

利用人工智慧提升圖片解析度

專題學生：林峰正
黃仰辰
張耀文

指導教授：葉慶隆 教授



大同大學
資訊工程學系

專題報告

中華民國 112 年 1 月

利用人工智慧提升圖片解析度

大同資工系專題報告

林峰正
黃仰辰
張耀文

大同大學
資訊工程學系
專題報告

利用人工智慧提升圖片解析度

林峰正

黃仰辰

張耀文

經考試合格特此證明

專題考試委員

指導教授

華慶隆
謝尚琳
卓世明

華慶隆

中華民國 112 年 1 月 3 日

概要

現今雖有高階設備提供高解析度的相片，但難免會因為各種因素導致相片成像不如預期，這時就須藉由軟體輔助以提升相片解析度。然而，目前已有許多提升相片解析度的研究與實驗，但有些研究提升了解析度的同時，卻損失了原始相片該有的細節以及真實性。

因此，我們以 Super-Resolution Generative Adversarial Network (SRGAN) 為基礎進行修改，並參考 Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) 和 Inception v3 的部分概念，目的希望除了提升解析度的同時維持該有的細節與真實性外，也希望在有限的記憶體空間下融入這兩個論文的部分概念，使提升相片解析度成效比 SRGAN 好，而最終實驗結果證實了我們有達到此目的，並且在我們使用的硬體設備下，我們模型訓練速度也比 SRGAN 快。

致謝

感謝本專題的指導教授-葉慶隆教授，通過大三整學期的雜誌研讀與教授授課的

人工智慧相關課程，使我們熟悉了 Python 的語法及神經網路相關資源的利用，

為製作專題打下了基礎，並在數次的群組會議中幫助我們確認專題的方向。

目錄

概要.....	i
致謝.....	ii
目錄.....	iii
表目錄.....	v
圖目錄.....	vi
第 1 章 緒論.....	8
1.1 背景與動機.....	8
1.2 目的及方法步驟.....	9
1.3 專題專案管理.....	9
1.4 專題成果.....	11
1.4.1 圖片品質比較.....	11
1.4.2 訓練時間比較.....	12
1.4.3 穩定模型訓練.....	12
第 2 章 文獻探討.....	14
2.1 生成對抗網路.....	14
2.2 SRGAN.....	15
2.2.1 BN 層.....	15
2.2.2 殘差塊.....	16
2.3 ESRGAN.....	17
2.4 GoogLeNet.....	17
2.5 評價指標.....	18
2.5.1 RMSE.....	18
2.5.2 PSNR.....	18
2.5.3 SSIM.....	19
2.5.4 FSIM.....	19
第 3 章 問題分析與系統設計.....	21

3.1 問題分析.....	21
3.1.1 GAN (Generative Adversarial Network)	21
3.1.2 SRGAN (Super-Resolution Generative Adversarial Network)	22
3.1.3 ESRGAN (Enhanced Super-Resolution Generative Adversarial Network)	24
3.1.4 SRGAN 的問題分析	25
3.2 系統設計.....	26
3.2.1 去除生成器所有 BN 層	26
3.2.2 修改 RRDB.....	27
3.2.3 修改生成器整體結構.....	33
3.2.4 判別器.....	34
第 4 章 神經網路實作與結果.....	36
4.1 建立神經網路模型.....	36
4.2 開發工具與環境.....	37
4.3 模型訓練.....	38
4.4 相關研究之比較.....	40
4.4.1 生成圖片比較.....	40
4.4.2 評估指標比較.....	42
4.4.3 訓練速度比較.....	44
4.5 網站.....	45
4.5.1 網站功能.....	45
4.5.2 網站架構.....	46
4.5.3 網頁與模型連結.....	47
4.5.4 網頁結果呈現.....	48
第 5 章 結論.....	52
參考文獻.....	54

表目錄

表 1-1 : 時間進度表	10
表 1-2 : 工作分配表	10
表 1-3 : SRGAN 與我們模型(Ours)平均評估指標	11
表 1-4 : 平均訓練時間	12
表 3-1 : Set5 dataset 平均評估指標	43
表 3-2 : Set14 dataset 平均評估指標	43
表 3-3 : Urban100 dataset 平均評估指標	44

圖目錄

圖 1-1 : 去除 BN 層的 SRGAN 與我們模型(Ours GAN)的生成器損失值變化	13
圖 1-2 : 去除 BN 層的 SRGAN 與我們模型(Ours GAN)的判別器損失值變化	13
圖 3-1 : GAN 基本運作流程	22
圖 3-2 : SRGAN 運作流程	23
圖 3-3 : MSE 與原圖關係[15]	23
圖 3-4 : SRGAN 生成器結構	24
圖 3-5 : SRResNet 基本結構[5]	24
圖 3-6 : Residual-in-Residual Dense Block (RRDB)	25
圖 3-7 : 將 SRGAN 生成器去除 BN 層	27
圖 3-8 : Inception v3	28
圖 3-9 : 淺層 inception	30
圖 3-10 : 中層 inception	30
圖 3-11 : 深層 inception	31
圖 3-12 : Residual-in-Residual Inception Module (RRIM)	31
圖 3-13 : RRIM 內部結構	32
圖 3-14 : 修改後的生成器結構	33
圖 3-15 : Sub-pixel convolution 示意圖[12]	34
圖 3-16 : SRGAN 判別器結構[2]	35
圖 4-1 : 生成器模型結構	36
圖 4-2 : 生成器訓練流程圖(左)與判別器訓練流程圖(右)	39
圖 4-3 : 生成器訓練損失值變化	40
圖 4-4 : 判別器訓練損失值變化	40
圖 4-6 : Set5 dataset 結果比較圖	41
圖 4-7 : Set14 dataset 結果比較圖	42
圖 4-8 : Urban100 結果比較圖	42
圖 4-9 : 模型訓練時間比較	44

圖 4-10：網站架構圖	47
圖 4-11：Django Folder Structure	48
圖 4-12：網頁首頁(Home)畫面	49
圖 4-13：提升解析度網頁畫面	49
圖 4-14：參考文獻網頁畫面	49
圖 4-15：選擇圖檔準備上傳畫面	50
圖 4-16：上傳完並提升解析度後畫面	50
圖 4-17：預覽早期利用雙三次插值(Bicubic)提升解析度圖	50
圖 4-18：預覽我們作品提升解析度圖	51
圖 4-19：成功下載時畫面	51

第 1 章 緒論

1.1 背景與動機

提升相片解析度的應用範圍廣，如醫學成像、衛星成像、手機拍照都有提升解析度的需求。雖然現今的設備已經能提供相當高的解析度，能夠拍出非常清晰的照片，但難免會因各種因素而造成解析度不夠高，如拍攝光線不足、拍攝距離過近或過遠等，且並不是任何人都擁有較好的設備，因此這時候就需藉由軟體來對相片提升解析度。

現今提升解析度的主要做法有運用數學的雙三次插值法、運用人工智慧的 GAN[1] 及 CNN[16] 模型，還有像是通過銳化、去噪等後期圖像處理方法，其中本專題打算以人工智慧的方法來提升相片解析度。然而，提升解析度之神經網路在現今已有眾多研究，但部分研究存在一個重要缺點，就是提升解析度的同時會損失相片細節與真實性，簡單來說就是相片成效不夠自然，其原因是因為這些研究是依像素間數值關係，而非從人對相片的感受觀點來生成相片。在 2016 年 SRGAN [2] 研究團隊先行提出以貼近人的觀感生成高畫質相片概念，因

此我們以 SRGAN 進行發想，嘗試結合多個概念來寫出新的提升解析度神經網路。

1.2 目的及方法步驟

本專題藉由改進 SRGAN 來實作提升圖片解析度的神經網路模型，通過輸入低解析圖片，使模型能夠生成出輸入圖片的高解析版本。

我們在閱讀與提升圖像解析度和 SRGAN 相關的論文及討論的文章，從中得知它仍存在一些需要改進的地方，所以我們從這些資料中選擇了 ESRGAN[5] 及 GoogleNet[6] 做為參考依據，擷取它們部分的優點用來修改 SRGAN 模型，同時為了方便參考開源的提升解析度相關程式碼，因此挑選 TensorlayerX[8] 作為我們開發模型的框架，最後我們通過數十次實驗與改進架構，配合與原 SRGAN 的成果進行比較，設計出了提升解析度效果較好的模型。

1.3 專題專案管理

本專題從 111 年 4 月底開始規劃，並於 7 月中旬正式進行研究，該專案分為三步驟進行，從發現問題到研讀相關論文，最後著手進行改進與測試，小組三人

分頭資料收集、編寫程式、Debug 及網頁呈現，最終歷時 7 個多月完成專案。

表 1-1：時間進度表

年月 工作項目	2022						2023
	7 月	8 月	9 月	10 月	11 月	12 月	1 月
研讀蒐集資料							
編寫程式							
網頁製作							
專題報告							

表 1-2：工作分配表

工作內容	工作分配		
	林峰正	黃仰辰	張耀文
蒐集、研究資料	10%	50%	40%
修改與測試模型	25%	5%	70%
模型訓練	100%	-	-
程式 Debug	90%	-	10%
網頁製作	15%	-	85%
論文製作	10%	25%	65%

1.4 專題成果

我們模型相較於 SRGAN 模型，生成圖片品質較好(1.4.1 小節)、訓練速度也較快(1.4.2 小節)。此外，也克服了去除 BN 層[3]所帶來的訓練不穩定的問題(1.4.3 小節)。

1.4.1 圖片品質比較

我們將 Set5[14]、Set14[14]、Urban100[14]這三個資料集內的所有圖片，利用 RMSE、PSNR、SSIM[17]和 FSIM[18]這四個評估指標產生的評估數值，進行加總平均，以比較 SRGAN 與我們模型的圖片品質。表 1-3 中，粗體字表示對於該評估指標而言，此模型生成圖片的品質較好。

表 1-3 : SRGAN 與我們模型(Ours)平均評估指標

模型 評估指標	SRGAN	Ours
RMSE	0.0041	0.0040
PSNR	24.3758	24.4915
SSIM	0.6892	0.6947
FSIM	0.6168	0.6205

1.4.2 訓練時間比較

表 1-4 為 SRGAN 模型與我們模型每訓練 100 次所花費的平均訓練時間，以及訓練 400 次的總花費平均訓練時間。

表 1-4：平均訓練時間

模型 訓練次數	SRGAN	Ours
Epoch 0 ~ 99	137.63s	136.59s
Epoch 100 ~ 199	138.43s	139.11s
Epoch 200 ~ 299	137.22s	138.16s
Epoch 300 ~ 399	141.41s	139.41s
Total average	138.67s	138.32s

