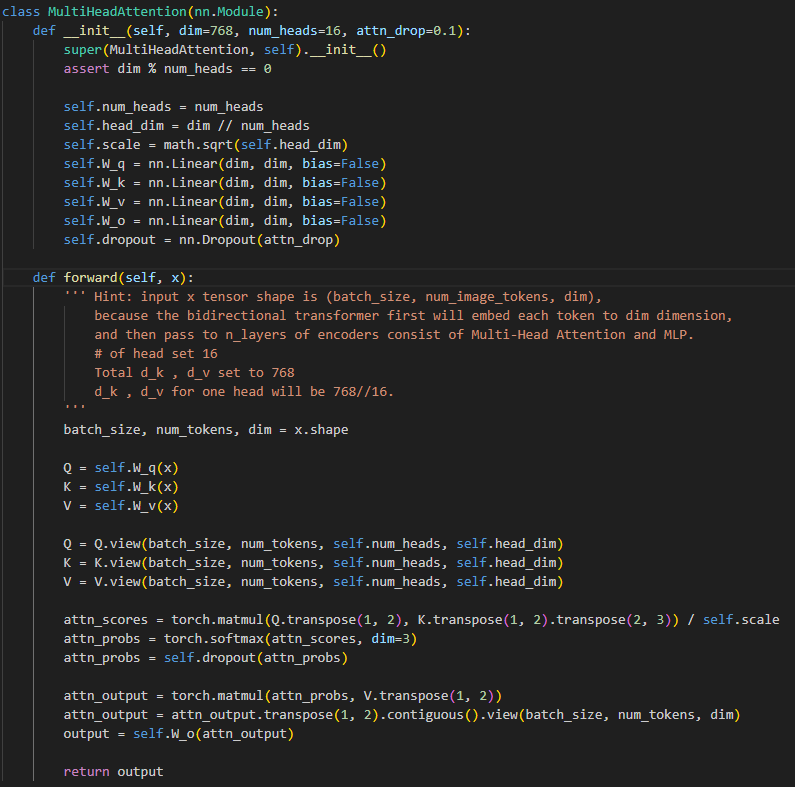
1. Introduction

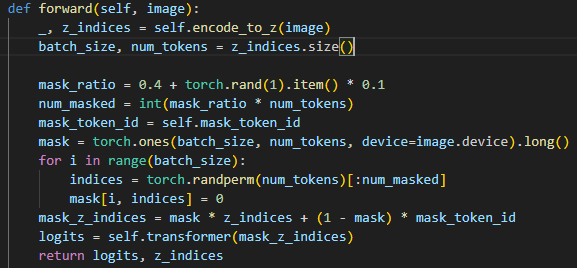
本次實驗實作目標將圖片補全，須完成multi-head attention、訓練細節和拼接不同模型等任務，最後透過decoder將transformer的猜測結過對應回原始圖片，並用fid當作模型效能的指標。

1. Implementation Details
   1. Multi-Head Self-Attention

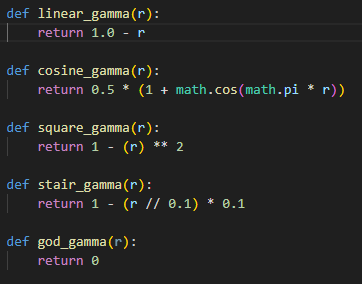


設置將input x轉換成QKV的矩陣，計算好每個head的維度(768/16=48)。在froward中，生成出QKV矩陣後，將Q和每一個K做內積，此用矩陣乘法表示，而鑲成矩陣的維度為(b, head, token, dim) x (b, head, dim, token) = (b, head, token, token)，接著過softmax和dropout後和V相乘，之後把head拼起來再經過linear以融合不同head。

