

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

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 meeting 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.....	1
1.1 研究背景.....	1
1.2 專題目的.....	2
1.3 專題專案管理.....	2
1.4 專題成果.....	3
第 2 章 文獻探討.....	5
2.1 Generative Adversarial Network.....	5
2.2 SRGAN	5
2.2.1 BN 層	6
2.2.2 Residual Block	6
2.3 ESRGAN.....	6
2.4 GoogLeNet.....	7
2.4.1 Inception module	7
2.5 評價指標.....	8
2.5.1 RMSE.....	8
2.5.2 PSNR.....	8
2.5.3 SSIM	8
2.5.4 FSIM	9
第 3 章 神經網路架構與應用設計.....	10
3.1 Model.....	10
3.1.1 Generator.....	10

3.1.2 Discriminator	15
3.1.3 Hyperparameter.....	16
第 4 章 神經網路實作與結果.....	20
4.1 開發工具與環境.....	20
4.2 網站.....	20
4.2.1 網站功能.....	20
4.2.2 網站架構.....	21
4.2.3 網頁與模型連結.....	22
4.3 網頁結果呈現.....	24
4.4 結果比較.....	28
4.5 評估表格.....	30
4.6 訓練速度.....	31
4.7 程式碼連結.....	32
第 5 章 結論.....	33
參考文獻.....	35

表目錄

表 1-1：時間進度表	3
表 1-2：工作分配表	3
表 4-1：Set5 dataset 平均評估指標	30
表 4-2：Set14 dataset 平均評估指標	31
表 4-3：Urban100 dataset 平均評估指標	31
表 4-4：訓練速度比較表	32

圖目錄

圖 3-1 : Generator 架構圖	10
圖 3-2 : RRIM 架構圖	11
圖 3-3 : Residual block 架構圖	12
圖 3-4 : mini_inception 架構圖	12
圖 3-5 : Inception_A 架構圖	13
圖 3-6 : Sub-pixel convolution 示意圖[12].....	14
圖 3-7 : Deep_layer 架構圖	15
圖 3-8 : 判別器結構圖[2].....	16
圖 3-9 : 生成器訓練流程圖	17
圖 3-10 : 判別器訓練流程圖	18
圖 3-11 : 模型訓練流程圖	19
圖 4-1 : 網站架構圖	23
圖 4-2 : Django Folder Structure	23
圖 4-3 : 網頁首頁(Home)畫面	24
圖 4-4 : 提升解析度網頁畫面	25
圖 4-5 : 參考文獻網頁畫面	25
圖 4-6 : 選擇圖檔準備上傳畫面	26
圖 4-7 : 上傳完並提升解析度後畫面	26
圖 4-8 : 預覽早期利用雙三次插值(Bicubic)提升解析度圖	27
圖 4-9 : 預覽我們作品提升解析度圖	27
圖 4-10 : 成功下載時畫面	28
圖 4-11 : Set5 dataset 結果比較圖	29
圖 4-12 : Set14 dataset 結果比較圖	29
圖 4-13 : Urban100 結果比較圖	30

第 1 章 緒論

1.1 研究背景

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。

提升解析度之軟體雖然已有許多種類，但仍沒有代表性的軟體，再加上人工智慧為現今趨勢，因此我們有了藉以人工智慧的方式來提升相片解析度的想法。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，並非從人對相片的感受觀點來生成相片。

而在 2016 年 SRGAN 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 專題目的

本專題藉由 SRGAN 模型來實作提升圖像解析度的神經網路，並參考 ESRGAN 論文提及的去除 BN 層和殘差塊的結構變化、GoogLeNet 為增加網路寬度而使用到的 Inception module，主要目的如下：

1. 去除 BN 層讓圖片增加偽影和訓練時所需計算量的影響。
2. 利用殘差塊的結構變化，使深層網路能有效、穩定的訓練。
3. 修改 Inception module，使萃取的圖像特徵多樣性，以提升圖像細節的還原，同時降低 Inception module 架構本身對記憶體的需求。

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%	60%	30%
修改與測試模型	10%	35%	55%
模型訓練	100%	-	-
程式 Debug	90%	5%	5%
網頁製作	11%	-	89%
論文製作	-	45%	55%

1.4 專題成果

我們利用多種圖片資料集與評估指標來做測試，從得到的平均數值可以得知，

我們訓練 400 次的模型相較於 Bicubic 以及同樣訓練 400 次的 SRGAN，提升圖

像解析度的品質相對較好。

另外品質較好的同時，也成功避免去除 BN 層所帶來的負面影響，以及網

路層數較深時的訓練效果不佳、不穩定的問題，並且加入了修改後的 Inception

module，使得在有限的記憶體空間下也能順利運行且發揮功效。

而在實驗的過程中，我們使用 Colab GPU 免費版以及自身的硬體設備運

行，發現使用較高階的顯卡時我們模型訓練速度比 SRGAN 快。總結來說，我

們成功改變了 SRGAN 生成器的架構，提升圖像解析度的品質也比 SRGAN 還

要好，訓練速度也比 SRGAN 快。

第 2 章 文獻探討

2.1 Generative Adversarial Network

Generative Adversarial Network (GAN) [1]是一種用於生成新圖像、文本或其他數據的神經網絡模型。

GAN 由生成器(Generator)和判別器(Discriminator)兩個模型組成，生成器的負責生成新的數據，而判別器負責辨別真實數據與生成數據的差異。該神經網路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、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。