1.4.3 穩定模型訓練

圖 1-1 和圖 1-2 分別為去除 BN 層的 SRGAN 與我們模型在訓練時，生成器

(Generator)和判別器(Discriminator)損失值的變化狀況。可發現去除 BN 層的

SRGAN 在 step 500 多時，生成器和判別器損失值變動幅度開始逐漸增加，這代

表模型無法正常學習。

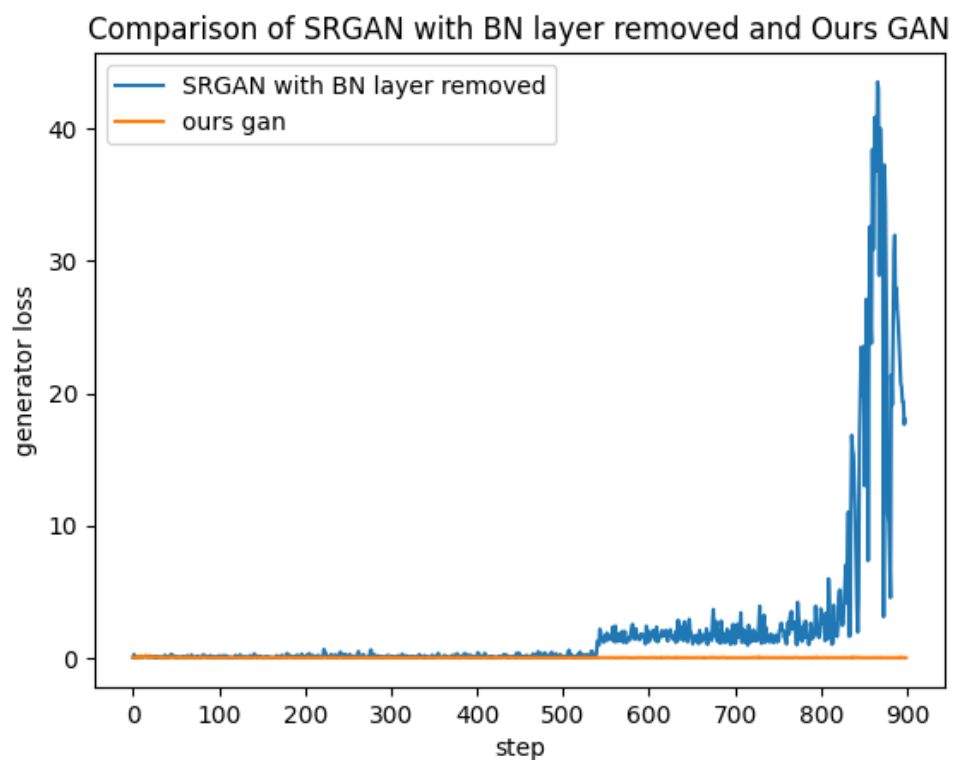


圖 1-1：去除 BN 層的 SRGAN 與我們模型(Ours GAN)的生成器損失值變化

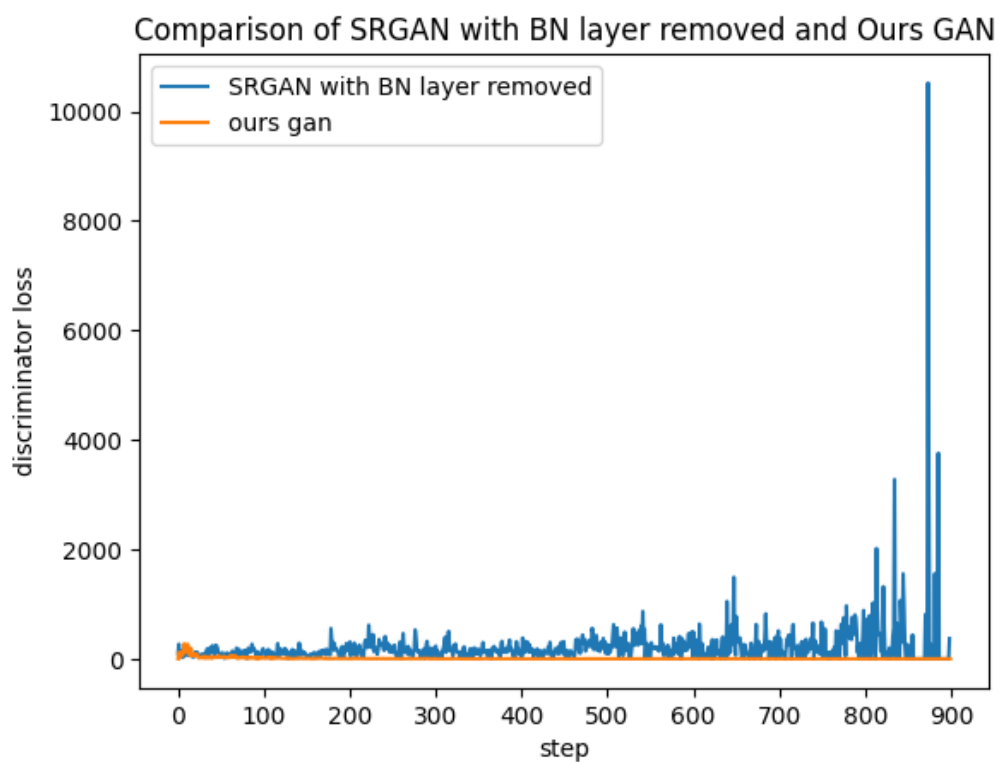


圖 1-2：去除 BN 層的 SRGAN 與我們模型(Ours GAN)的判別器損失值變化

第 2 章 文獻探討

於本章節中，2.1 節介紹的是本專題需要先了解的生成對抗網路，2.2 節介紹的是本專題主要修改對象的 SRGAN 模型，2.3 節及 2.4 節介紹的則是本專題主要參考對象 ESRGAN，與 Google 團隊提出的圖像識別模型 GoogLeNet。最後 2.5 節是我們用來評估模型效果的評價指標。

2.1 生成對抗網路

生成對抗網路(Generative Adversarial Network, GAN) [\[1\]](#)是一種用於生成模型的深度學習架構，由生成器(Generator)和判別器(Discriminator)兩個模型組成，被廣泛地運用在圖像生成、自然語言處理等領域。

生成器的輸入數據通常是隨機噪聲(noise)，目標是盡可能生成逼真或者新的數據，而判別器通常是輸入真實數據的樣本與生成器輸出的數據，目標是盡可能辨別出真實數據與生成數據的差異，並給予真實數據正面反饋，生成數據則是負面反饋。GAN 是藉由生成器試圖生成更加真實的數據，判別器試圖提高辨別是否為真實數據的準確度，使兩個模型以類似於對抗形式來訓練並學習。

2.2 SRGAN

Super-Resolution Generative Adversarial Network (SRGAN) [2]是一種用於圖像超解析度重建的 GAN 模型，應用於包括圖像放大、圖像質量和清晰度提升圖像超解析度重建的各種任務。

該模型生成器的輸入不再是隨機噪聲，而是輸入低解析圖像，目標是生成出原輸入圖像的高解析度版本，而判別器的任務則是需要辨別出輸入的兩個圖像分別是真實的或是生成的高解析度圖像，其中生成器使用了類似 Residual Network (ResNet) [4]作為主要神經網路架構，這樣有助於訓練時保留原始高分辨率圖像的高頻細節，對於圖像的紋理和結構的重建更加還原真實，同時在該架構中還大量使用了 BN 層。

2.2.1 BN 層

BN 層 [3] (Batch Normalization Layer)是一種正規化方法，因為正則化的運算是針對每個 mini-batch，能降低對特定數據的依賴增加模型泛化能力，同時也減少了過擬合的問題。

另一方面，正則化的動作也經常用於解決深度學習模型中經常出現的梯度消失或爆炸的問題，但 BN 的使用由於針對每個 mini-batch 去計算，這會導致計算量增加，再加上正則化對於較小批量的數據可能會產生噪聲影響結果，因此 BN 層的使用是根據模型的需求再進行添加的。

2.2.2 殘差塊

殘差塊[4] (Residual block)是一種常見的深度學習模型架構，最早出現於著名的深度學習模型 ResNet 中，主要用於解決深層神經網絡中的梯度消失和擴展性問題。殘差塊的架構非常簡單，主要由兩個相同的卷積層與跳躍連接組成，其中跳躍連接指的是第二個卷積層的輸出特徵與進入第一個卷積的輸入特徵相加，這樣做的好處是，即使在深層模型中出現了梯度消失問題，前面的輸出仍然會被直接加到後面的輸入上，因此模型仍然可以有效地更新參數。

另外，殘差塊具有很好的網路擴展性，多個殘差塊串聯起來會形成所謂的殘差模塊，而殘差模塊與殘差模塊又可以串聯起來，這樣的架構可以實現更加深層的網絡結構。

2.3 ESRGAN

Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) [5]是一種

在 SRGAN 的基礎上進行改進的圖像超分辨率重建的生成對抗網絡。

ESRGAN 與 SRGAN 除了判別器的輸出方式不同外，該論文還改進了

SRGAN 生成器的幾個部分，其中他們認為在提升解析度時使用 BN 層會影響原

圖像特有的細節，甚至可能產生偽影，因此去除了對 BN 層的使用，但此作法

會造成模型訓練的不穩定，所以他們同時也改進了殘差塊的結構，而本專題主

要參考了以上兩種想法。

2.4 GoogLeNet

GoogLeNet [6]是一種深度 Convolutional Neural Network (CNN)，由 Google 在

2014 年的 ImageNet 比賽中提出。該網路的特點是它使用了一種稱為 Inception

的架構，使得在進行卷積時獲取多樣性的特徵，另外再搭配一些正規化技術，

使得 GoogLeNet 成為一個能夠兼具網路深度與寬度的圖像辨識模型。

Inception module 常被用於圖像分類任務的神經網路中，主要的思考是使用

多個不同尺寸的卷積核來萃取圖像中不同尺寸的特徵，也就是拓展了網路寬

度。Google 團隊當初對於該架構有持續研究並改進，到目前共有 5 種版本：

Inception v1~v4、Inception-Resnet，其中本專題模型中的所有 Inception module

架構皆是參考了 Inception v3 [\[7\]](#)，包含透過分解卷積減少計算量的方法，如 3x3

拆成 1x3 和 3x1，同時也學習利用不同尺寸的卷積核來萃取多樣尺寸的特徵，

而 Inception module 的輸出就是疊加這些特徵成變成一個特徵圖。

2.5 評價指標

2.5.1 RMSE

Root Mean Squared Error (RMSE)，是數學與統計學中均方根公式的應用，原理

是利用預測值和實際值之間的差先進行平方，再求平均之後取平方根。

RMSE 通常用來衡量回歸問題中預測值和實際值之間的差異，但也可以用

於衡量預測圖像和實際圖像之間的差異，其中 RMSE 的值越小，表示預測值和

實際值的差異越小，模型的預測能力越強。

2.5.2 PSNR

Peak Signal-to-Noise Ratio (PSNR)是一個表示訊號最大可能功率和影響它的表示精度的破壞性雜訊功率的比值，它將最大像素值的平方作為參考值，計算圖像或視頻信號的 MSE 與參考值的比值，再將比值轉換成對數形式求得。

PSNR 常用於衡量影像品質或視頻壓縮算法的評估，也常被用來量化圖像重建誤差的大小或失真程度，其中 PSNR 的值越大，表示圖像品質越好，或者說是圖像重建誤差越小。

2.5.3 SSIM

Structural Similarity Index (SSIM Index)[\[17\]](#)是一個用來評估影像品質的指標，該評價指標是考慮到影像的結構信息，同時針對人類視覺系統的影像感知特性進行模擬而得出。

SSIM 的計算方法包含了衡量影像間的亮度、對比度和結構相似性，通過對這三個方面的比較和加權，計算出一個在 0 到 1 之間的值，其中越接近 1 表示影像品質越好，或者說預測影像與實際影像的結構相似性高。

2.5.4 FSIM

Feature Similarity Index Measure (FSIM)[\[18\]](#)是一個用來評估影像品質的指標，該指標通過提取影像的多種特徵信息，像是亮度、對比度、結構和邊緣等，通過計算預測影像與實際影像的相位、梯度變化，綜合考量這些特徵之間的關聯性與權重來進行評估，其中 FSIM 計算後的值在 0 到 1 之間，越接近 1 表示影像品質越好。

第 3 章 問題分析與系統設計

3.1 問題分析

因為這個專題的目的是想要提升圖像的解析度，所以我們針對有用於提升解析度的模型去研究，其中發現近期許多人提出相較於單一 CNN[16] 架構，GAN[1] 架構能更還原圖像的自然與真實性。而後找到了 SRGAN[2] 和 ESRGAN[5] 作為我們這次的主要核心，為了做出分別，我們選擇了以去除 BN 層[3] 的方式進行提升圖片解析度，但這當中卻面臨到了梯度爆炸的問題，於是我們引入了 GoogLeNet[6] 來協助改善此問題。

本節先介紹 GAN 的基本運作和學習方式(3.1.1 小節)，再介紹本專題主要修改對象 SRGAN 模型的運作，和選擇使用 GAN 在超解析度(Super-Resolution)任務上(3.1.2 小節)。最後再由 ESRGAN 分析 SRGAN 的問題，來推導出我們要修改的方向及順序(3.1.3 小節)。

