

- Experiment a.

- Assumption

Multi-image optimization-based approach 是透過多張不同拍攝角度的低解析度照片(拍攝角度要非常接近但不一致)，並採用以下公式來求得一張高解析度照片：


$$E(X) = (\sum_i ||D(X \otimes B_i) - Y_i||) + \lambda ||\nabla X||_{tv}, i \in [0, 7] \quad (5)$$



$$||\nabla X||_{tv} = \sum_p \sqrt{X(p)_x^2 + X(p)_y^2}. \quad (6)$$

我認為餵入越多張照片，可以為像素鄰近點帶來更多資訊，對高解析度照片的生成是有幫助的，會讓此高解析度的照片品質更好，但如上式(5)所示，越多張照片意味著目標函數內的 \sum 運算次數增加，故生成一張高解析度照片所需時間也會變得較長。

- Justification

為了驗證我的假設，我分別用 2 張、4 張、6 張照片做運算，為了讓實驗更嚴謹，我挑選的 4 張照片有 2 張是剛剛前面挑過的 2 張；挑選的 6 張照片有 4 張是剛剛前面挑過的 4 張。並記錄運算時間，輸出的照片再分別與 original HR image ('./reference/HR_zebra_test.png')計算 PSNR 值。結果如下表：

	result	PSNR	執行時間
2 images		20.8333 dB	669 sec

4 images		23.6862 dB	1107 sec
6 images		25.0554 dB	1536 sec

由上表可看出使用照片張數越多，PSNR 值越高，代表越接近真正的高解析度照片。且我的實驗方式是原本的照片量再多挑兩張的方式做多照片疊加運算，並不是隨機挑選 2 張 4 張 6 張的方式，因為隨機挑選照片可能因為某張低解析度照片對高解析度的資訊提供特別有幫助，那這樣就有失實驗嚴謹性。不過張數越多造成計算量增加，所以運算時間也隨之增加。驗證我的 Assumption。

- Experiment b.

- 專案程式碼

<https://drive.google.com/drive/folders/1i217eeqIKRGytM6mr1uaUVkf6msmVZtl?usp=sharing> (要複製連結去瀏覽器貼上，直接點超連結可能會出錯)




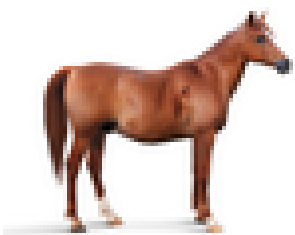








- Assumption

因為此兩個 model(net_F16B2_epoch_15.pth & net_F64B8_epoch_100.pth)所用的訓練資料類別太單調，model 學到的多半是斑馬的紋路與草坪的紋理，而 image_hidden/內的圖片的某些物體的紋理是 model 沒學到的，所以推測餵入 image_hidden/內的圖片給 model 會產出類似斑馬紋理或是草坪紋理的樣子出現，並且因為訓練次數的不同，越多訓練次數的 model 理論上學的越紮實，故推測 net_F64B8_epoch_100.pth 所產出的 HR image 會比

net_F64B8_epoch_100.pth 所產出的 HR image 看起來更銳利。

- Justification 1


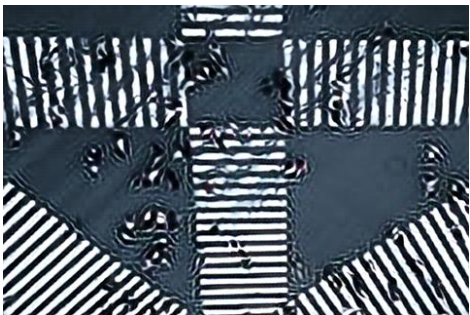




由下表可看出 model 所產出的圖片有些許奇怪，像是斑馬線上的行人已經被融入成斑馬條紋的感覺，所以會有點扭曲，另外像是模糊的小狗圖片經過 model 產出的 HR image 相當奇怪，也有一點斑馬紋路的感覺，而熊貓圖與小狗圖的草坪紋理以及馬的紋理在高解析度下的處理都不錯，因為 training dataset 內就有這些物件的紋路可以學習。而在不考慮圖片的奇怪性，net_F64B8_epoch_100.pth 所產出的 HR image 的確比 net_F16B2_epoch_15.pth 所產出的 HR image 更佳清晰銳利，上述驗證了 Assumption。