該模型生成器負責將低解析度圖像生成高解析度圖像，而判別器的任務則是辨別輸入圖像是否為真實的高解析度圖像，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

Batch Normalization Layer (BN 層) [3]是一種正規化方法，目的是為了解決深度學習模型中經常出現的梯度消失或爆炸的問題，同時還具有加速模型訓練的作用。

2.2.2 Residual Block

Residual block [4](殘差塊)主要用於解決深層神經網絡中梯度消失和擴展性問題。

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分辨率時使用 BN 層會影響原圖像特有的細節，甚至可能產生偽影，因此去除了殘差塊內的 BN 層，但此作法會造成模型訓練的不穩定，所以同時也對殘差塊進行了修改，而本專題參考了以上兩種做法。

2.4 GoogLeNet

GoogLeNet [6] 是一種深度 Convolutional Neural Network (CNN)。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 是一個兼具網路深度與寬度的模型。

2.4.1 Inception module

Inception module 常被用於圖像分類任務的神經網路中，主要思想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。而 Google 在之後也持續研究並改進，到目前共有 5 種版本: Inception v1~v4、Inception-Resnet，其中本專題參考了 Inception v3 [7] 減少卷積計算的方法，並使

用於模型中的所有 Inception module 架構中。

2.5 評價指標

2.5.1 RMSE

Root Mean Squared Error (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。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)，通常用來衡量圖像品質或圖像重建誤差的大小。PSNR 的值越大，表示圖像品質越好，或者圖像重建誤差越小。

2.5.3 SSIM

Structural Similarity Index (SSIM Index) 用來衡量圖像品質或圖像重建誤差的大小，常用於圖像壓縮、圖像放大、圖像去雜訊等問題中。

SSIM 衡量圖像間的亮度、對比度和結構相似性。SSIM 的值在 0 到 1 之間，越接近 1 表示圖像品質越好，或者圖像重建誤差越小。

2.5.4 FSIM

Feature Similarity Index Measure (FSIM)用來衡量圖像視覺質量，常用於圖像壓

縮、圖像放大、圖像去噪等問題中。FSIM 的值在 0 到 1 之間，越接近 1 表示圖

像質量越好。

第 3 章 神經網路架構與應用設計

3.1 Model

3.1.1 Generator

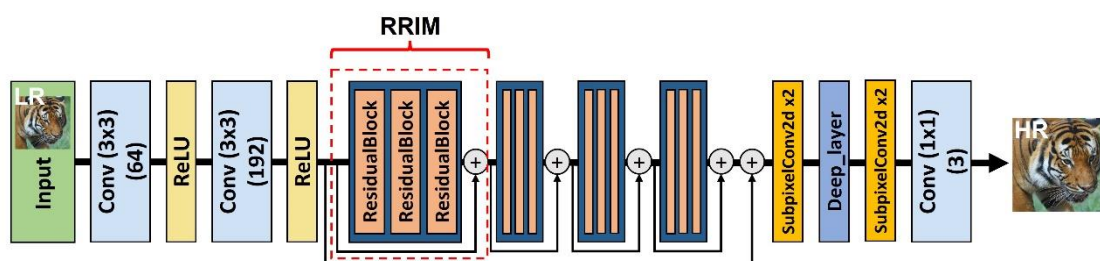


圖 3-1 : Generator 架構圖

在圖 3-1 中，輸入為高解析圖前處理成縮小 4 倍所產生的低解析度圖，此部分為 SRGAN 所使用的方法，而由於網路總層數較多因此多次使用殘差的概念，其中 \oplus 則代表殘差。

使用殘差的條件要圖片的長、寬、通道數相同，因此在前兩層進行提升通道數的動作，使得後續殘差得以運行，而後在 Residual-in-Residual Inception Module (RRIM) 架構中多次使用殘差，最後透過 SubpixelConv2d 將圖片放大 2 倍，其中在兩個 SubpixelConv2d 插入 Deep_layer，目的是對放大 2 倍的圖再進一步的獲取圖片特徵，最後一個卷積則是進行減少通道數的動作，使其變成

Red、Green、Blue 此 3 個通道數。

Residual-in-Residual Inception Module (RRIM) :