3.1.1 GAN (Generative Adversarial Network)

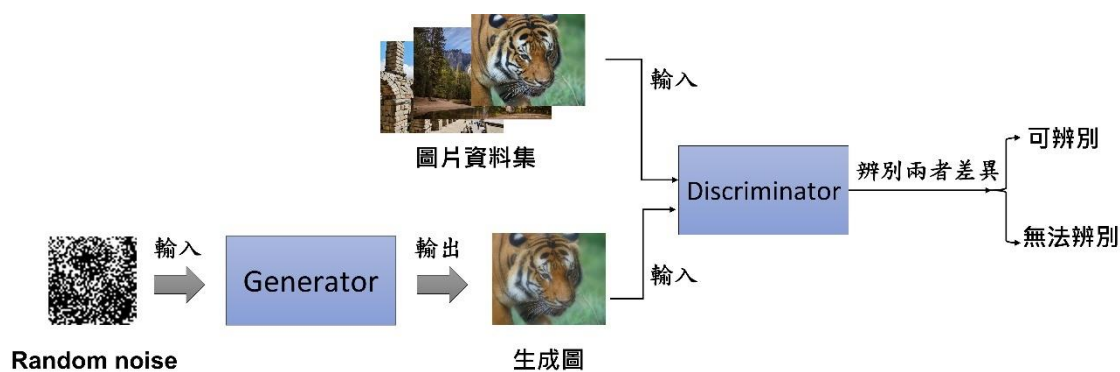


圖 3-1 : GAN 基本運作流程

GAN 由生成器(Generator)和判別器(Discriminator)兩個神經網路架構組成(圖 3-

1)，生成器接收一個隨機噪聲(Random noise)作為輸入，並生成與訓練資料集相

似的數據。判別器接收訓練資料集數據和生成數據作為輸入，並試圖區分兩者

差異。

生成器的學習目標是生成讓判別器難以區分與訓練資料集數據差異的逼真

數據。判別器的學習目標是區分出訓練資料集數據和生成數據，使生成器的生

成品質越好。兩者以互相對抗的方式進行學習，從而不斷的提升生成器性能。

3.1.2 SRGAN (Super-Resolution Generative Adversarial Network)

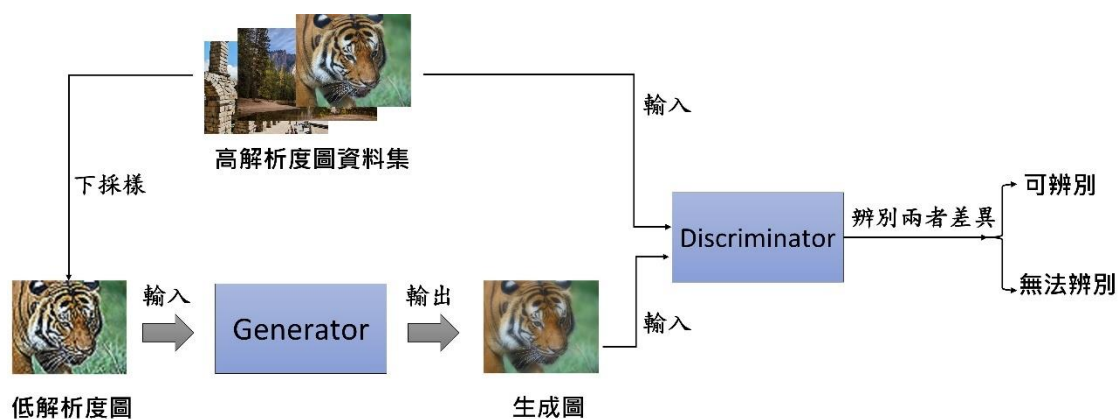


圖 3-2 : SRGAN 運作流程

SRGAN 與基本 GAN 運作流程主要差異是，生成器輸入從隨機噪聲改成經過下

採樣(Downsampling)產生的低解析度圖(圖 3-2)。而 SRGAN 選擇 GAN 運用在超

解析度任務上是因為相較其他類型神經網路，透過判別器與生成器互相對抗能

生成出更加真實的圖像，看起來更加自然。

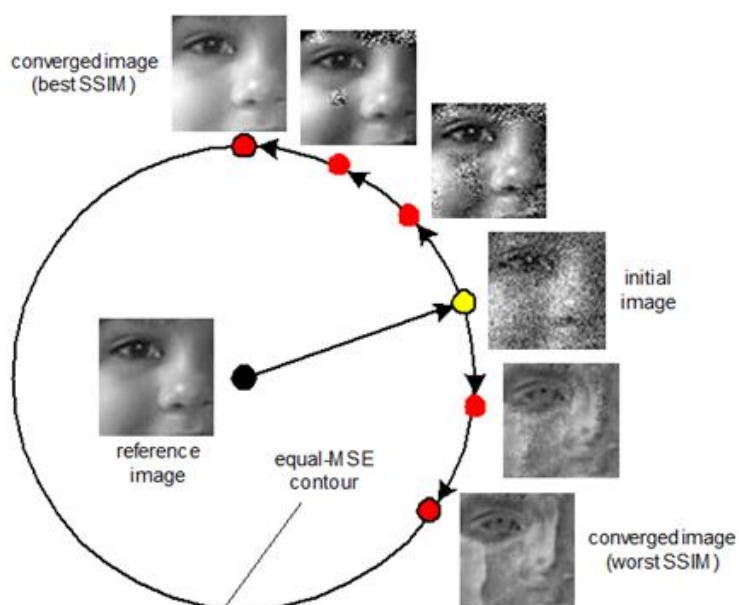


圖 3-3 : MSE 與原圖關係[15]

此外，由於 SRGAN 還引入感知損失(Perceptual loss)，所以能夠保留更多的圖像細節與結構，這使 SRGAN 生成更加逼真更高品質的圖片。因為大多超分辨率神經網路都類似於 SRCNN，此類型神經網路一般都單純使用均方誤差(MSE)做為生成器的損失值。雖然使用 MSE 能讓生成圖片與資料集圖片產生的峰值訊噪比(PSNR)數值高，但對於人的感官而言不一定是最好的結果。從圖 3-3 中可以看出對 MSE 而言，圓圈外的圖與圓圈內的圖 MSE 值一樣，但從人的感官卻是正上方的圖比較好。

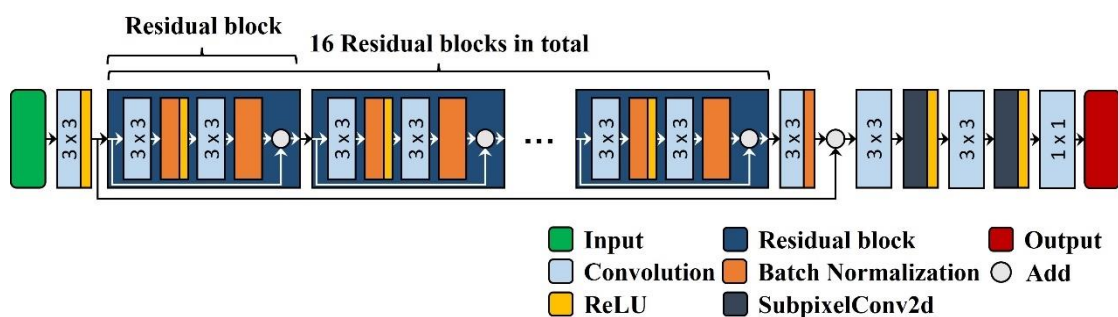


圖 3-4 : SRGAN 生成器結構

3.1.3 ESRGAN (Enhanced Super-Resolution Generative Adversarial Network)

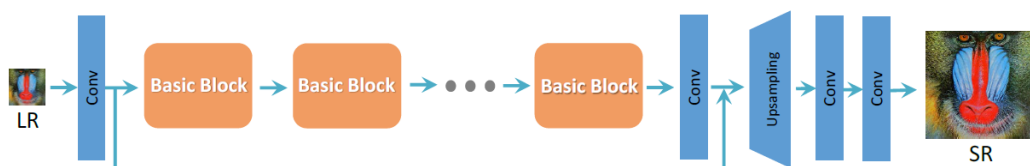


圖 3-5 : SRResNet 基本結構[5]

ESRGAN 將 SRGAN 生成器重構成以 SSResNet 為基礎的結構(圖 3-5)，並去除

SRGAN 生成器使用的 BN 層(圖 3-4)，以及將殘差塊(Residual Block)使用

Residual-in-Residual Dense Block (RRDB)組成(圖 3-6)。

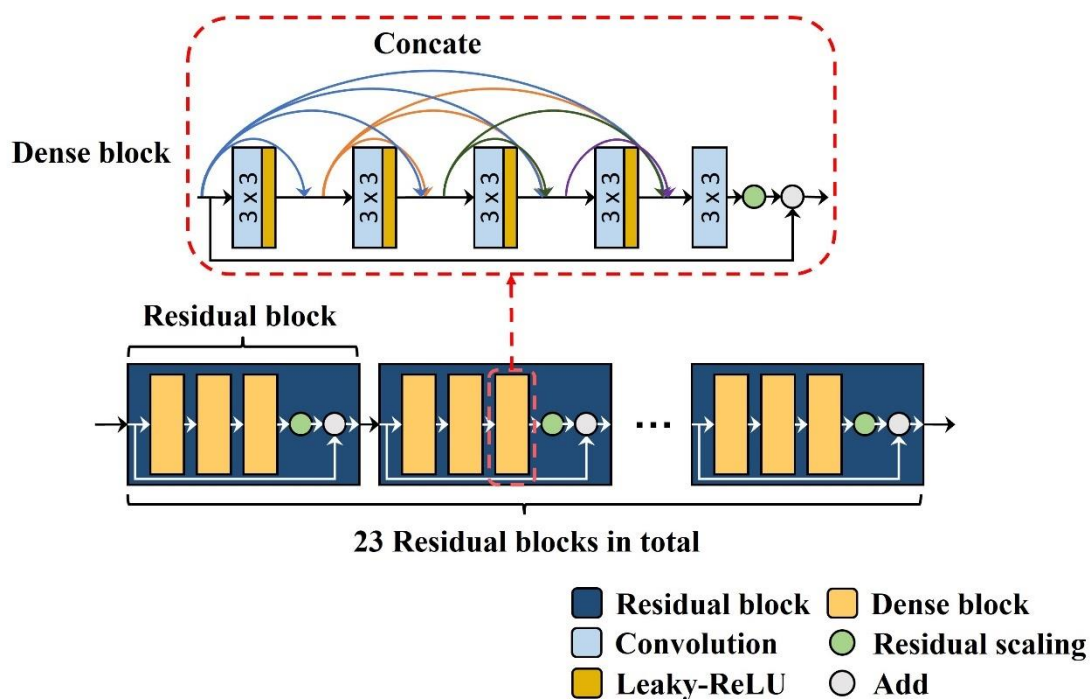


圖 3-6 : Residual-in-Residual Dense Block (RRDB)

ESRGAN 對生成器更改的主要原因是因為，去除 BN 層會提高生成器的泛化能力、減少計算複雜度以及減少占用的記憶體，且當訓練集和測試集內容差異大的時候，生成的圖片容易產生偽影。此外，為了避免去除 BN 層所導致的模型訓練不穩定問題，所以 ESRGAN 才使用 RRDB 結構來穩定。

3.1.4 SRGAN 的問題分析

鑒於 ESRGAN 對 SRGAN 的看法，SRGAN 主要問題在於生成器中，BN 層導致的生成圖片品質不好以及模型訓練效率的問題。

因此我們學習 ESRGAN 去除 BN 層、RRDB 的使用和重構生成器整體結構這三個方向的修改。而因為去除 BN 層造成的訓練不穩定問題，會導致模型無法正常學習，所以修改核心為 RRDB。

因此修改的順序為：

1. 去除生成器所有 BN 層。
2. 修改 RRDB 以提升生成圖片品質，同時穩定模型訓練。
3. 隨著 RRDB 的改變，修改生成器整體結構。

3.2 系統設計

根據問題分析所推導出的模型修改順序，系統設計流程為去除生成器所有 BN 層、修改 RRDB、修改生成器整體結構。

3.2.1 去除生成器所有 BN 層

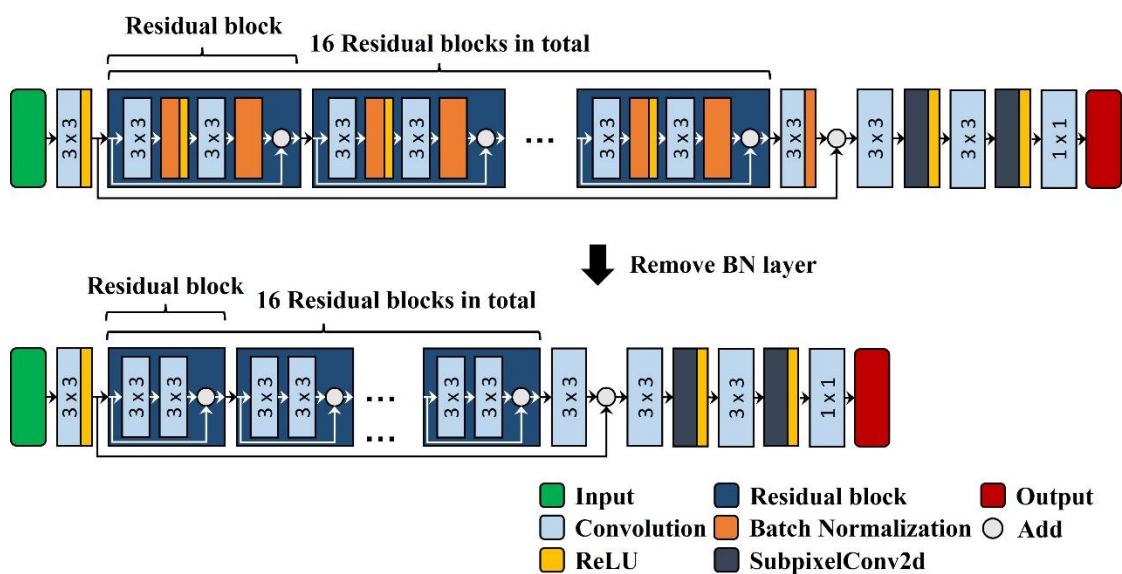


圖 3-7：將 SRGAN 生成器去除 BN 層

圖 3-7 中，上半圖為 SRGAN 生成器整體結構，下半圖為去除所有 BN 層的生成器整體結構。

3.2.2 修改 RRDB

我們修改 RRDB 目的除了穩定模型訓練，也要提升生成圖片品質。而提升生成圖片品質的部分，我們參考了 GoogLeNet[6]提出的 Inception v3[7]架構的設計方式。