Original LR image	net_F16B2_epoch_15.pth	net_F64B8_epoch_100.pth
		
		
		
		

- Justification 2

由 Justification 1 看出小狗圖與馬路圖的 HR image 表現極差，所以我找了 3 張小狗的圖片與 3 張斑馬線的圖片當作 training & validation set 來重新訓練 model，而 training & validation set 的圖片之高解析度圖片就是原圖，低解析度圖片採用 opencv.resize 將長寬各除以 4 來

得到。我訓練了兩個 model，分別是小狗的 SR model 與斑馬線的 SR model。不單靠一個 model 訓練的原因是這種 dense to dense problem 有別於一般的 classification problem，如果把所有類別的 data 全部丟入同一 model 做訓練，對 super resolution task 來說會有過多的雜訊，產出的 HR image 效果會不佳，這種 dense to dense task 傾向先採用 classifier 辨識場景或類別，再透過該單一類別的 model 來處理。而我設置的參數與 Implementation 1.4 相同， $\{n_Fear, nResBlock, nEpoch\} = \{64, 8, 100\}$ 。結果如下：

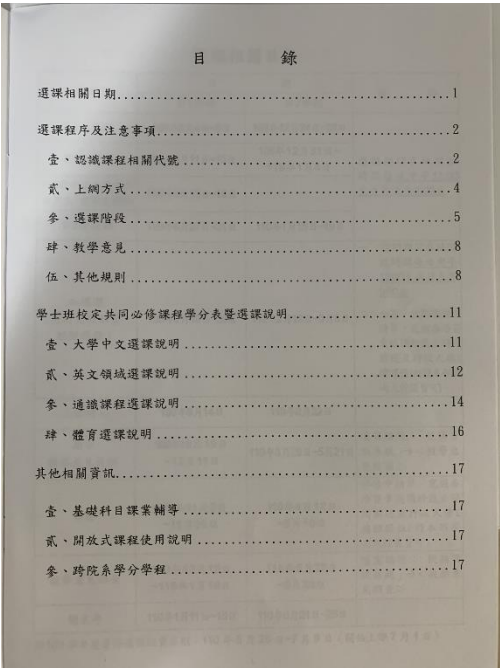
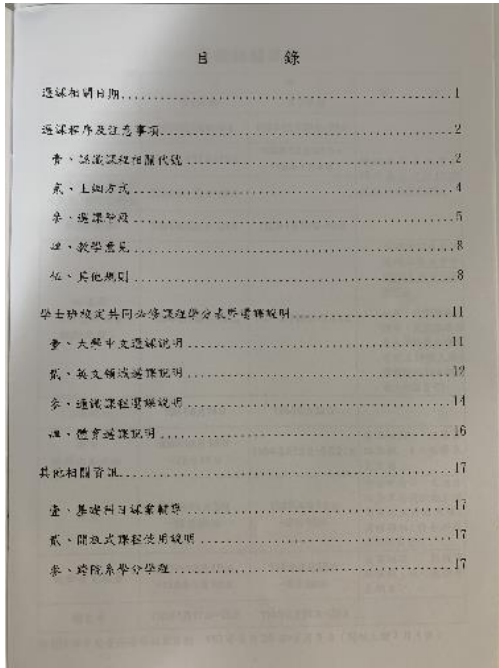
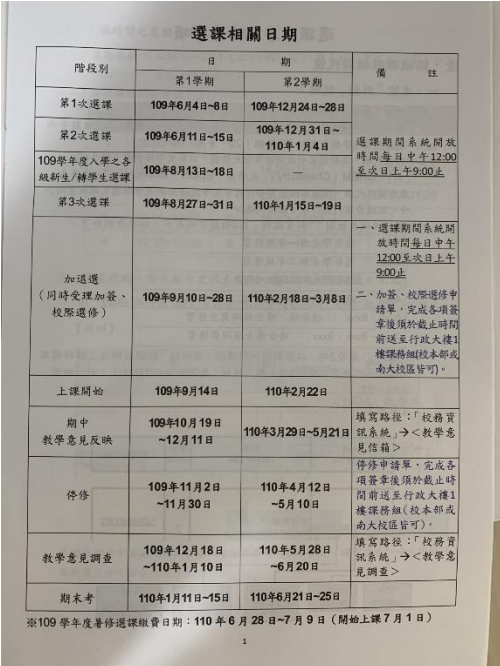
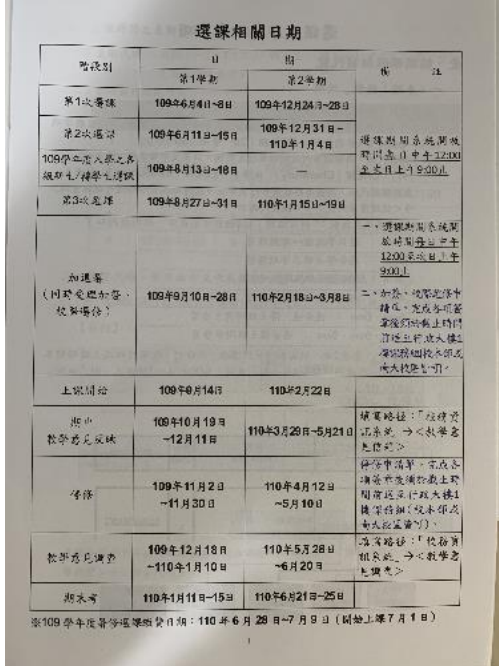
Original LR image	原本 Implementation 的 model 產出(長寬各放大四倍) $\{n_Fear, nResBlock, nEpoch\} = \{64, 8, 100\}$	新訓練資料集的 model(長寬各放大四倍) $\{n_Fear, nResBlock, nEpoch\} = \{64, 8, 100\}$
		