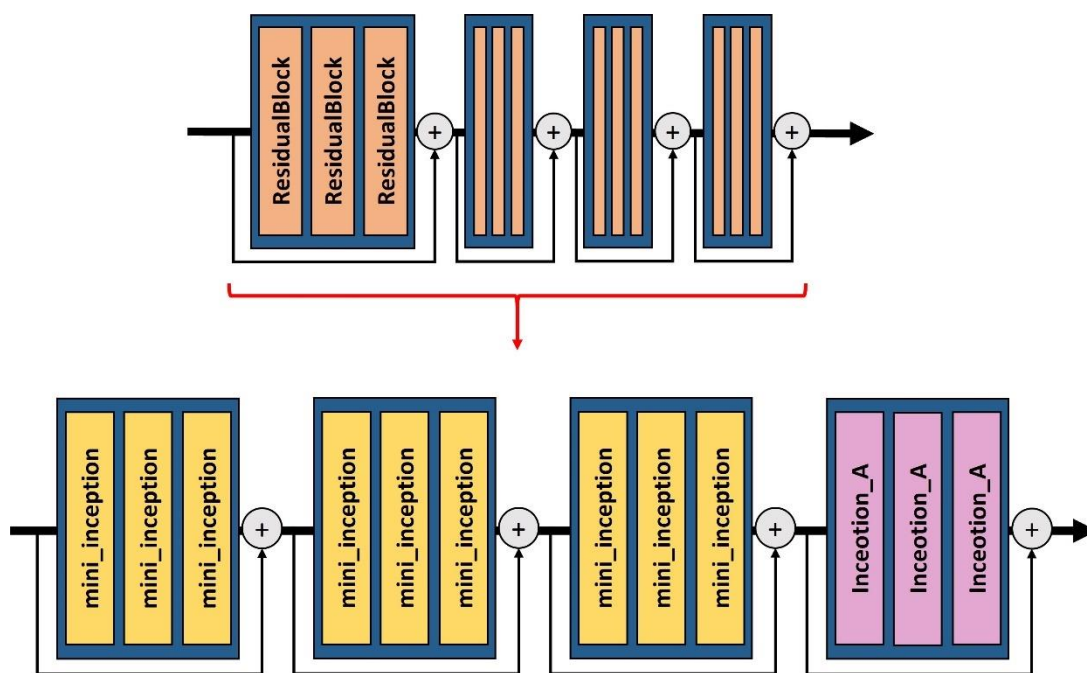


圖 3-2 : RRIM 架構圖

RRIM 架構是參考 ESRGAN 的 Residual-in-Residual Dense Block (RRDB)，我們

將 Dense block 取代成 Inception module，且使用兩種不同類型的 Inception 架

構。

其中 mini_inception 架構又比 Inception_A 架構簡單且使用次數較多，其原

因是在進行多次實驗時，發現若在網路的淺層架構過於複雜時反而會無法有效

獲取正確的特徵；而這兩個 Inception 其實還有多一層殘差(圖 3-3)，因此才叫

做 Residual-in-Residual，其作法和原因是參考 ESRGAN 論文提及避免去除 BN

層所帶來的影響。

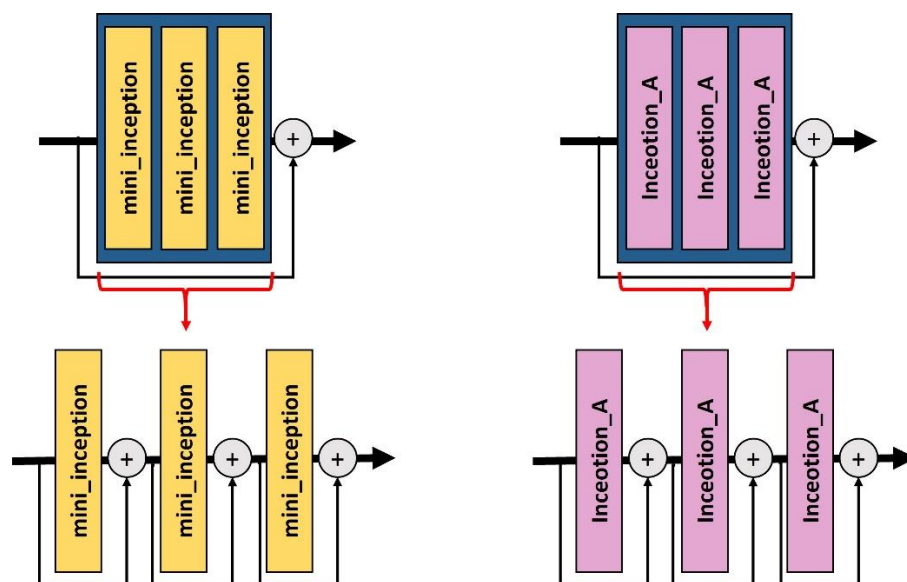


圖 3-3 : Residual block 架構圖

mini_inception :

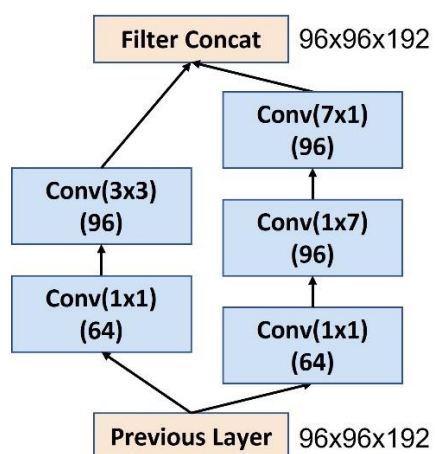


圖 3-4 : mini_inception 架構圖

mini_inception 為結構較簡單的 Inception module，圖 3-4 中，Previous Layer、

Filter Concat 旁的數字是輸出的張量大小(長 x 寬 x 通道數)，而每個卷積下的括號則是 outchannel。

其中，Conv(1x1)減少通道數目的是減少所需計算複雜度，Conv(1x7)、Conv(7x1)的目的是降低計算複雜度的同時，也保有 Conv(7x7)特徵萃取的效果，最後再進行 concatenation (concat)，使得同個輸入但不同尺寸的特徵圖加以串接，以輸出多樣性的特徵圖，此作法和原因是參考 Inception v3。