Inception v3 :

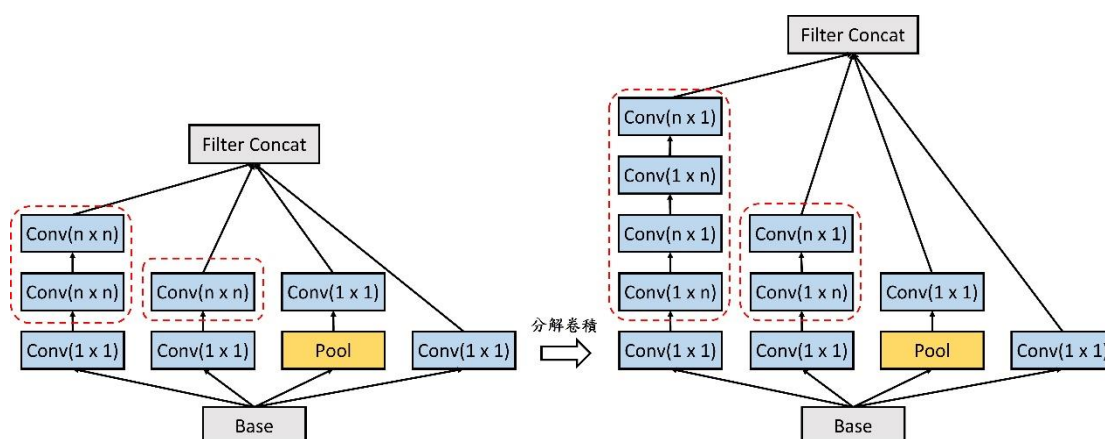


圖 3-8 : Inception v3

Inception v3(圖 3-8)概念是利用分解卷積(Factorizing Convolutoins)，使在不降低

神經網路效率的情況下減少計算複雜度，同時降低過擬合可能性。此外

Inception v3 利用將多個同個輸入但不同尺寸和不同深度的特徵圖進行串接，以

輸出更具多樣性特徵的圖片，使圖片品質提高。

我們修改 Inception v3 的三個考量：

我們設計了淺層、中層、深層的 inception module，這些 module 都遵循了以下考

量。

1. 記憶體容量問題：不論是神經網路的寬度或深度都會影響記憶體容量的占

比，因此要對 Inception v3 內部使用的卷積層數量要做取捨。

2. 不使用 Pool : Inception v3 原本是使用於辨識圖片的神經網路中，但我們神

經網路的目的是提升圖片解析度，所以盡可能的保留所有特徵值。而 Pool

會丟棄一些特徵值，所以與我們的目的相反。

3. 卷積層的 kernel size：神經網路深度越深越適合較小的 kernel size；反之，

越淺層越適合較大的 kernel size。因此在修改 Inception v3 時，卷積層使用

的 kernel size 會逐漸縮小。

設計 inception module：

在設計過程中，發現若在淺層位置構造過於複雜，則生成器無法有效的學習正

確的特徵值，因此對淺層、中層、深層的 inception module 設計，會隨著神經網

路的深度而逐漸複雜。

所以根據修改 inception module 的三個考量，以及隨著神經網路深度逐漸增

加結構複雜度，因而設計出淺層 inception (圖 3-9)、中層 inception (圖 3-10)、深

層 inception (圖 3-11)。

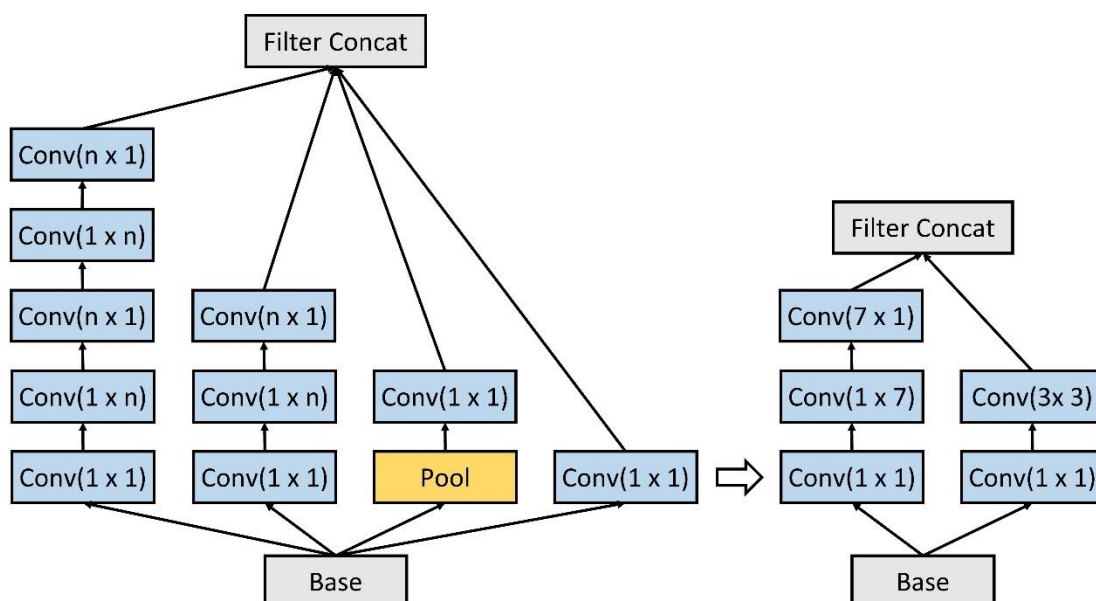


圖 3-9：淺層 inception

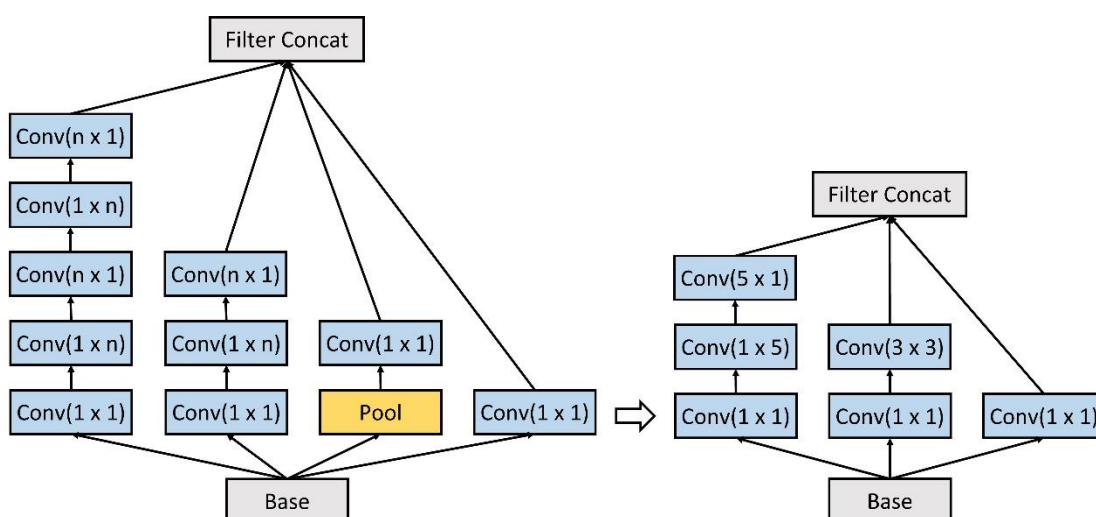


圖 3-10：中層 inception

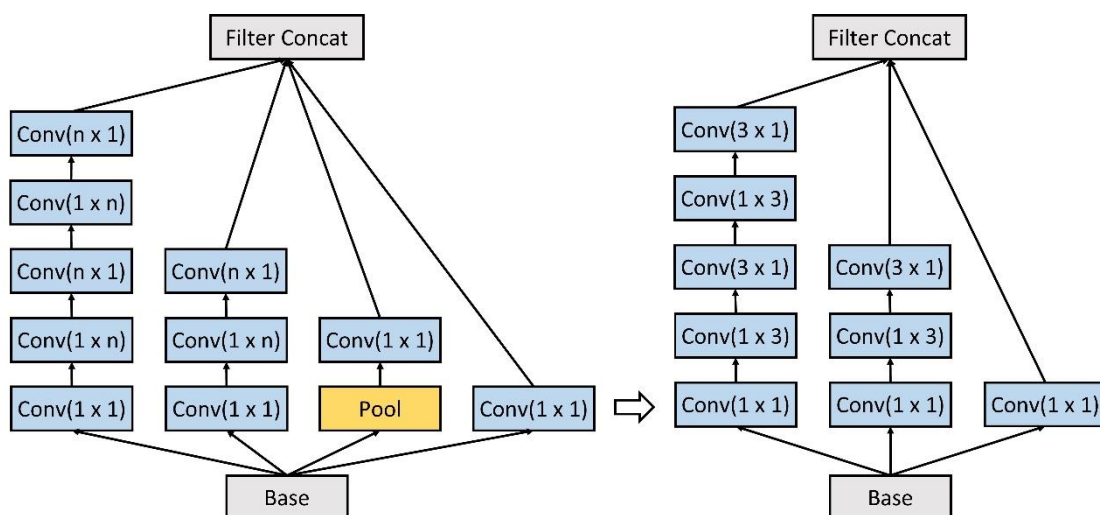


圖 3-11：深層 inception

修改後的 RRDB 結構：

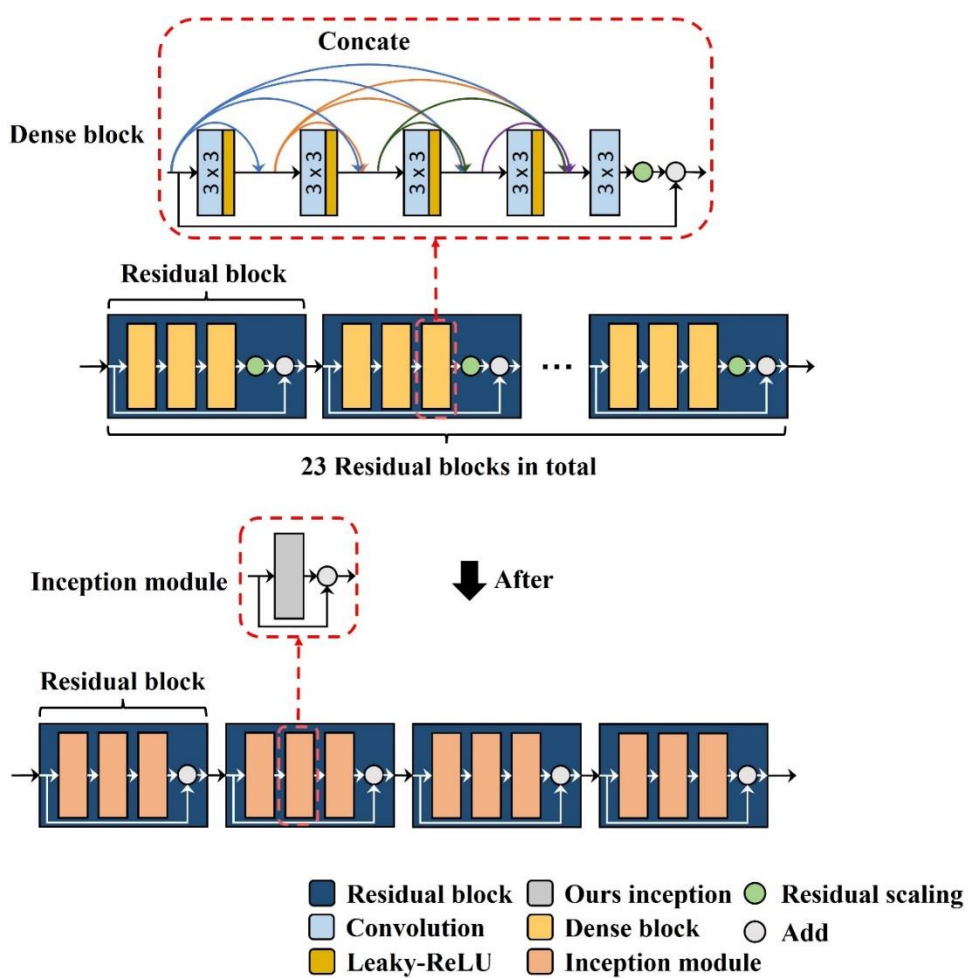


圖 3-12：Residual-in-Residual Inception Module (RRIM)