* 1. stage2 training
     1. MVTM

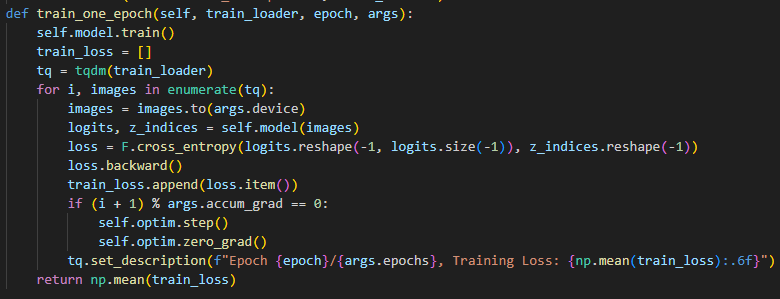


將image丟進vqgan得到z\_indices後，對batch size中每一筆資料做40%~50%的獨立遮罩，並餵給transformer。



gamma設計新增階段式和一次全部猜測等模式。

* + 1. training

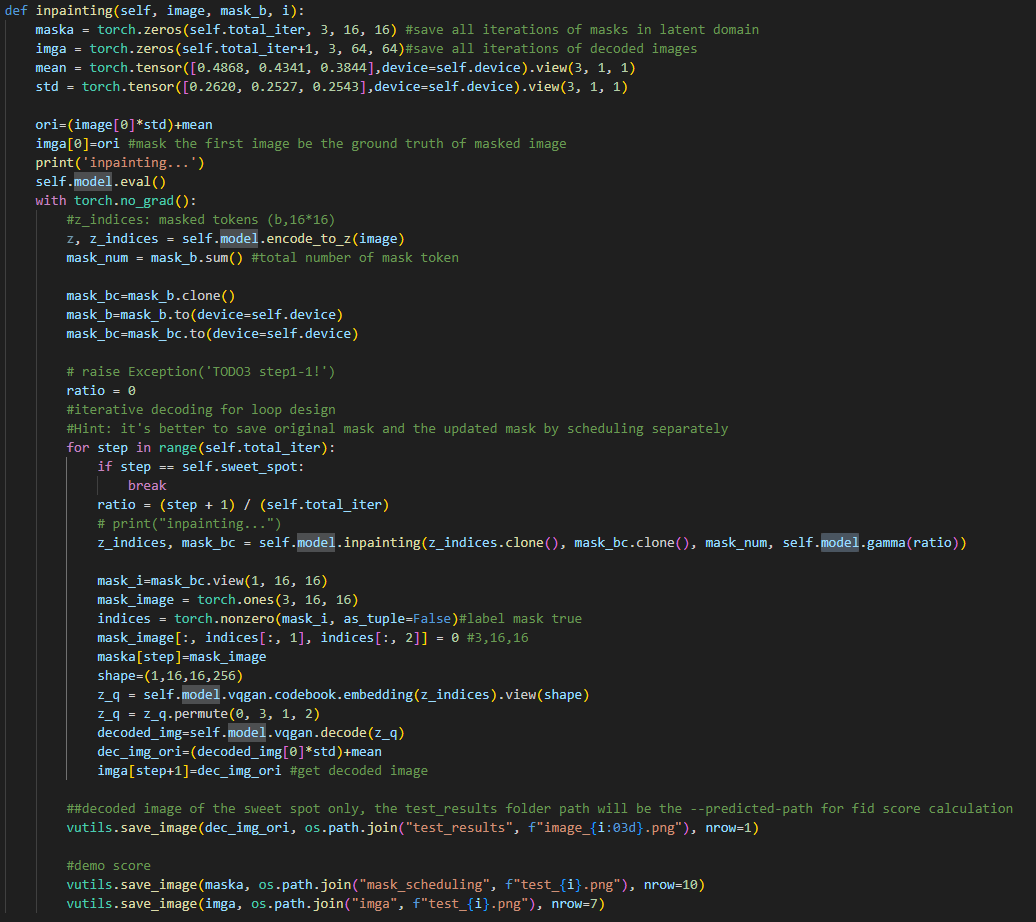


loss使用了cross\_entropy，由於擔心如果只計算遮罩的loss，會使模型減弱記住原始圖片的能力。

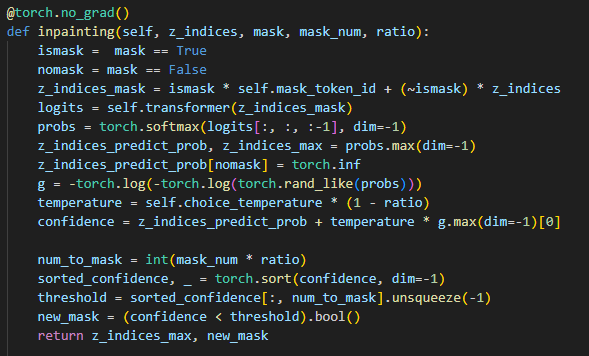


training時候如果有多顆GPU，則全部使用。

* + 1. inference for inpainting task