Inception_A :

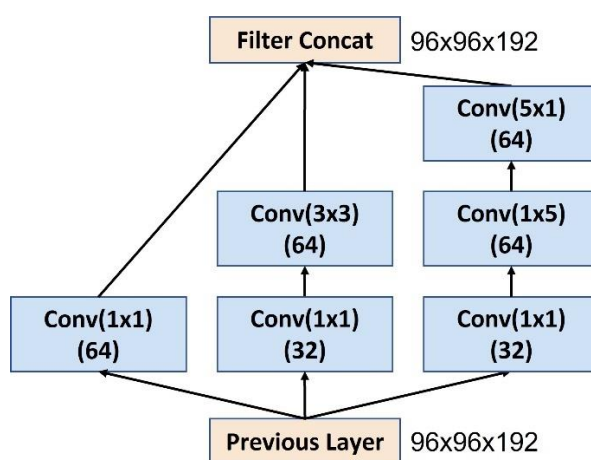


圖 3-5 : Inception_A 架構圖

Inception_A 架構參考概念也是 Inception v3，但 Inception v3 中的架構會多加一個經過 pooling 的卷積進行串接，沒有採納此作法原因是希望在有限的記憶體下得以運行，而多加一個卷積串接記憶體更容易 out of memory 外，另一原因是

pooling 會損失許多的特徵值，我們作品是提升圖片解析度，因此要盡可能的保

留所有特徵值，pooling 的作法與我們目的相反。

Sub-pixel convolution[12]:

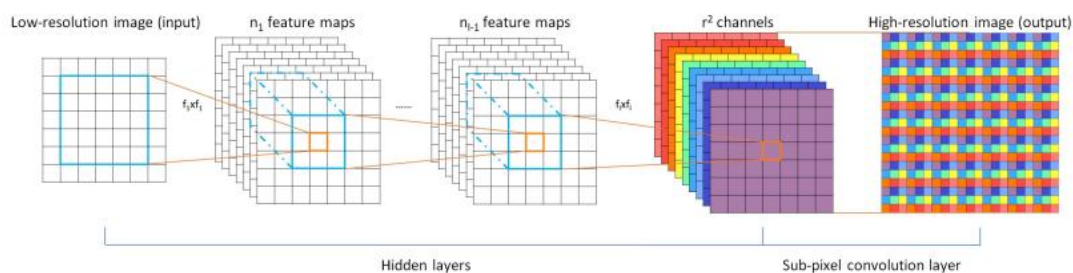


圖 3-6 : Sub-pixel convolution 示意圖[12]

Sub-pixel convolution(又簡稱 pixel shuffle)，是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圖像或特徵圖中提取特徵，再對特徵/像素點進行組合與排列。

如圖 3-6 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。

Deep_layer :

