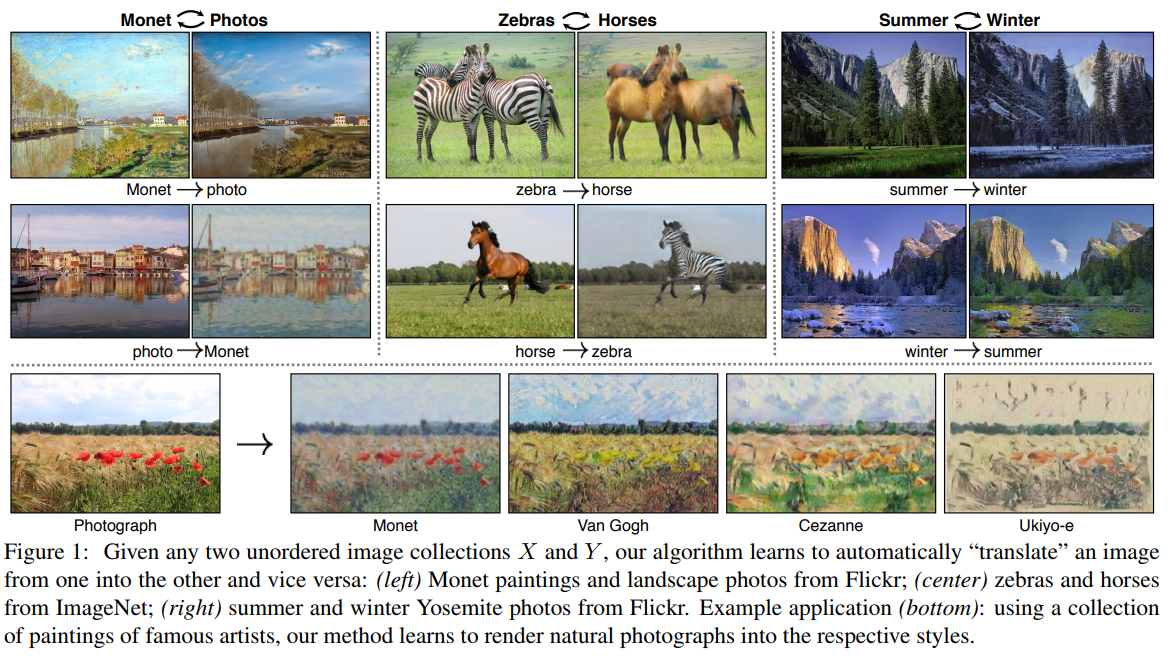
GAN是生成對抗網路，目前最主流的生成模型，其實我們可以把這次的task看成domain transfer，但是如果用一般的GAN，判斷他只能確定是不是成功轉到另一個domain(也就是黑白圖)，不會看有沒有保留原本資訊，這時候我們嘗試了幾個不同的想法，首先，cycle gan因為要轉回原本的domain，所以會保留原本的資訊，但這跟我們的目標不完全相同，我們擔心為了轉回原本的domain，減少reconstuct loss，他在一些他覺得不重要的地方產生奇怪pattern，只為了儲存轉回去的資訊，所以我們改了loss function，我們讓他變成類似監督式學習，加入跟ground truth的比較，希望可以有比較好的performance，另一個方法則是conditional adversarial networks，以下我們針對cycle gan、conditional adversarial networks，以及我們對他們做的改變，做一些介紹。

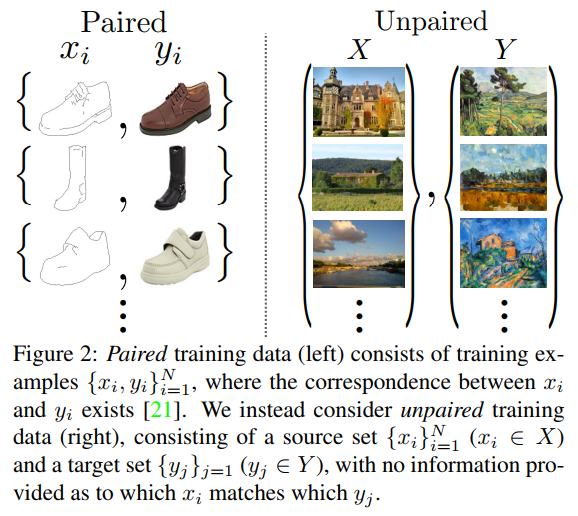
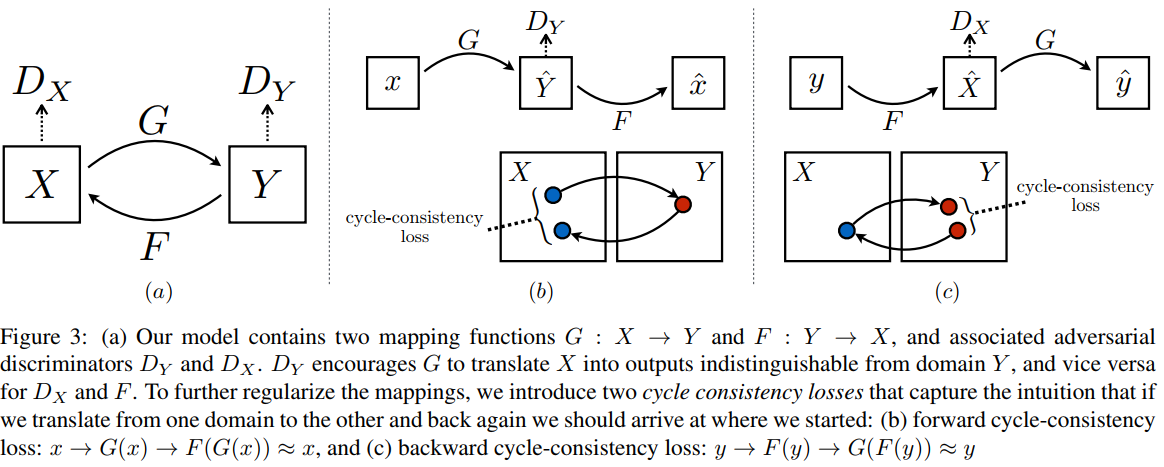
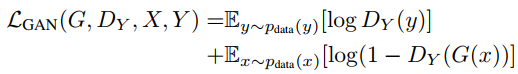
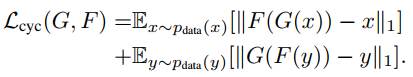
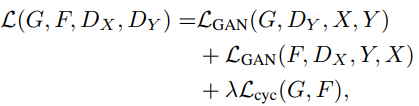
## **Image-to-Image Translation with Conditional Adversarial Networks**