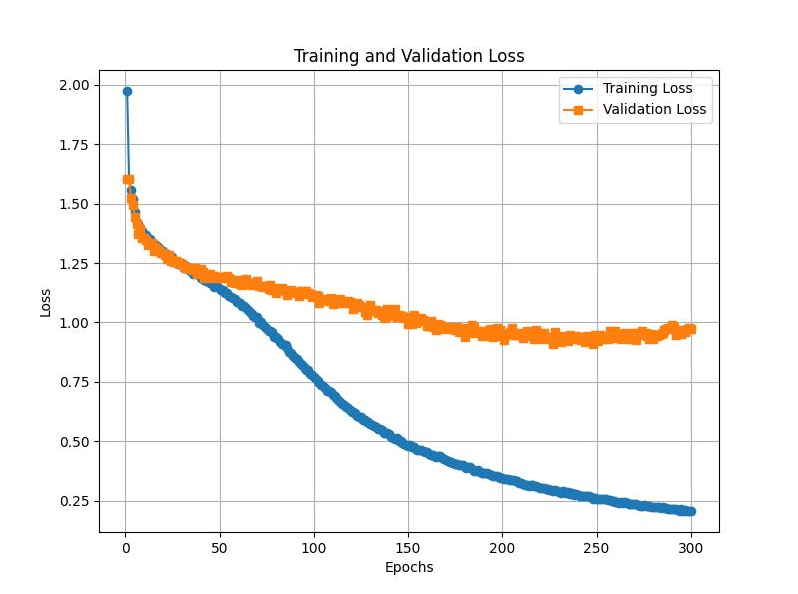


inpainting.py當中，先將圖片經過vqgan模型，交給VQGANTransformer做一定比例的猜測(0~1023分類)，慢慢跌代將圖片還原。



在VQGAN\_Transformer.py當中，先將目前得到的z\_indice套上mask後丟給transformer，並算出各類別的機率，但由於要猜測的類別不會包含mask\_token\_index所以第1024類不能包含在內，之後sort取得confidence排序將最前面一定比例的遮罩去除變成transformer所猜測的值。

1. Discussion
   1. Loss Graph



learning rate: 1e-4

epochs: 300

weight\_decay: 0.01 (regularization)