		

可看出新訓練資料集的 model 在超參數不變的情況下，效果都比原本的 model 好，所以驗證了資料集對 model performance 有很大影響。新 model 產出的 HR image 雖可看出是小狗群與斑馬線，但還是稍顯模糊，我推斷是我 model 餵入的訓練圖片過少(模仿 Implementation 只餵入兩張)以及訓練圖片本身解析度不高所導致。

- Experiment c. (該專案程式碼位置位於 Experiment c/資料夾)
 - 專案程式碼
<https://drive.google.com/drive/folders/18nWhqMND7ucNsGXnSwmxPCaOcw3WMLs?usp=sharing> (要複製連結去瀏覽器貼上，直接點超連結可能會出錯)

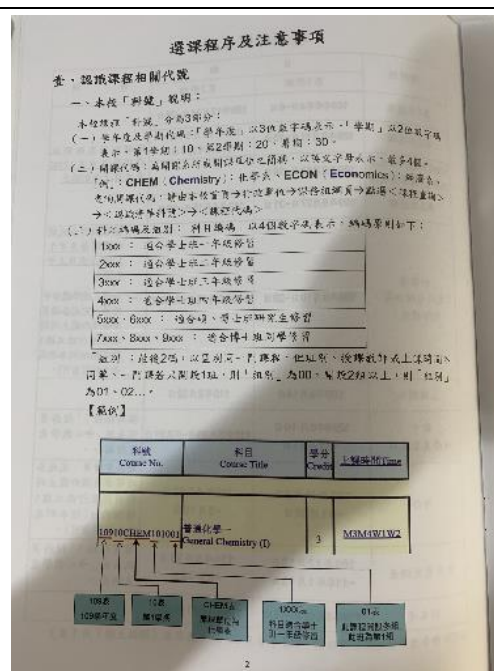
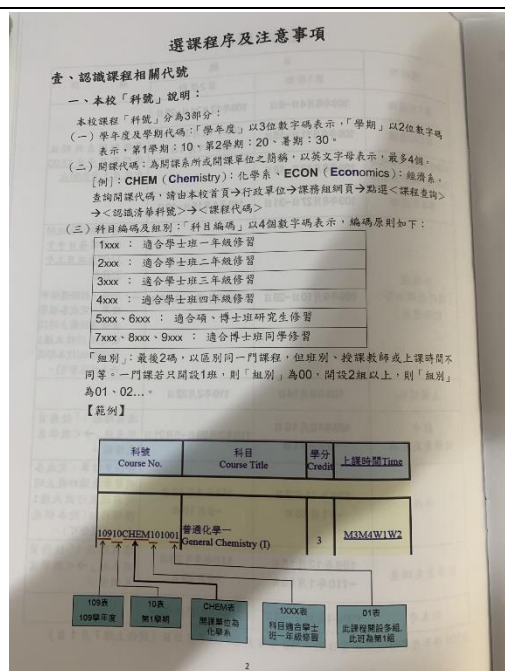
● Assumption