圖 3-12 上半部，RRDB 結構在 Residual block 內加入三個 Dense block，並在進行殘差前都會將圖片的所有特徵值乘以 0.2，此動作稱為 Residual scaling，目的是減少梯度爆炸的風險。圖 3-12 下半部，修改後的 RRDB 結構將 Dense block 內部以一個 inception module 取代，並因為特徵值縮小對於我們的架構而言，會降低生成圖片品質，因此將所有殘差前面乘以 0.2 的運算刪除。而修改後的 RRDB 結構，我們稱為 Residual-in-Residual Inception Module (RRIM)。

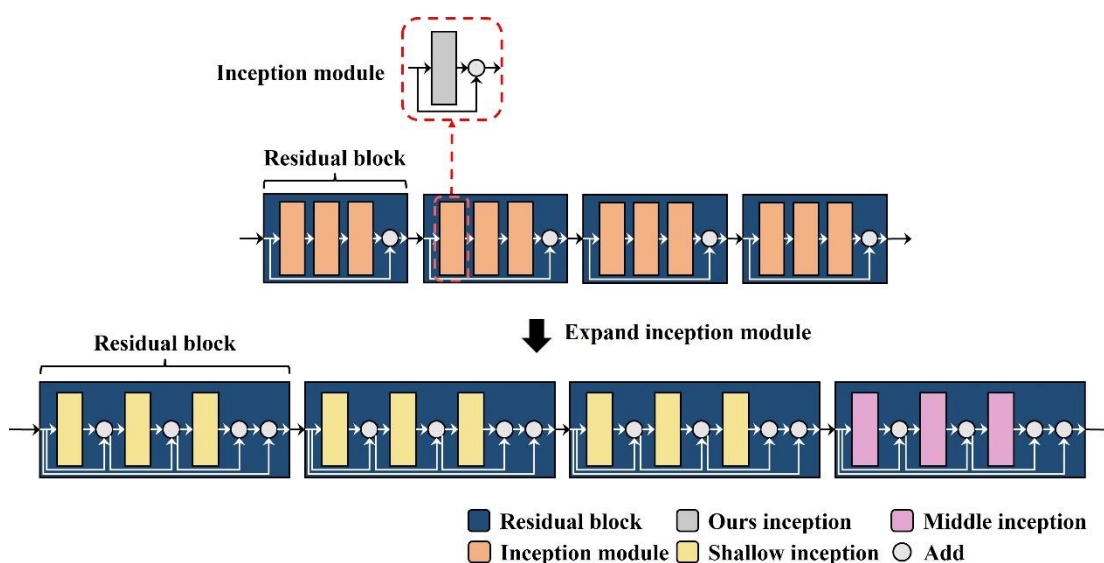


圖 3-13 : RRIM 內部結構

此外，RRIM 使用的 inception 為設計的淺層、中層和深層 inception，不過因為記憶體空間不足問題，因此 RRIM 只用淺層和中層的 inception，且占比為 3 比 1 (圖 3-13)。而深層 inception 則在 RRIM 外部，於生成器後半部的位置才

使用(圖 3-14)。

3.2.3 修改生成器整體結構

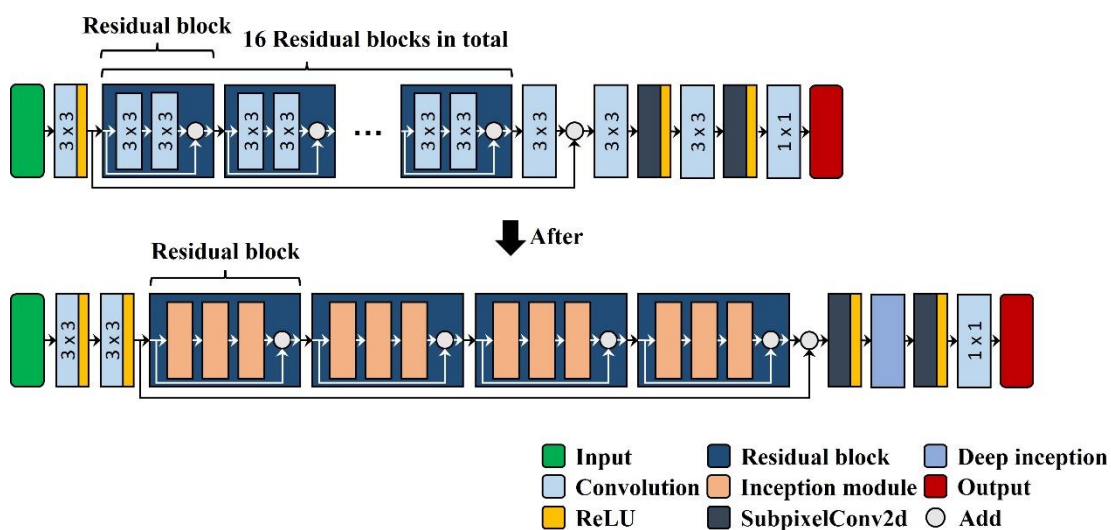


圖 3-14：修改後的生成器結構

圖 3-14 為去除 BN 層後的 SRGAN 與修改完成的生成器整體結構比較圖。我們

將 SRGAN 共 16 個的殘差塊取代成 RRIM，而由於圖片的通道數要相同才能進

行殘差，因此生成器前半部分多加一個卷積層目的是進行升維的動作。最後生

成器後半部分，考量記憶體不足的問題，將深層 inception module 置於兩個

SubpixelConv2d (Sub-pixel convolution)之間，並刪除兩個 kernel size 為 3x3 的卷

積層。

Sub-pixel convolution[\[12\]](#):

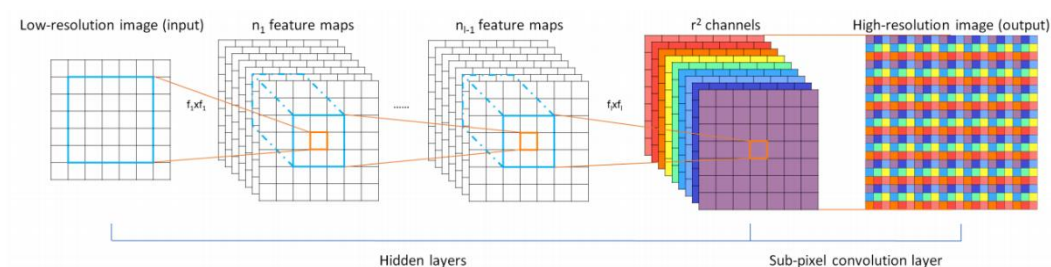


圖 3-15 : Sub-pixel convolution 示意圖[12]

SubpixelConv2d (Sub-pixel convolution)是 Tensorlayerx 函式庫中的一個函式，為一種用於將圖像或特徵圖提升解析度的方法，其運作原理先利用卷積從特徵圖中提取特徵，再對特徵進行組合與排列。

圖 3-15 中，9 張 7×7 的特徵圖經過 Sub-pixel convolution 後，可以排列組合成 21×21 的圖像，反之，若圖像要放大 3 倍，就需要先有 9 張相同大小的特徵圖。

3.2.4 判別器

我們使用原本 SRGAN 判別器結構(圖 3-16)，其架構與普遍 GAN 的判別器類似，先經過卷積萃取特徵後由全連接層匯集，最後輸出一個機率值表示輸入的高解析資料集圖片與生成圖片的差異程度；不同的是，由於提升解析度的理想是要保留全域的特徵，這裡判別器經過卷積運算後不進行池化，而是銜接 BN

層來幫助模型穩定與學習，同時也因為 BN 有加速模型收斂速度的特性，訓練

判別器的時間會較為縮減。

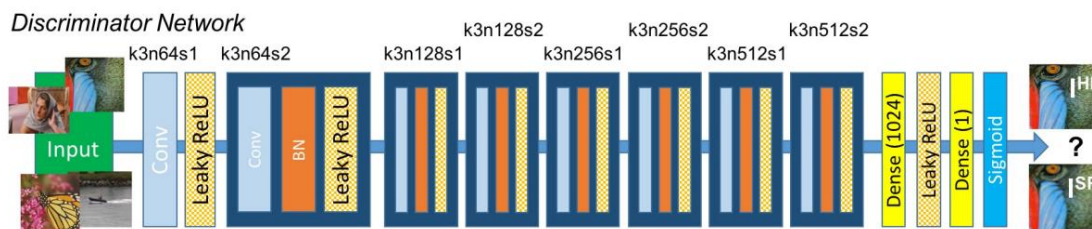


圖 3-16 : SRGAN 判別器結構[2]

2. 經過兩個卷積層將圖片維度提升，使得之後的殘差動作得以進行。
3. 進入 RRIM 結構，使特徵圖能獲取多樣性的特徵，並搭配殘差塊來增加模型訓練的穩定性。
4. 經過 3 個淺層 inception 和 1 個中層 inception 後跳出 RRIM 結構，並將 RRIM 的輸出結果與步驟 2 的輸出結果相加，使模型在深層時仍可有效的學習。
5. 利用 SubpixelConv2d 函式將特徵圖長寬放大兩倍並提升解析度(也就是進行超解析度的動作)，同時特徵圖維度也會縮小兩倍。
6. 再將特徵圖經過深層 inception，以獲得一個更多樣性的特徵圖。
7. 最後再重複一次步驟 5 後，透過一個卷積層使生成的圖片維度為 3 (也就是 Red、Green、Blue 三個顏色的通道)。

4.2 開發工具與環境

開發環境：Visual Studio Code。

模型開發框架：TensorlayerX[\[8\]](#)

TensorLayerX 是建立在 TensorFlow[\[10\]](#)之上的深度學習庫，旨在提供用於構建和訓練深度學習模型的高級 API，像是一系列預建構的層和模型，其中還包括用於數據加載、預處理、模型評估及模型部署的工具，並且適用於廣泛的應用，包括圖像和視頻識別、自然語言處理和強化學習。

網頁開發框架：Django[\[11\]](#)。

訓練資料集：DIVERse 2k (DIV2K)[\[9\]](#)。

DIV2K 是高解析度圖像數據集，通常用於機器學習研究，特別是在電腦視覺領域。該數據集由 1000+1000 張高低解析度的圖像組成，並帶有用於各種任務（如圖像分類、物件檢測和語義分割）的標籤註釋。DIV2K 由德國康斯坦茨大學的 DIVERse 團隊創建，已成為評估機器學習模型在高解析度圖像上性能的廣泛使用的基準。

硬體設備型號：MSI RTX3060Ti Gaming Z Trio 8G LHR (顯卡)，芝奇 G.SKILL 幻光戟 16G D4-3200 * 2 (記憶體)。