1. Architecture Model  
     
   GAN是利用generator與discriminator之間對抗式的關係，來讓generator and discriminator的類神經網路同時持續進步以及更新，generator努力生出逼近真實的結果，而discriminator努力分辨出真實與生成的圖片，而這種對抗式方法，可以使生成出來的東西，效果很好。  
   Generator:   
   input是一組noise以及conditional term，就本次的task來說，input為空拍圖以及random noise，output則為房子是白色其他都是黑色的黑白影像。  
   Discriminator:  
   input為conditional term及generator 生出來的圖(synthetic image)以及real data，就本次的task來說，condtional term為空拍圖、synthetic image為黑白圖、real data為黑白圖。
2. Loss Function  
   (1)原始的conditional gan的loss function  
     
   有點像cross entropy(做classfication)，將real and fake做分類，與原始的gan的loss相比，多了一項y(conditional term)。  
   (2)L1 norm的loss  
     
   對generator來說，生成的結果與ground truth越接近，得到的loss就越低。  
   (3)final objective function  
     
   將第一項及第二項的loss相加，lamda則為可調參數，這邊我們將它設為100，希望output與ground truth越接近越好。
3. Network Architectures  
   (1)Generator  
     
   Generator可以是auto-encoder也可以是U-net的結構，本篇paper有提到，U-Net的結果比前項好，所以後來使用的NN為U-Net  
   (2)Discriminator  
   Discriminator使用PatchGAN，一般來說discriminator會對整張圖片做分析，而PatchGAN則是將圖片切割，並對每一個小區塊做分析。
4. Experiments (for 100 epoch)  
     
   

## **Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks**

  
CycleGAN是做domain transfer最經典的GAN，舉例來說，給一堆馬的圖片以及斑馬的圖片，cycleGAN就可以將馬轉成斑馬，以及斑馬轉成馬，又或者是，給一堆數位相機的照片以及梵谷的作品，cycleGAN就可以將數位相機的照片轉成梵谷style的圖片，以及梵谷的作品轉成真實style的圖片。

1. Paired v.s. Unpaired  
     
   與上面所介紹的conditionalGAN相比，CycleGAN為unsupervied learning，output並沒有ground truth。
2. Model  
     
   (1)Generator  
   cycle gan有兩個generator，G做Dx到Dy的轉換，F做Dy到Dx的轉換(這裡的Dx、Dy指的是兩個不同的domain)  
   (2)Discriminator  
   cycle gan有兩個discriminator，分別對生出的A、B style的圖片做discriminate
3. Loss Function  
   (1)原始GAN的Loss Function，可以看成一個分辨真與偽的classfication  
     
   (2)Cycle GAN有做一個reconstruct的動作，以下為reconstruct error  
     
   (3)Fianl Objective Function  
     
   因為有兩個generator and discriminator，所以前兩項為original gan的loss function，第三項為reconstruct loss function  
   (4)修正後的Objective Function:  
   在final objective function後加上與ground truth的loss，並乘上weight=1。
4. Experiments  
   
5. Problems  
   Cycle GAN效果沒有出來，我們覺得可能的原因有兩個，一個是training epoch只有到60，第二個可能是與ground truth的error只有乘上weight=1，說不定設20效果就會比conditional gan好，因為cycle gan多了reconstruct loss，有點類似normalized term，防止test時的overfitting。

Conclusion:

GAN本來就存在不好train的問題，這在當初goodfellow提出這model的時候就有這個問題了，而現在也沒有很好的解決方式，這或許是我們這次performance沒有很好的原因，另一個原因就是，我們用的GAN model，其實比較像是在做domain transfer，然而，我們的objective function，我們的generator跟discriminator，並沒有完全針對他的pixel by pixel的error，還有房子的數量，去做學習，因為目標的不同，model對圖的quality定義和project最後的measurement是不一樣的，讓我們的performance不會很好。儘管如此，這次期末專題也是滿有收穫的，我們survey了一些model，我們嘗試了一些很新、很酷的model，而且針對這個task，我們把他做了一些調整，雖然可能performance沒有很好，但過程中我們學到了很多東西，也是一個滿大的收穫。