課堂中老師有提到 **super resolution task** 中最難做到的就是一些整齊的線條紋理或是文字，因為這種類型的圖片只要解析度不高，很容易被肉眼察覺。所以我打算挑戰看看處理文字方面的照片，看效果如何。下表分別是我採用的 **training data & validation data & testing data**。(做法都是將手機拍攝之照片做 **opencv.resize**，**HR image** 是將原始長寬各除以 2，**LR image** 是將原始長寬各除以 8，故跟 **implementation part** 一樣是實作 4 倍長寬縮放的 **super resolution task**)

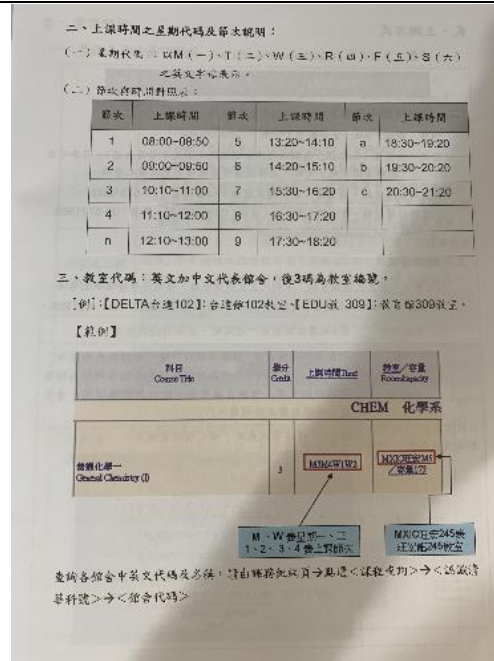
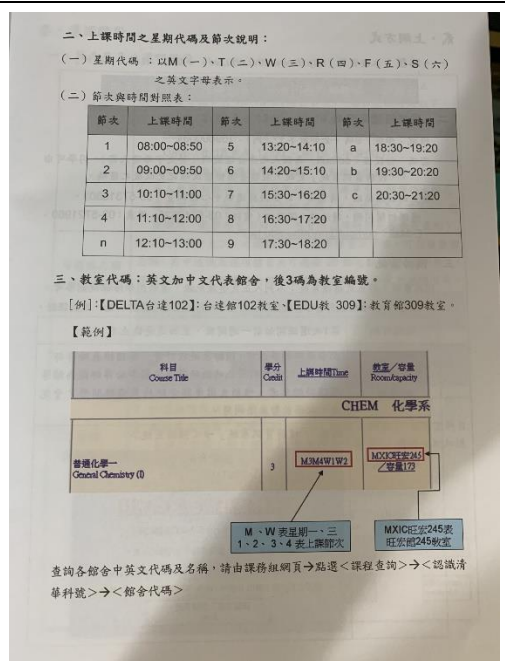
	HR image	LR image
Training data 1		
Training data 2		