batch\_size: 100

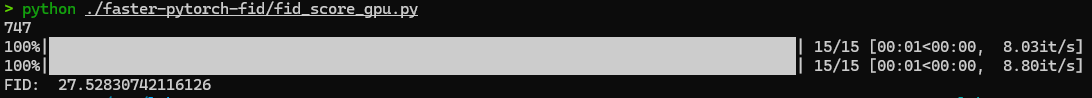
accum\_grad: 1

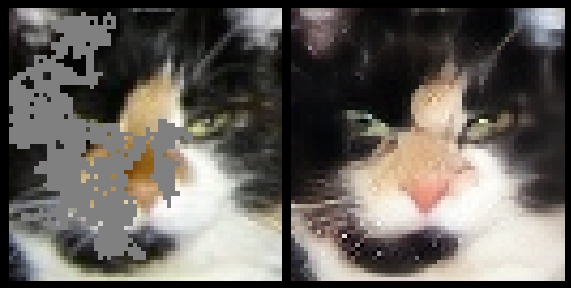
scheduler: None

* 1. About inpainting strategy

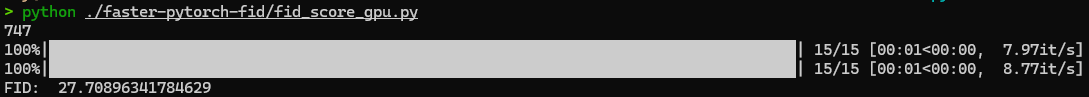
當初有些好奇為什麼照片要做一次次跌代慢慢修復，因此實驗增加了2種修復策略。

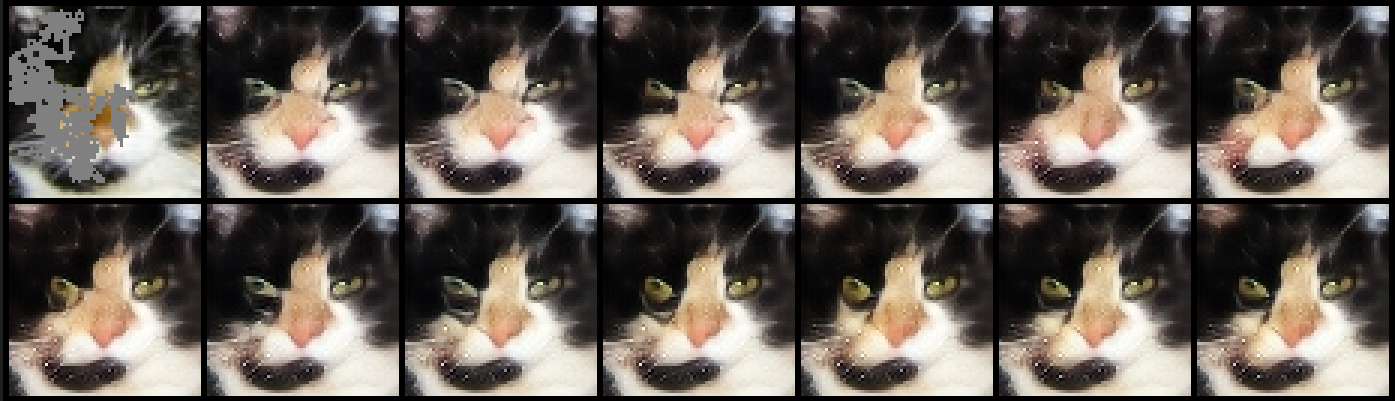