4.3 模型訓練

我們以 Initial epochs 為 10 次、Epochs 為 400 次、Batch_size 為 4 和 Learning

Rate 為 0.03 作為模型訓練的超參數。圖 4-2 為訓練流程，而圖 4-3 和圖 4-4 為

生成器和判別器在訓練時損失值的變化狀況。

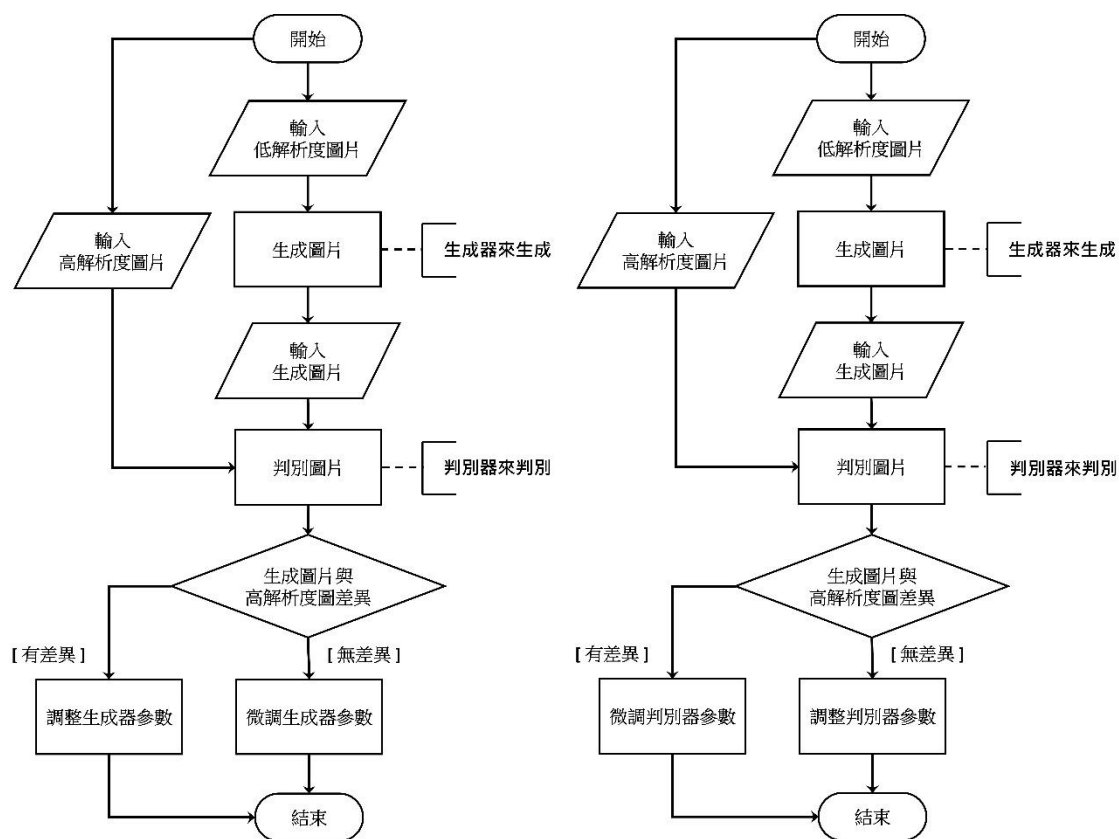


圖 4-2：生成器訓練流程圖(左)與判別器訓練流程圖(右)