Validation data

1



Testing data 1

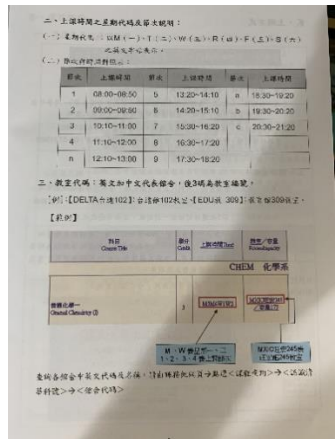
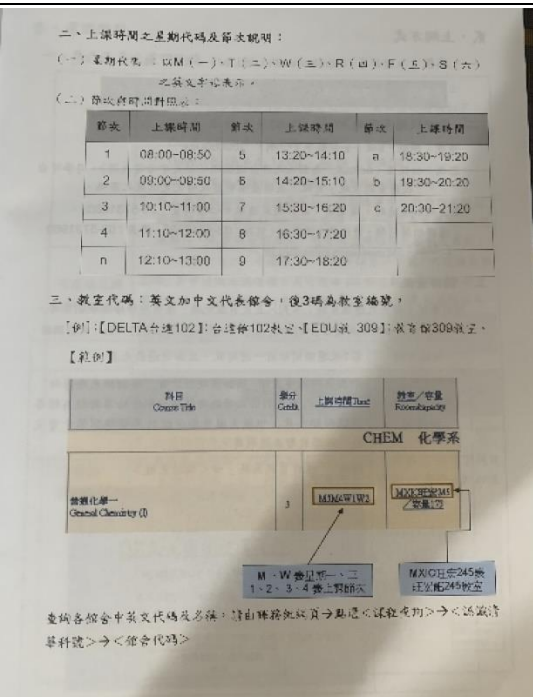
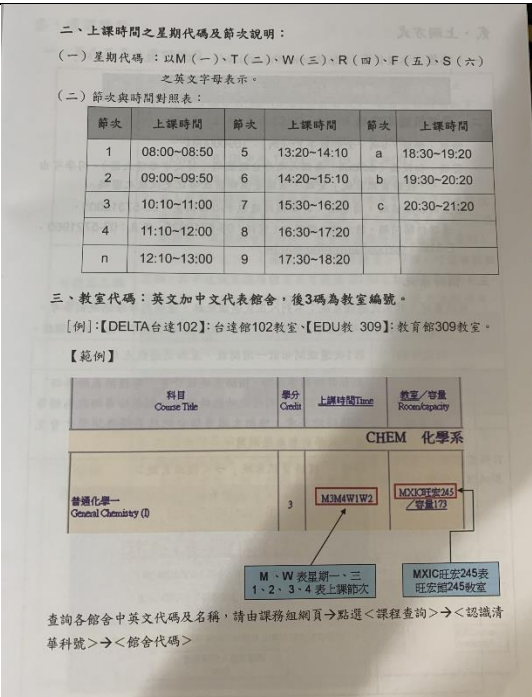


上面所有照片都從同一本書所拍攝的，故性質雷同，預期 testing data 效果會不錯，並且推測若 learning rate 設置妥當，每次更新的 model 其 validation set 之 avg. psnr 會越來越大。理論上最後一次更新的 model 是最後要用來做 testing 的模型。

● Justification

我的超參數設置與 implementation 1.4 相同，{n_Fear, nResBlock, nEpoch} = {64, 8, 100}。而我用這 100 個 model 來檢驗看看哪個 model 所產出的 HR image 與 reference data 計算 PSNR 值最高，結果竟然是第 5 次 update 的 model 表現最好，我知道這樣的方法是作弊的方式，不符合 model training rule，正確的方式應該是在 update model 的過程

中，看哪次對 validation data 的 PSNR 值最佳就選用該 model 來做 testing，但是我在訓練過程中忘記紀錄每次 update model 對 validation data 的 PSNR 值，這是我這次實驗的瑕疵。不過這邊我就將錯就錯，將第五次更新的 model(最佳的模型)其餵入 testing data 之 output 來做比較：

LR image	Model output	Reference(HR image)
		

Model output 與 HR image 計算 psnr 值為 24.0884，可以發現一些簡單筆畫的國字與數字表現得還不錯，但是較複雜筆畫的國字其表現就不太好，另外像是原始圖片上的紅框處，經過 model 處理過後變成灰框，原因是因為 training dataset 中沒有紅色的部分，都只有黑色白色居多，所以 model 沒有學到紅色方面的 information。

● Free Study 1

● 專案程式碼



https://drive.google.com/drive/folders/1N9iDFwQhQbxE_5v4ktMzqlUY8H9RHGAK?usp=sharing (要複製連結去瀏覽器貼上，直接點超連結可能會出錯)