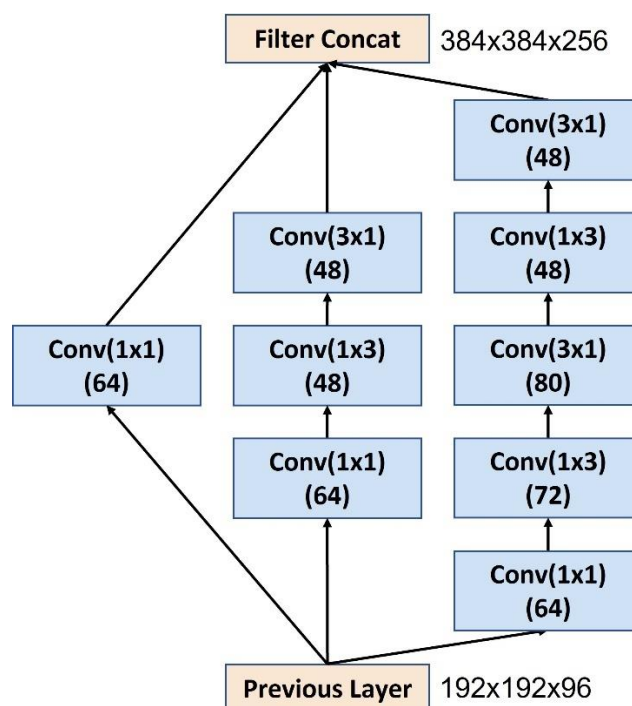


圖 3-7 : Deep_layer 架構圖

Deep_layer 的作法和原因與 Inception_A 大致相同，不同處在於，因為較深層的

網路所學到的特徵較多，因此較適合 kernel size 較小的卷積來萃取細緻的特

徵，所以進行 concat 時，Conv(1x1)的 outchannel 數量較多是這個原因；也因

為深層適合萃取細緻特徵，因此本作品的 3 種不同類型的 Inception 使用的

kernel size 隨著網路越深 kernel size 越小。

3.1.2 Discriminator

我們使用原 SRGAN 的架構，任務是判別輸入為真高解析圖還是假高解析圖，

其架構與普通 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

後輸出機率值表示真或假；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，

同時也因為加速了模型收斂的速度，訓練判別器的時間會較為縮減。

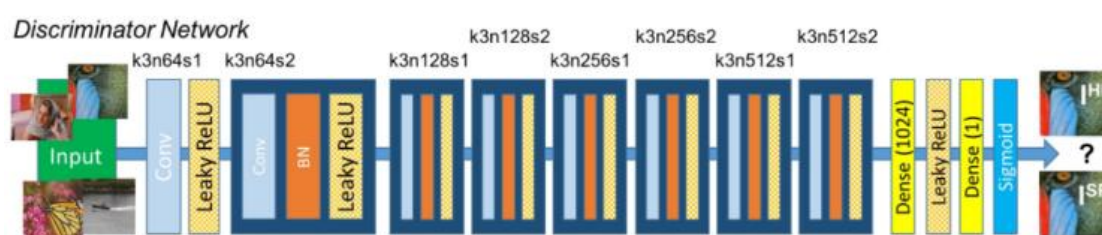


圖 3-8：判別器結構圖[2]