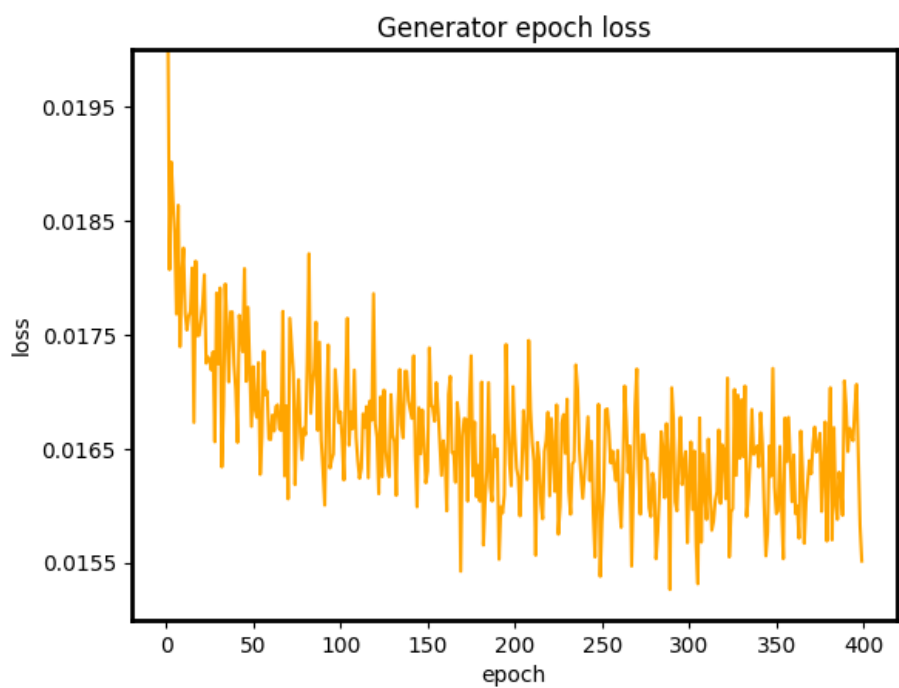


圖 4-3：生成器訓練損失值變化

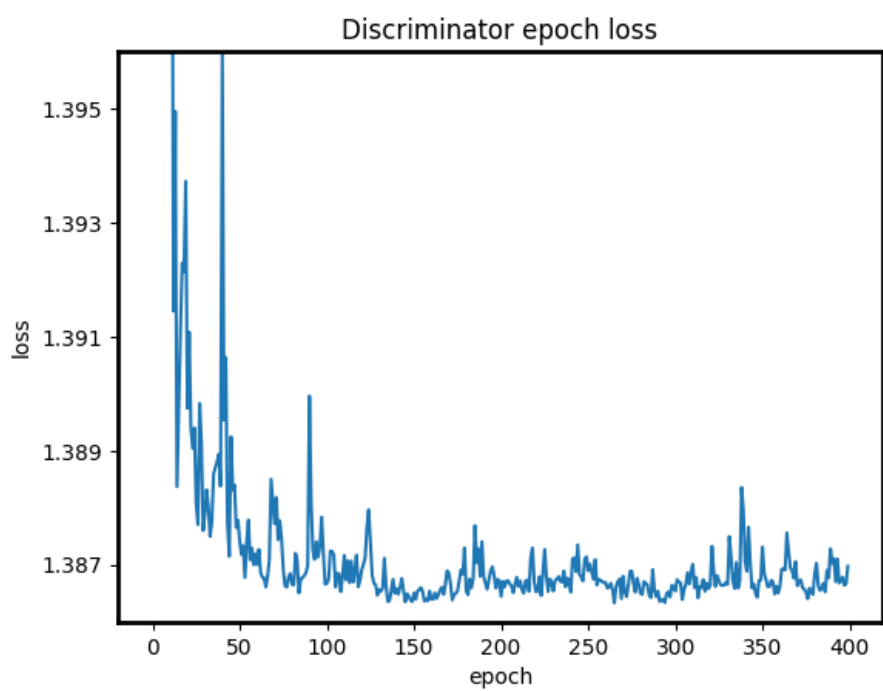


圖 4-4：判別器訓練損失值變化

4.4 相關研究之比較

4.4.1 生成圖片比較

圖 4-5、圖 4-6、圖 4-7 為 Bicubic、SRGAN 和我們模型(Ours)生成的圖片，與原

始高解析圖片的結果比較圖。其中，使用的圖片資料集來源分別為 Set5[14]、

Set14[14]、Urban100[14]，而我們模型與 SRGAN 的訓練次數皆為 400 次且

batch size 為 4。

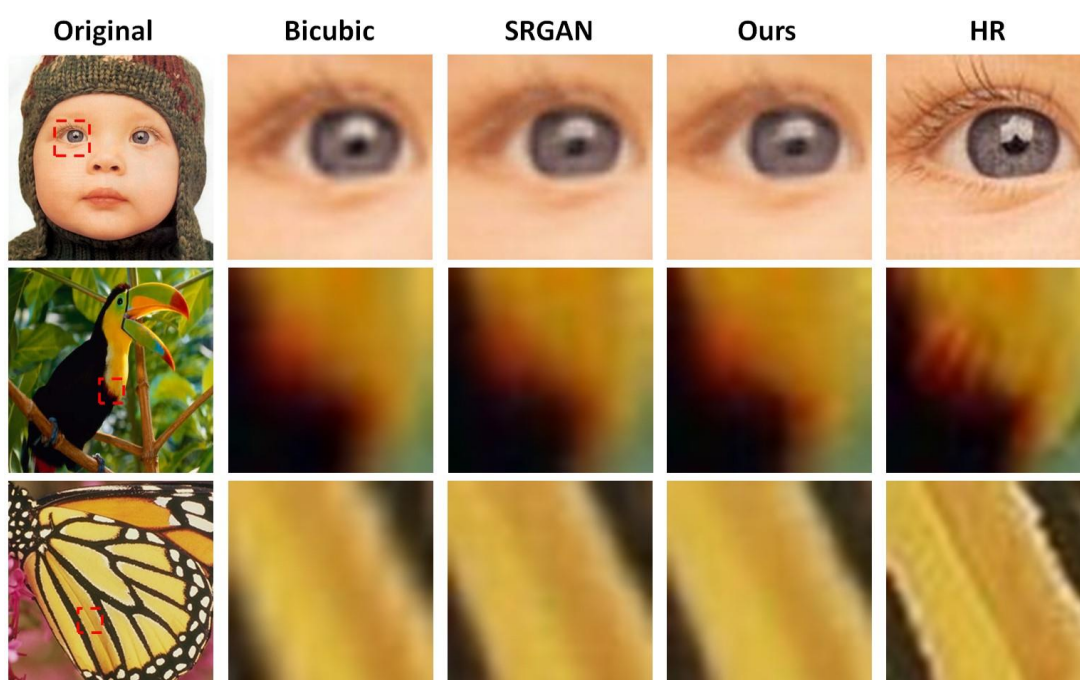


圖 4-5 : Set5 dataset 結果比較圖

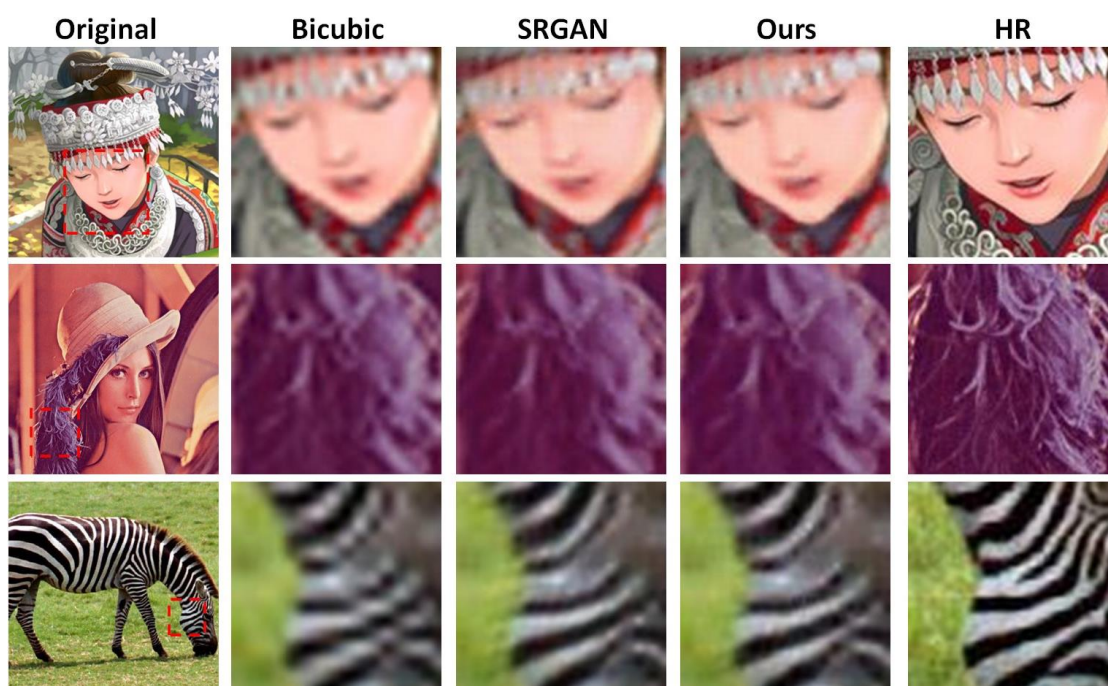


圖 4-6 : Set14 dataset 結果比較圖

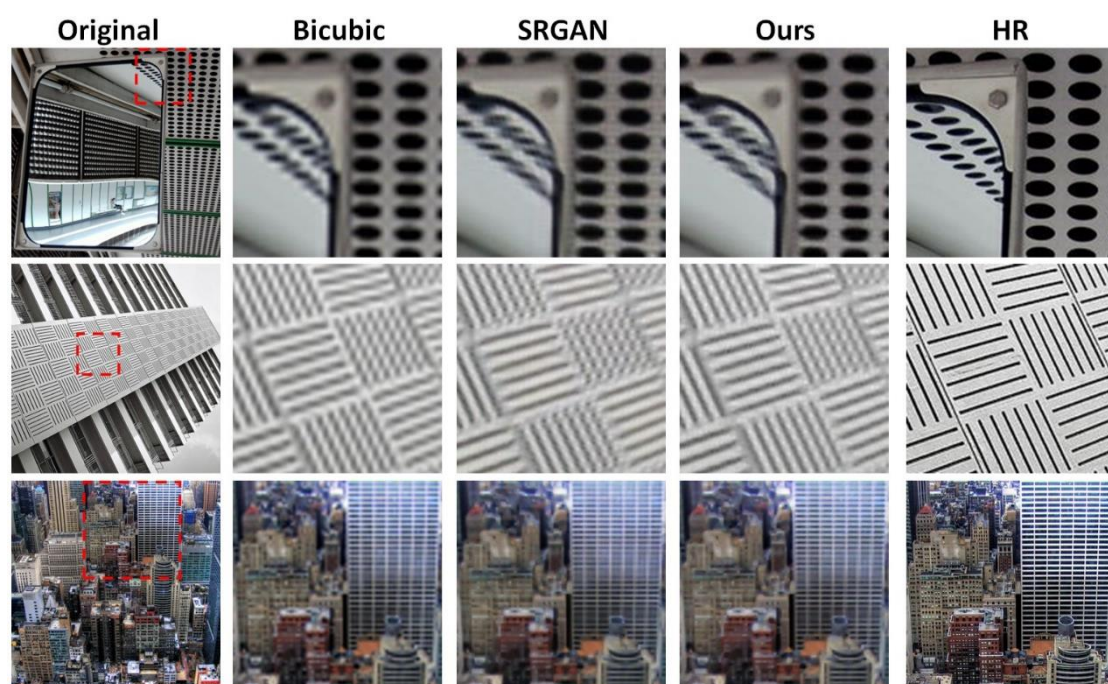


圖 4-7 : Urban100 結果比較圖

4.4.2 評估指標比較

表 4-1、表 4-2、表 4-3 為 Set5、Set14 和 Urban100 各自的資料集內所有圖片加

總平均的評估指標，其中粗體字表示評估數值最好。

表 3-1 : Set5 dataset 平均評估指標

模型 評估指標	Bicubic	SRGAN	Ours
RMSE	0.0031	0.0031	0.0030
PSNR	26.8905	26.7233	26.8981
SSIM	0.7762	0.7696	0.7768
FSIM	0.6601	0.6565	0.6609

表 3-2 : Set14 dataset 平均評估指標

模型 評估指標	Bicubic	SRGAN	Ours
RMSE	0.0040	0.0039	0.0039
PSNR	24.3709	24.3715	24.4972
SSIM	0.6687	0.6583	0.6639
FSIM	0.6039	0.6026	0.6064

表 3-3 : Urban100 dataset 平均評估指標

模型 評估指標	Bicubic	SRGAN	Ours
RMSE	0.0054	0.0053	0.0052
PSNR	21.8013	22.0325	22.0793
SSIM	0.6366	0.6396	0.6433
FSIM	0.5835	0.5913	0.5941

4.4.3 訓練速度比較

圖 4-8 為 SRGAN 與我們模型每訓練 20 次所花費的訓練時間狀況。可發現我們模型訓練時間相較穩定，且平均訓練速度較快。

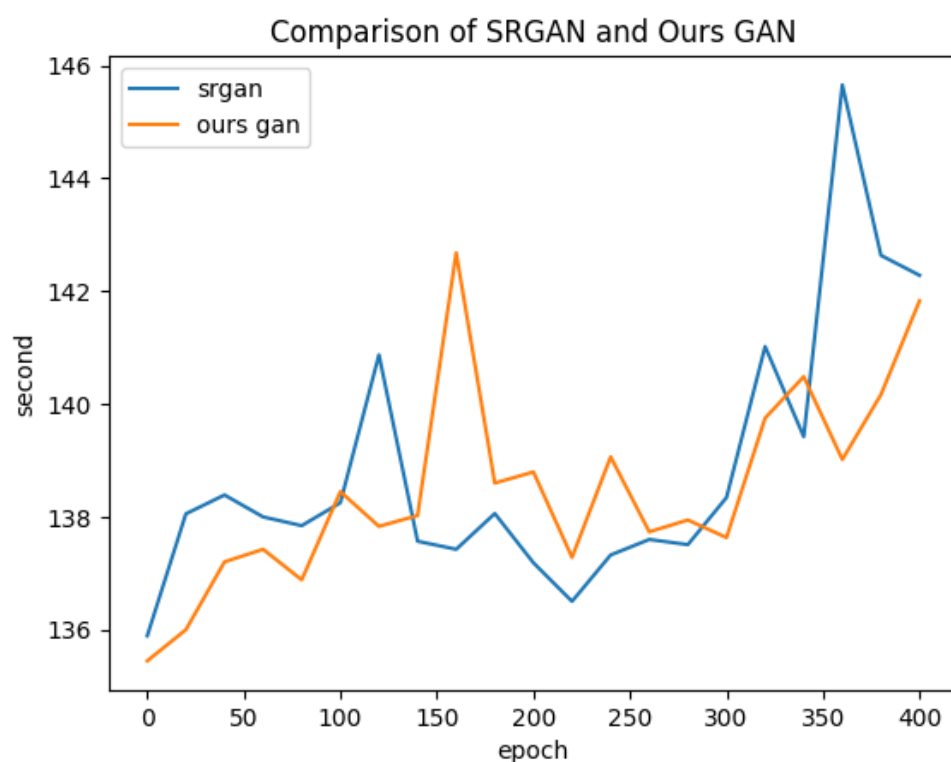


圖 4-8：模型訓練時間比較

4.5 網站

網站主要功能為上傳圖片並呈現提升解析度後的圖片結果，目的是提升專題作

品的實用度，另外還加入了專題的研究背景、目的與參考資料到網站中。

4.5.1 網站功能

為了能較好的展現和介紹我們作品，因此將網站設有以下 3 個網頁：

Home :

網站的首頁，內容包含專題研究背景和目的。

提升解析度 :

可在此網頁上傳圖片並提升解析度，網頁中含有 5 個功能：

1. 選擇圖檔：選擇所需提升解析度的圖檔。
2. 上傳圖檔：將圖檔上傳後即可提升解析度。其中必須選擇圖檔並上傳才會
有預覽圖片、重新上傳和下載圖片的功能。
3. 預覽圖片：比對利用雙三插值法(Bicubic)與我們提升解析度後的圖。
4. 重新上傳：選錯圖檔或提升另一張圖片解析度時可重新回到上傳動作。

5. 下載圖片：下載利用我們作品提升解析度圖片。

參考文獻：

包含我們模型與網頁製作所主要參考的網頁連結。

4.5.2 網站架構

網頁主要分為 3 個架構：

使用者介面：

使用者介面之功能及呈現，主要參考了具有多個開源 html、css、js 的

W3School[\[13\]](#)內提供的教學。

後端應用程式：

後端應用程式主要負責接收、回應使用者介面請求，並根據請求在正確的資料

庫位置存取資料。因為使用 Django 作為網頁開發框架，而 Django 具有開箱即

用的功能，所以只要在對應的 python 檔內寫出所需功能的程式碼即可。

在後端應用程式寫了使用者介面觸發選擇圖檔、上傳圖檔、下載圖檔、預

覽圖檔、重新上傳功能時所對應的回應。

資料庫：

儲存了上傳的圖檔和雙三插值及我們作品提升解析度的圖檔。

4.5.3 網頁與模型連結

在使用 GAN 模型時，只需利用生成器即可生成高解析度圖，因此在後端應用程式

式的呼叫我們生成器模型，並設定輸出的資料庫位置即可將網頁與模型相連。

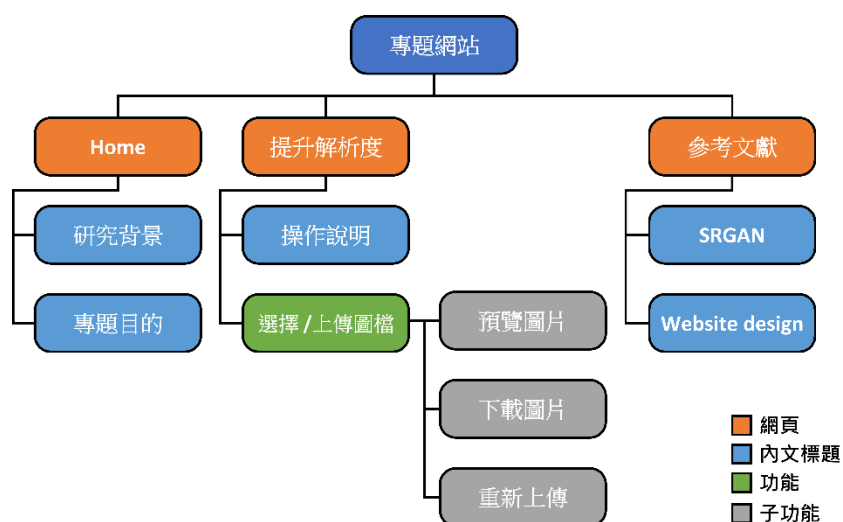


圖 4-9：網站架構圖

圖 4-9 中，子功能意指功能執行完畢後才會出現的功能。

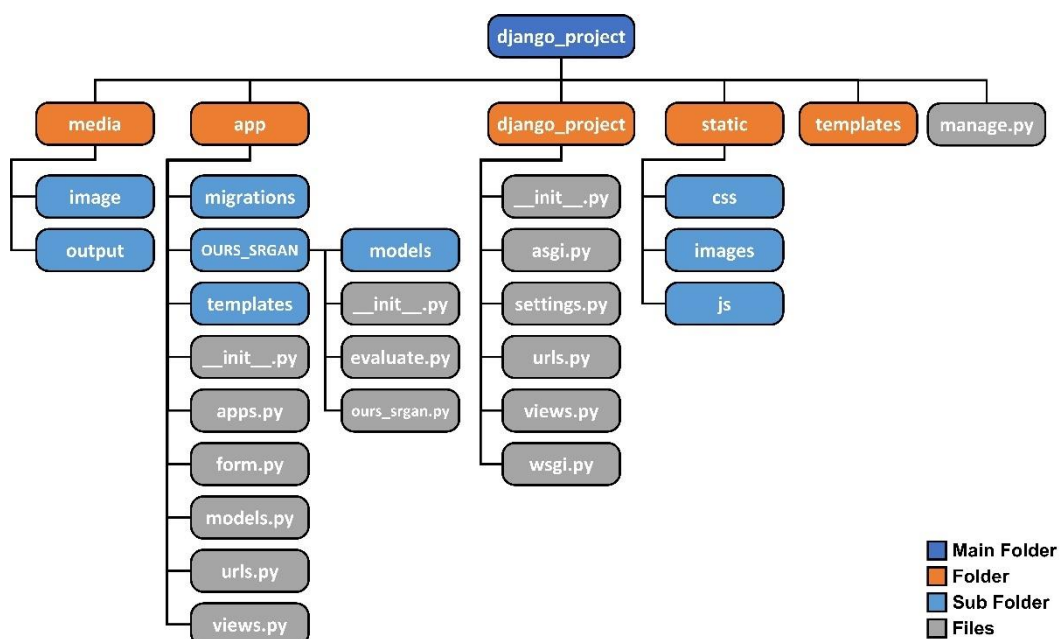


圖 4-10 : Django Folder Structure

圖 4-11 中，media 屬於資料庫，app 為提升解析度時所需的網頁及其它程式碼。Folder 類別的 django_project 儲存設定 django 的相關檔案，而 templates 為首頁、參考文獻網頁的 html 存放位置。static 為網頁使用的 css、images 和 js 的資料夾。