* god: 一次全部修復





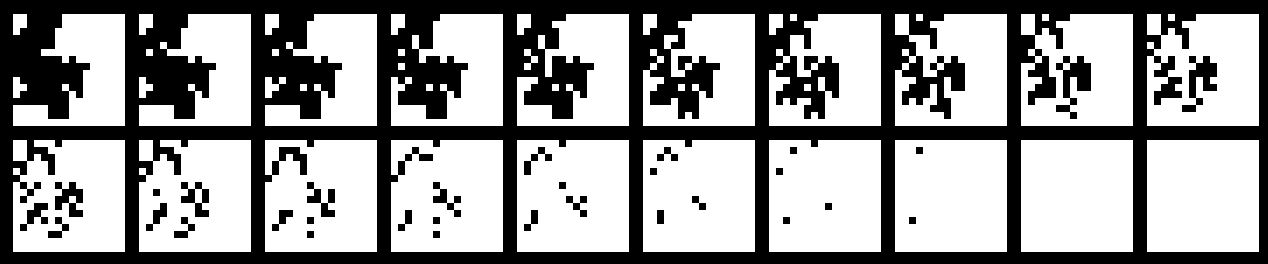
* stair: 一次修復10%



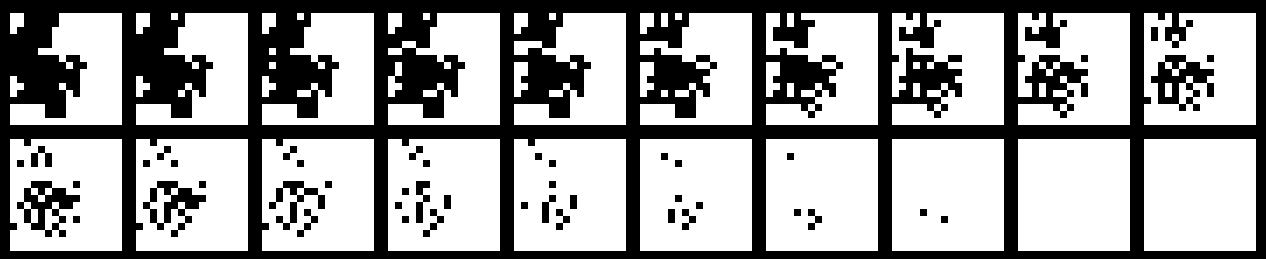


看似策略不比其他的還要差，推測主要的影響因素還是取決於model。

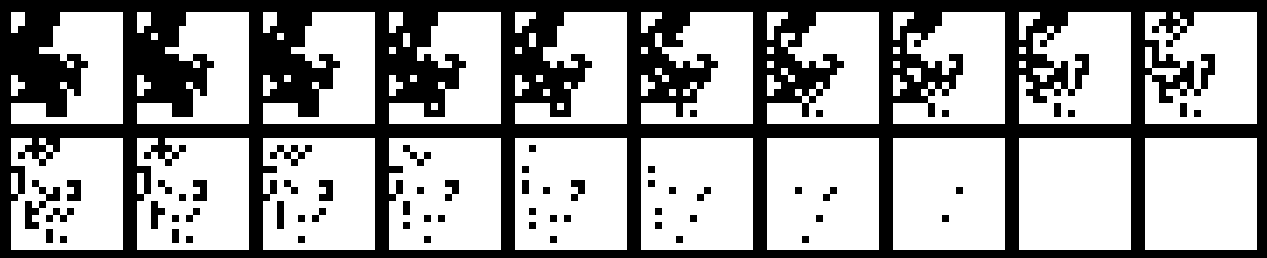
1. Experiment Score
   1. iterative decoding
      1. Mask in latent domain
         1. cosine