3.1.3 Hyperparameter

Initial epochs : 10 次 / Epochs : 400 次 / Batch_size : 4 / Learning Rate : 0.03 。

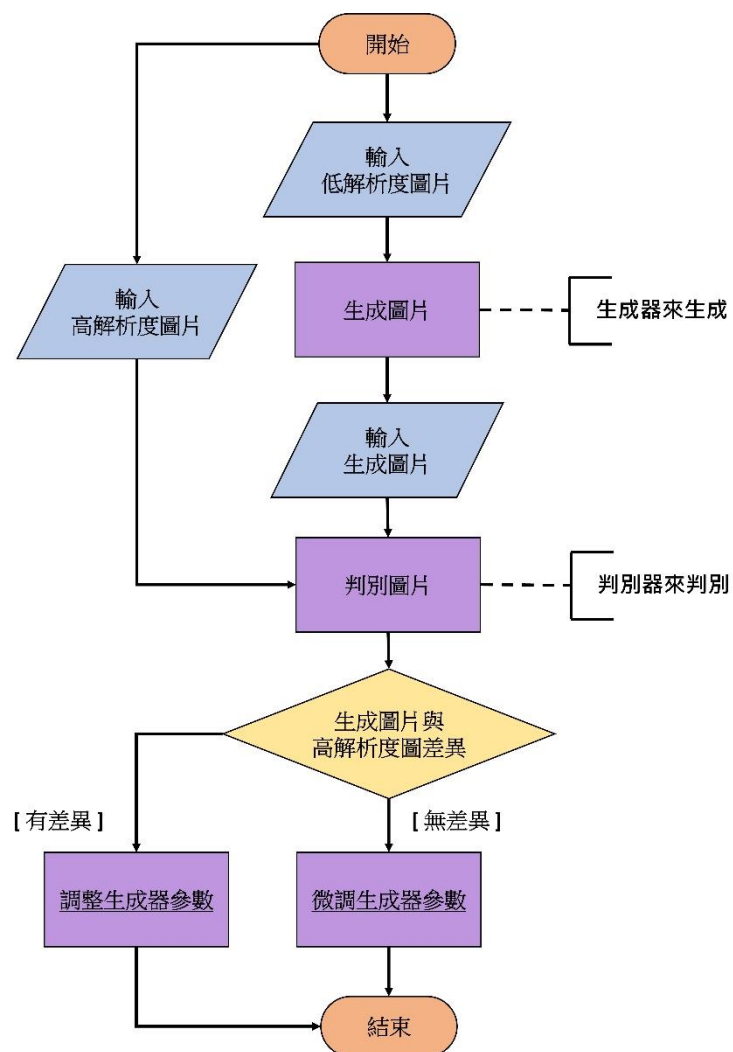


圖 3-9：生成器訓練流程圖

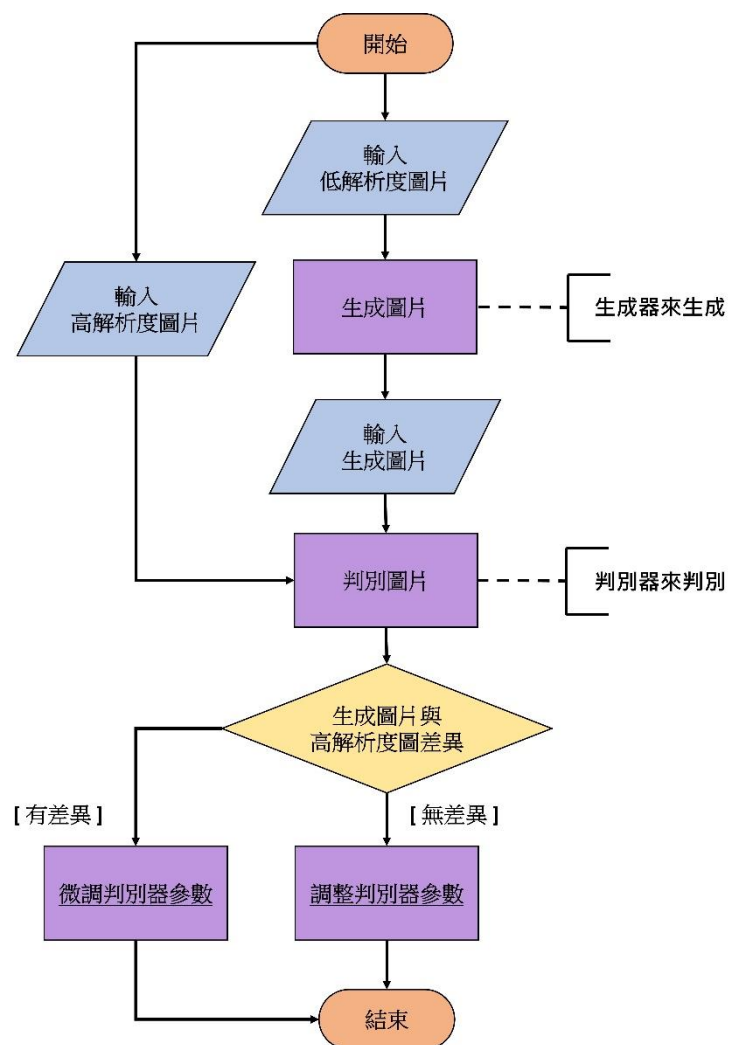


圖 3-10：判別器訓練流程圖

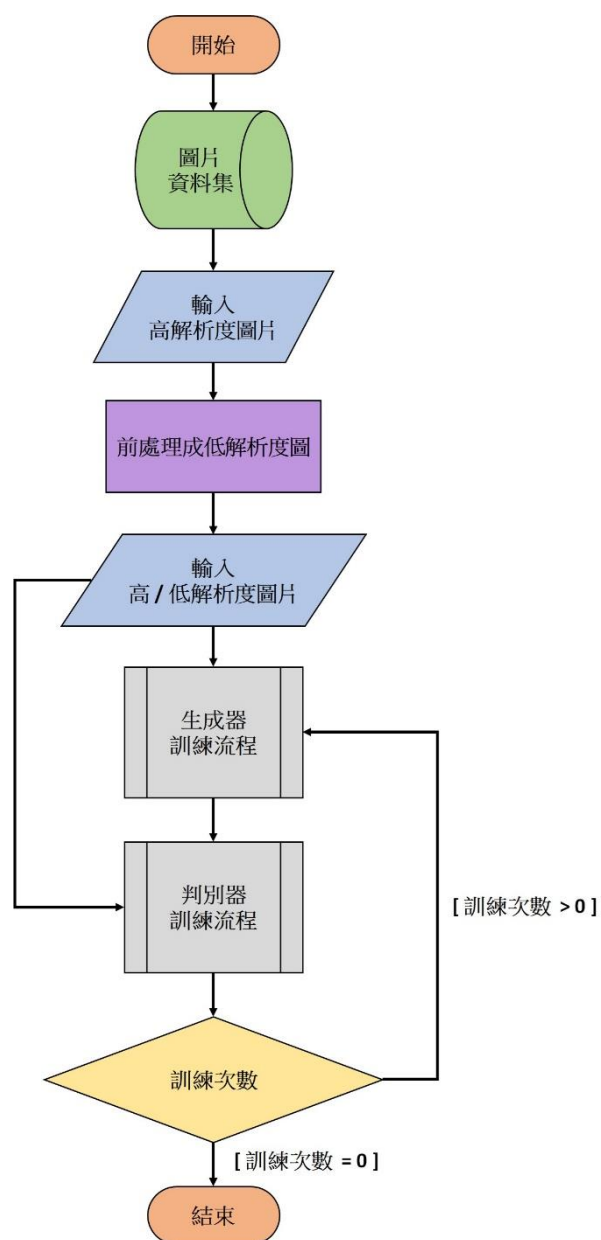


圖 3-11：模型訓練流程圖

圖 3-8 和圖 3-9 中，由於生成器和判別器的目的相反，因此調整、微調參數的條件也相反。

第 4 章 神經網路實作與結果

4.1 開發工具與環境

編輯器：Visual Studio Code。

模型開發框架：TensorlayerX[8] / **網頁開發框架**：Django[11]。

訓練資料集：DIVerse 2k (DIV2K)[9]。

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)