4.5.4 網頁結果呈現

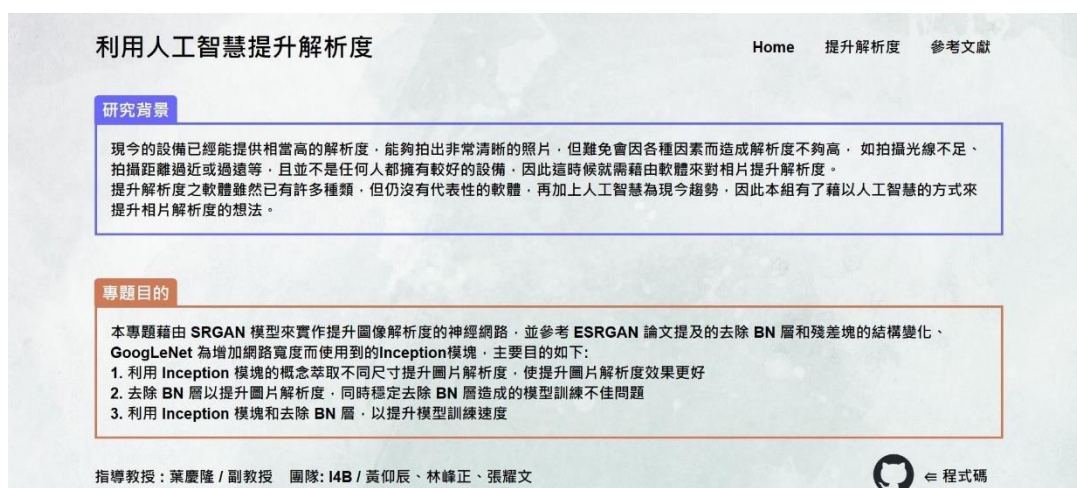


圖 4-11：網頁首頁(Home)畫面



圖 4-12：提升解析度網頁畫面

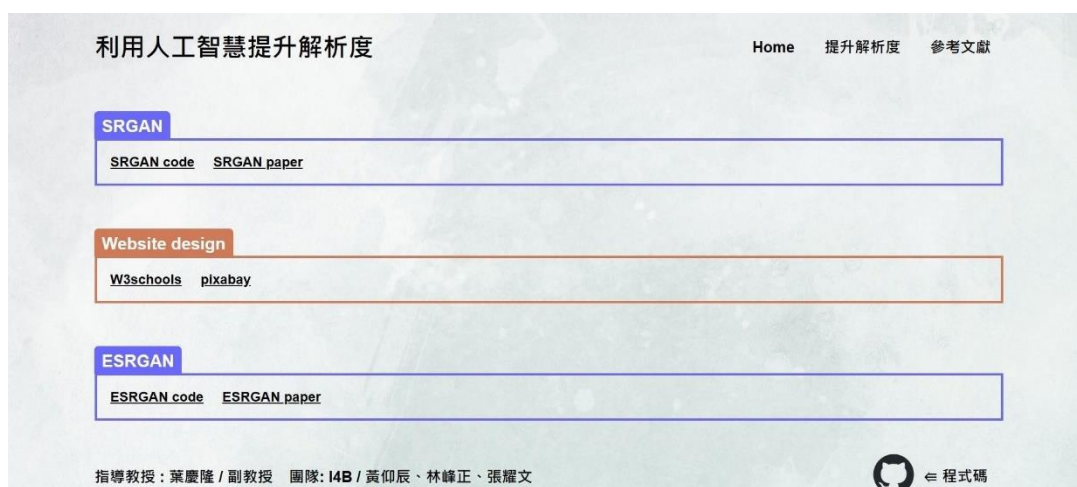


圖 4-13：參考文獻網頁畫面



圖 4-14：選擇圖檔準備上傳畫面



圖 4-15：上傳完並提升解析度後畫面

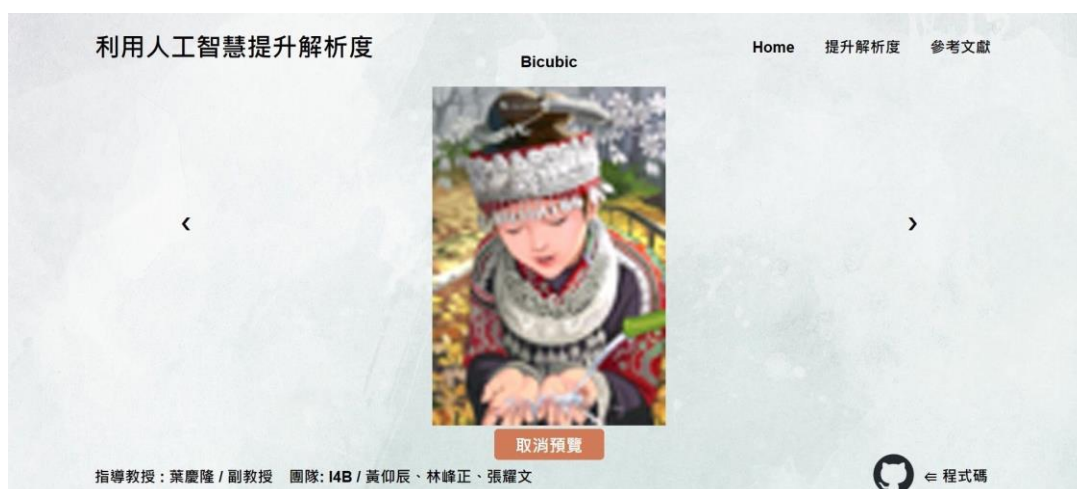


圖 4-16：預覽早期利用雙三次插值(Bicubic)提升解析度圖

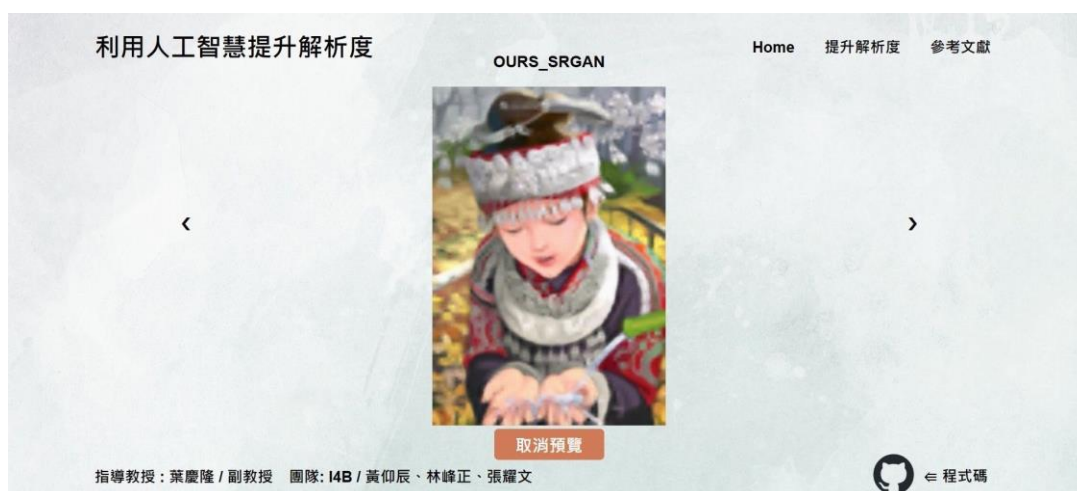


圖 4-17：預覽我們作品提升解析度圖



圖 4-18：成功下載時畫面

第 5 章 結論

我們專題研究目的是希望在於提升圖像解析度的同時，也能讓圖像的細節與真實性更好，因此以 SRGAN 為基礎進行修改。而因為 ESRGAN 論文中提出 SRGAN 的 BN 層所帶來的負面影響和殘差塊結構的設計，以及 GoogLeNet 使用的 Inception v3 的概念使得圖像萃取特徵有更好的效率和成效，所以本專題將 SRGAN、ESRGAN、Inception v3 的部分概念融合。

專題在研究的過程中，由於希望所有萃取的特徵都盡量能夠保留，所以在生成器的網路架構中沒有使用任何正規化技術，但這容易造成訓練的結果過擬合，因此在設計時除了要讓提升解析度品質較好之外，也要盡量避免網路架構過於複雜，並且因為 Inception module 架構會使網路的寬度、深度都增加，所以如何適當的使用、修改 Inception module 架構是本專題最難也是最重要的一個考量。

最終，本專題研究結果得知 Inception module 在網路淺層、中層使用時，要避免網路太寬才能有效訓練模型，而為了簡化網路架構使網路層數減少也會讓

網路訓練效果較差，因此設計生成器整體架構得出的結論是，網路需具有一定深度，且網路寬度設計隨著網路深度增加而變寬，這樣才能讓 Inception module 效果發揮最好。至於參考 ESRGAN 的部分概念，主要影響於它的殘差塊結構設計，使得我們的模型在網路深層部分，以及去除 BN 層時訓練的穩定性較好。

總結來說，Bicubic 和訓練 400 次的 SRGAN 與我們的模型，在有限硬體資源和訓練次數不多的狀況下進行相比，雖然差距不明顯，但我們模型提升解析度的效果仍較好，同時也因為去除 BN 以及減少卷積帶來的計算量，我們模型的訓練速度也優於 SRGAN，因此本專題成功地將 ESRGAN 及 Inception v3 部分概念融入 SRGAN 中，並且克服了在有限的記憶體空間下訓練模型。另外，在本專題中，我們對於 SRGAN 的改進主要著重於生成器的部分，由於判別器採用的是傳統的 CNN 架構，我們認為可以根據上面的實驗結果，將 Inception v3 的概念也使用在判別器上，並且有很大的機會能讓我們再減少模型訓練的時間，同時也因為 Inception 的架構很常使用於辨識圖像的模型上，判別器的辨識能力也可能再提升。

参考文献

- [1] Ian Goodfellow、Jean Pouget-Abadie、Mehdi Mirza、Bing Xu、David Warde-Farley、Sherjil Ozair、Aaron Courville、Yoshua Bengio, “Generative Adversarial Networks,” *International Conference on Neural Information Processing Systems (NIPS)*, 2014.
- [2] Christian Ledig、Lucas Theis、Ferenc Huszar、Jose Caballero, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” *IEEE Transactions on Computer Vision and Pattern Recognition*, 2016.
- [3] Sergey Ioffe、Christian Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *International Conference on Machine Learning (ICML)*, 2015.
- [4] Kaiming He、Xiangyu Zhang、Shaoqing Ren、Jian Sun, “Deep Residual Learning for Image Recognition,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [5] Xintao Wang、Ke Yu、Shixiang Wu、Jinjin Gu、Fangzhou Liao、Xiaoyong Shen、Chen Change Loy、Dahua Lin, “Enhanced Super-Resolution Generative Adversarial Networks,” *IEEE Transactions on Image Processing*, 2018.
- [6] Christian Szegedy、Wei Liu、Yangqing Jia、Pierre Sermanet、Scott Reed、Dragomir Anguelov、Dumitru Erhan、Vincent Vanhoucke、Andrew Rabinovich, “Going Deeper with Convolutions,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [7] Christian Szegedy、Vincent Vanhoucke、Sergey Ioffe、Jonathon Shlens、Zbigniew Wojna, “Rethinking the Inception Architecture for Computer Vision,” *IEEE Conference on Computer Vision and Pattern Recognition*

(*CVPR*), 2015.

- [8] Jiajun Qi 、 Wei Song 、 Jingyun Mao, “Welcome to TensorLayerX,” Tencent, [Online]. Available: <https://tensorlayerx.readthedocs.io/en/latest/>.
- [9] Christian Ledig 、 Lucas Theis 、 Ferenc Huszar 、 Jose Caballero, “DIV2K dataset: DIVERse 2K resolution high quality images as used for the challenges @ NTIRE (CVPR 2017 and CVPR 2018) and @ PIRM (ECCV 2018),” 2017. [Online]. Available: <https://data.vision.ee.ethz.ch/cvl/DIV2K/>.
- [10] Google Brain, “TensorFlow,” Google, [Online]. Available: <https://www.tensorflow.org/?hl=zh-tw>.
- [11] Adrian Holovaty 、 Jacob Kaplan-Moss, “Django : The web framework for perfectionists with deadlines.,” Django Software Foundation, [Online]. Available: <https://www.djangoproject.com/>.
- [12] Shixiang Wu 、 Jinjin Gu 、 Fangzhou Liao 、 Xintao Wang 、 Ke Yu 、 Chen Change Loy 、 Dahua Lin, “Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [13] “W3Schools,” Refsnes Data, [Online]. Available: <https://www.w3schools.com/>.
- [14] P. Arbeláez 、 M. Maire 、 C. Fowlkes 、 J. Malik 、 E. Ong, “Set5 、 Set14 、 Urban100,” Université de Bordeaux 、 Laboratoire de l'Université de Bordeaux.
- [15] Zhou Wang, Bovik, A.C, “Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures,” *IEEE Signal Processing Magazine*, 2009.
- [16] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, “Gradient-Based Learning Applied to Document Recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.
- [17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality

assessment: From error visibility to structural similarity,” IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004.

- [18] L. Zhang, L. Zhang, X. Mou, and D. Zhang, “FSIM: A feature similarity index for image quality assessment,” IEEE Transactions on Image Processing, vol. 20, no. 8, pp. 2378-2386, Aug. 2011.