* + - 1. linear



* + - 1. square

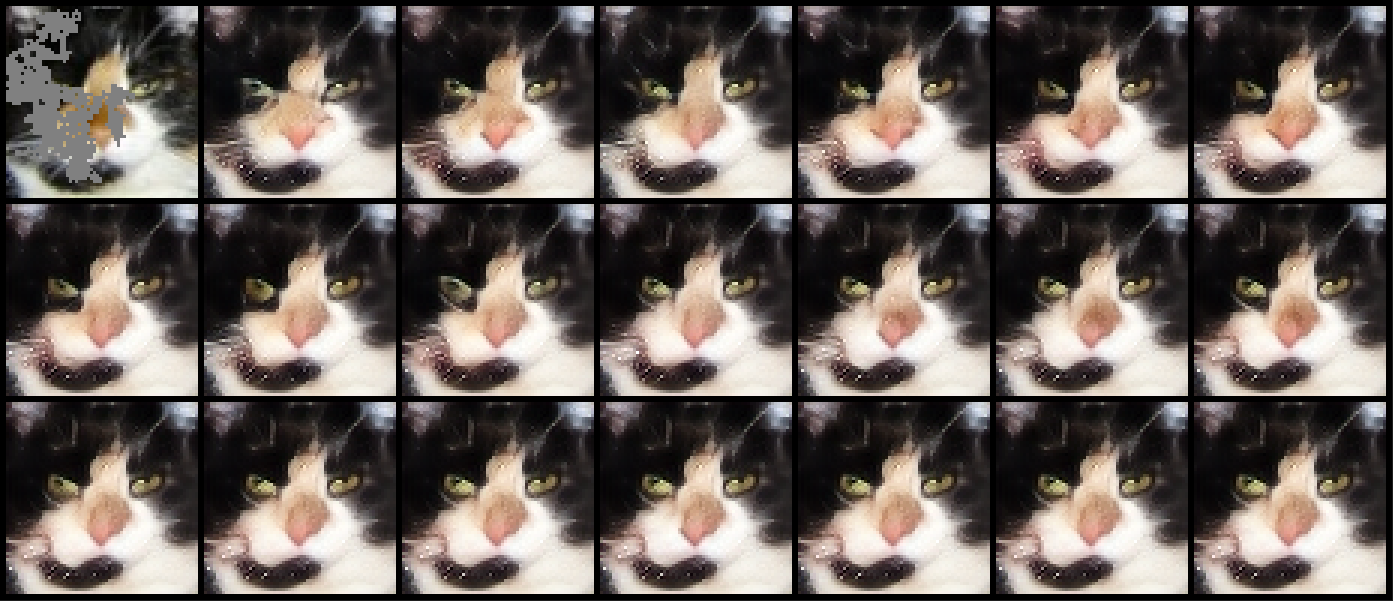


可以看到不同的inpainting策略在最後都有完成所有猜測。

* + 1. Predict Image
       1. cosine



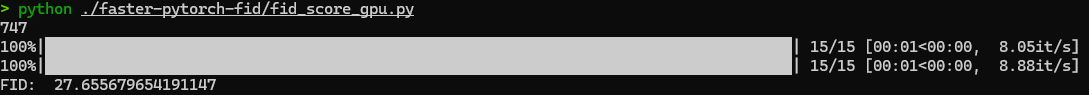
* + - 1. linear



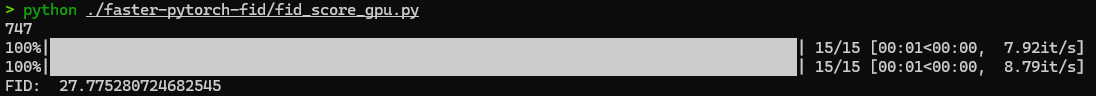
* + - 1. square



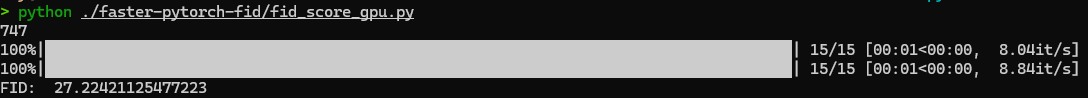
* 1. Best fid score
* 最高的fid為square **27.22**
  + 1. cosine



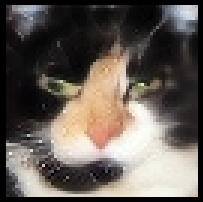
* + 1. linear



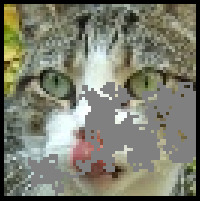
* + 1. square



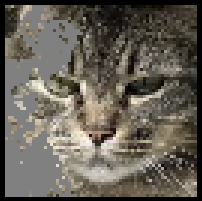
* + 1. masked image v.s. result











* 1. setting

Training: 在workspace下執行

python training\_transformer.py

Inpainting: 在workspace下執行

python inpainting.py

fid score: 在workspace下執行

python ./faster-pytorch-fid/fid\_score\_gpu.py

(**不用cd進去**)