4.2 網站

網站主要目的要能上傳圖片並能呈現結果，以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。

4.2.1 網站功能

為了能較好的展現和介紹我們作品，因此附上專題程式碼的 Github 連結並有以

下 3 個網頁：

Home：

網站的首頁，其中介紹了專題研究背景和目的。

提升解析度：

為提升圖片解析度的網頁，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：

附上了我們模型與網頁製作所主要參考的網頁連結。

4.2.2 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：

後端應用程式為使用者介面與資料庫間都溝通橋樑，因此這部分程式碼主要負

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。

因為使用了 Django 作為網頁開發框架，而 Django 具有開箱即用的功能，因此

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。

4.2.3 網頁與模型連結

由於使用 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

程式的部分呼叫我們模型，並設定輸出的資料庫位置即可將網頁與模型相連。

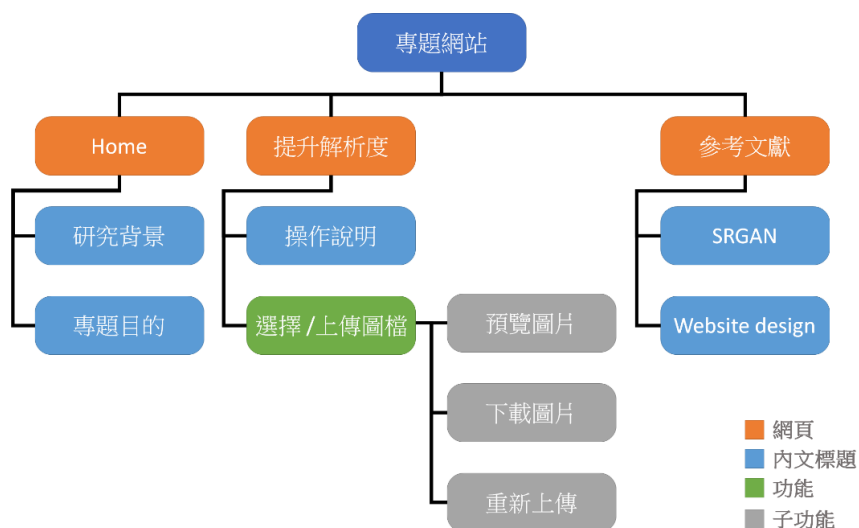


圖 4-1：網站架構圖

圖 4-1 中，子功能意指功能執行完畢後才會出現的功能。

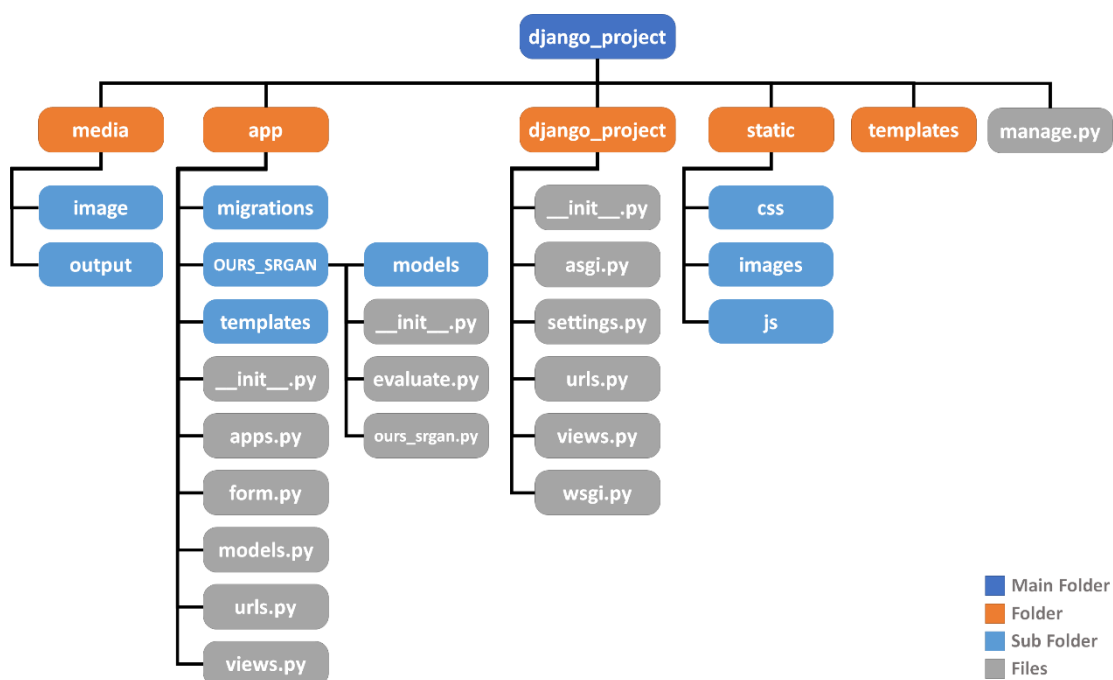


圖 4-2 : Django Folder Structure

圖 4-2 中，media 為資料庫；app 為提升解析度時所在的網頁以及後端應用

程式的資料夾(其中，Sub Folder 類別的 SRGAN 為我們作品的相關程式碼資料夾)；Folder 類別的 django_project 為創建 django 後自動生成的資料夾，屬於 django 的核心，儲存了設定 django 的相關檔案；static 為儲存使用者介面所使用的 css、images 和 js 的資料夾；Folder 類別的 templates 為首頁、參考文獻網頁的 html 存放位置。

