影像處理 HW3

109550087 單宇晟

這次作業我選擇實機練習的第一個: *E-NeRV: Expedite Neural Video Representation with Disentangled Spatial-Temporal Context*來進行操作

* **論文介紹、原架構缺點及可改善之處**

NeRV (Neural Representation for Videos)是一種圖像隱式神經表示法，也就是上次作業實作的方法，是為了解決傳統INR (Implicit Neural Representation)的一些限制才被設計出來的。然而，NeRV還是有一些缺點，那就是隨著通道維度的增加，整個model的大小也會迅速增加，導致出現很多不必要的parameter。因此，這篇paper提出了一種名為E-NeRV (Expedite Neural Video Representation)的方法來改善這些缺點。

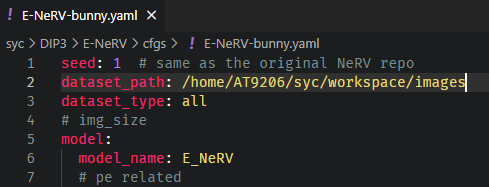
在NeRV的架構中，發現MLP的最後一層其實很冗長。為了解決這個問題，E-NeRV提出了spatial-temporal資訊的disentangled formulation。E-NeRV並不像原本NeRV那樣，直接從input中生成frame feature map，而是把spatial和temporal的context分開。接下來，為了把兩者重新結合，會進行element-wise multiplication。如此一來，就可以知道spatial-temporal representation。

另外，NeRV因為做了subsequent pixel-shuffle operation，導致trainable weight的數量的大幅增加。作者在這裡對NeRV的block進行一些調整，他們使用維度較小的兩個consecutive convolution kernels取代原本的convolution kernel，並改在中間做pixel-shuffle operation、加入intermediate dimension。

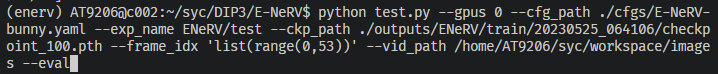
整體來說，NeRV和E-NeRV都使用十分類似的架構，只是E-NeRV對特定部分做了修改，透過上面提到的disentangling和fusion，還有對NeRV結構的改善，E-NeRV減少了NeRV中的redundancy，甚至提高了性能和收斂速度，也保持了較少的參數數量。至於未來的方向，作者打算把這個方法運用到其他下游的任務，像是optical flow estimation和video super-resolution。

* **實機練習**

由於這次github的指示並沒有寫得像上次NeRV那篇清楚，我首先了解了作者code的大概架構，並將data path改成自己的資料(bunny.yaml裡)，同時也調整成我想跑的epoch數量(100)，而最重要的資料集，我是用和HW2一樣的照片，方便進行比較。

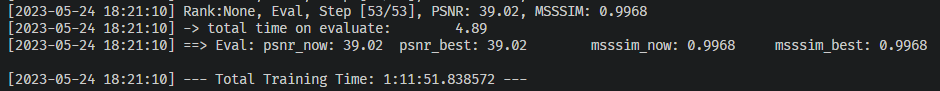


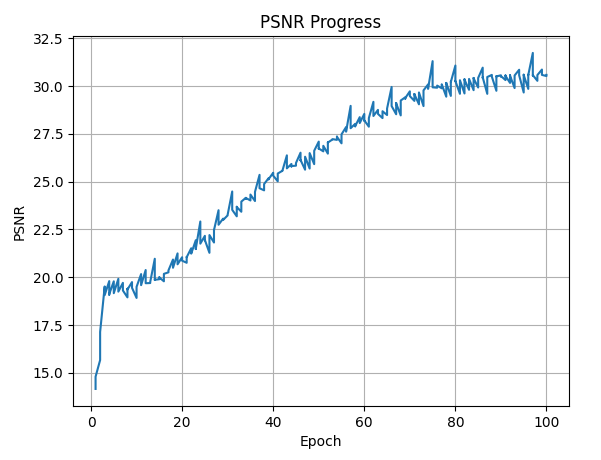
接著，我參考網路上的code重新將main.py改寫，分成了train.py以及test.py，剩下的部分則沒有更改，並按照指令輸入到terminal去執行。



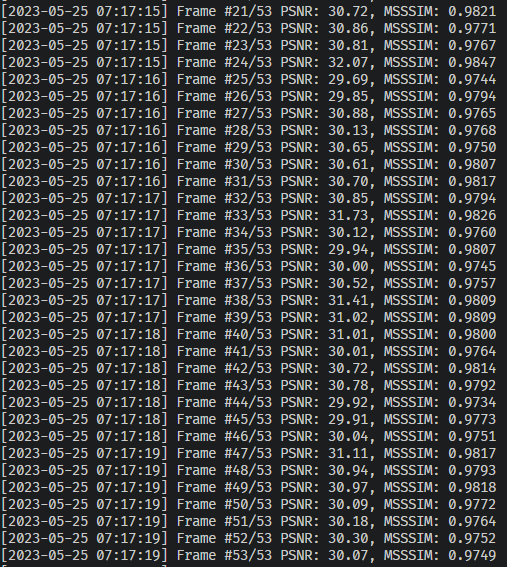
* **模型訓練與評估結果**

*train*





*test*

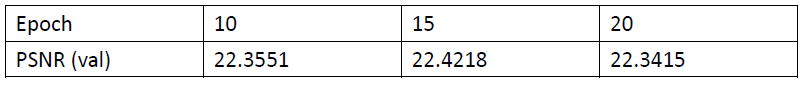


* **分析與討論**

透過PSNR Progress可以看到，在training的前幾個epoch，psnr上升的十分迅速，直到將近epoch 100才趨於穩定，足見此方法的有效性。另外，雖然這次的epoch數達到100，但實際的訓練時間也不超過2小時，比起NeRV的訓練時間縮短了很多很多，上次作業epoch數量只有20，但卻需要4-6個小時才能訓練完成，我覺得這跟最一開始paper裡提到的NeRV的缺陷有關，由於我所準備的圖片畫質較高，可能會造成在training的過程，input的dimension提高，讓NeRV model整體的參數數量過大，訓練時長也才會那麼久，而E-NeRV則改善了這方面。

最後，雖然E-NeRV輸出的結果不是gif，而是一張張圖片，但藉由跟上次的圖片比較，整體的清晰度甚至還有所上升，從psnr也可看出來，上次我的PSNR只到22左右，這次卻到了30上下。總體而言，我認為E-NeRV藉由spatial和temporal的運用，以及block的重新安排，對於原本設計的改善與提升是十分明顯的，也讓我了解到其實只要對神經網絡的架構、組成做一些微調，或是在feature extraction上運用不同的方法，就能對結果造成十分巨大的改變。

*HW2 NeRV結果*



*NeRV result*



*E-NeRV result*