● Assumption 1

N 的大小決定 kernel size，kernel size 為 N 的平方，當 N 越大 kernel 就會越大，會讓每次卷積層對鄰近點的涵蓋範圍變大，進而造成最後計算出來的圖片較平滑較模糊，物體邊緣變得較不銳利

● Justification 1

為了驗證 Assumption 1，我固定 std 與 lambda，只改動 N 的值，結果如下：

N=13, std=1.2, lambda=1e-2	N=5, std=1.2, lambda=1e-2
	



上面右圖的 N 值較小，可發現斑馬紋路的部分較不平滑，有點鋸齒狀的感覺，故驗證 Assumption 1

- Assumption 2

Std 為先給定的 prior，代表對該圖在 gaussian kernel 範圍內其像素點之間變異程度的預先推論值。所以當 std 越高，代表 kernel 範圍內像素變異量大，此時的 gaussian 鐘型曲線分布會讓人覺得較大，就類似於 Assumption 1 中提到的 kernel size 變大的效果，會讓計算出來的圖片有模糊化的感覺，並且物體邊緣會較不銳利。

- Justification 2

為了驗證 Assumption 1，我固定 N 與 lambda，只改動 std 的值，結果如下：

N=13, std=1.2, lambda=1e-2	N=13, std=4.5, lambda=1e-2
	

可發現 std 放大 3-4 倍時，畫面變得有點模糊，甚至斑馬紋路有點

鋸齒狀的感覺，高解析的圖像銳利感也消失了。因此驗證 Assumption 2。

- Assumption 3



$$E(X) = \left(\sum_i \|D(X \circledast B_i) - Y_i\| \right) + \lambda \|\nabla X\|_{tv}, \quad i \in [0, 7] \quad (5)$$

$$\|\nabla X\|_{tv} = \sum_p \sqrt{X(p)_x^2 + X(p)_y^2}. \quad (6)$$

Lambda 控制了上式(5)的後面 L1 norm 的大小，當 lambda 增加，為了達到 objective function 的目標，predicted image gradient 的 L1 norm 就必須縮小，而 gradient 的值被限縮代表每次更新的幅度變小，或是說不完全往 objective function 的目標去更新，而這樣會造成在相同的更新次數下，輸出影像的一些高頻或細節處會變得較模糊(不銳利)。

- Justification 3

為了驗證 Assumption 3，我固定 N 與 std，只改動 lambda 的值，結果如下：

N=13, std=1.2, lambda=1e-2	N=13, std=1.2, lambda=1e-1
	

我將 lambda 值放大十倍，這兩張圖差異不大，但是越遠離攝影機的斑馬條紋，右邊 lambda 較大的結果較模糊，因此驗證 Assumption 3

- Free Study 6

- 專案程式碼

<https://drive.google.com/drive/folders/1e9MnB1Ob0aXgJlslHIZ8EFAI9XlVG4tS?usp=sharing> (要複製連結去瀏覽器貼上，直接點超連結可能會出錯)



- Assumption

因為神經網路屬於 data driven model，而我認為 Implementation

1.4 中 training & validation set 樣本數過少，若提升訓練與驗證照片的張數，讓 model 學到更多東西(紋理、顏色...)，應該有助於 model performance 提升。

- Justification

原本 Implementation 1.4 的 training set 與 validation set 各只有 2 張照片與 1 張照片，我又找尋其他斑馬&草原相關的圖片，讓 training set 與 validation set 各有 10 張照片與 5 張照片。為了實驗的公平性，我只改變資料集的照片量，其他參數都不變， $\{n_Fear, nResBlock, nEpoch\} = \{64, 8, 100\}$ ，重新訓練模型，並且都是用第 100 次更新的模型來做 testing，結果如下表：

Implementation 1.4 PSNR = 19.0344 (與 original HR image 相比)	Implementation 1.4(more data) PSNR = 19.0833 (與 original HR image 相比)
	

肉眼其實看不太出來差異，不過 PSNR 確實有一點提升，但提昇的幅度不大，若想要有更大幅度提升，我認為可以嘗試看看不同的 optimizer，例如 Adam、Adagrad，以及根據更新次數調整 learning rate(Implementation 1.4 中是固定的 learning rate)...這些都有機會讓 model performance 提升。