4.3 網頁結果呈現



圖 4-3：網頁首頁(Home)畫面



圖 4-4：提升解析度網頁畫面



圖 4-5：參考文獻網頁畫面



圖 4-6：選擇圖檔準備上傳畫面



圖 4-7：上傳完並提升解析度後畫面

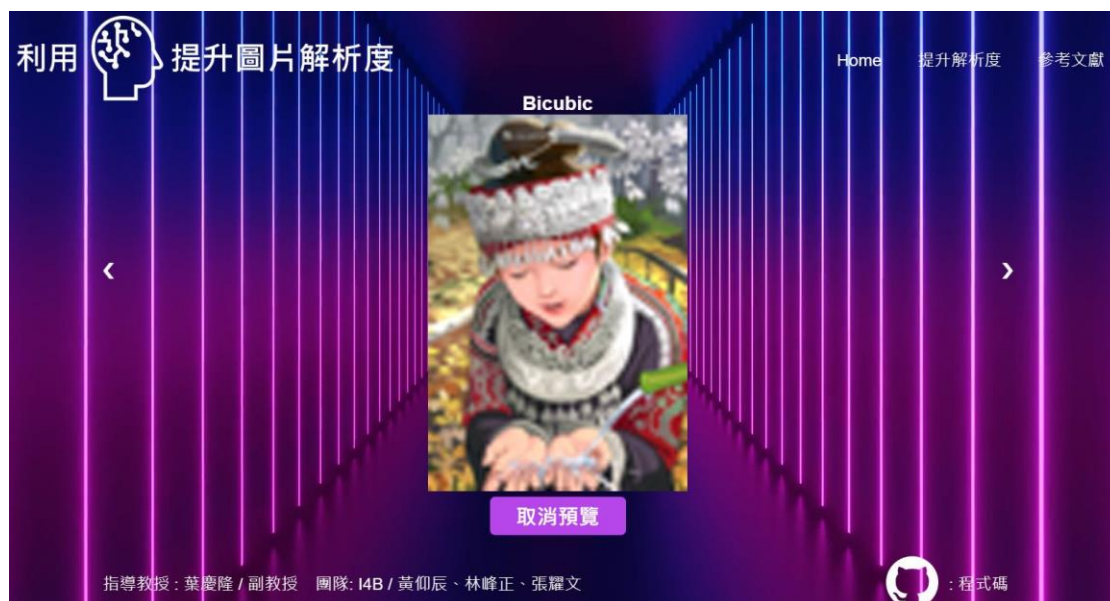


圖 4-8：預覽早期利用雙三次插值(Bicubic)提升解析度圖



圖 4-9：預覽我們作品提升解析度圖



圖 4-10：成功下載時畫面

4.4 結果比較

圖 4-11、圖 4-12、圖 4-13 為訓練 400 次的 Bicubic、SRGAN、我們模型(Ours)

和原始的高解析圖的結果比較圖。其中圖片資料集分別為 Set5[14]、Set14[14]、

Urban100[14]。

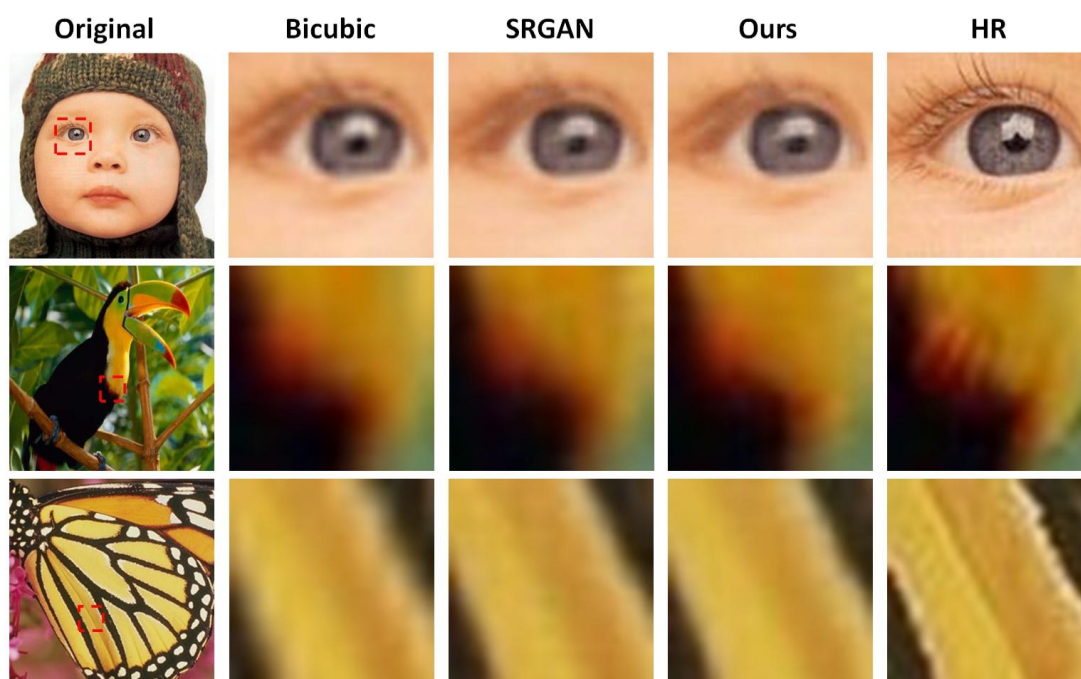


圖 4-11 : Set5 dataset 結果比較圖

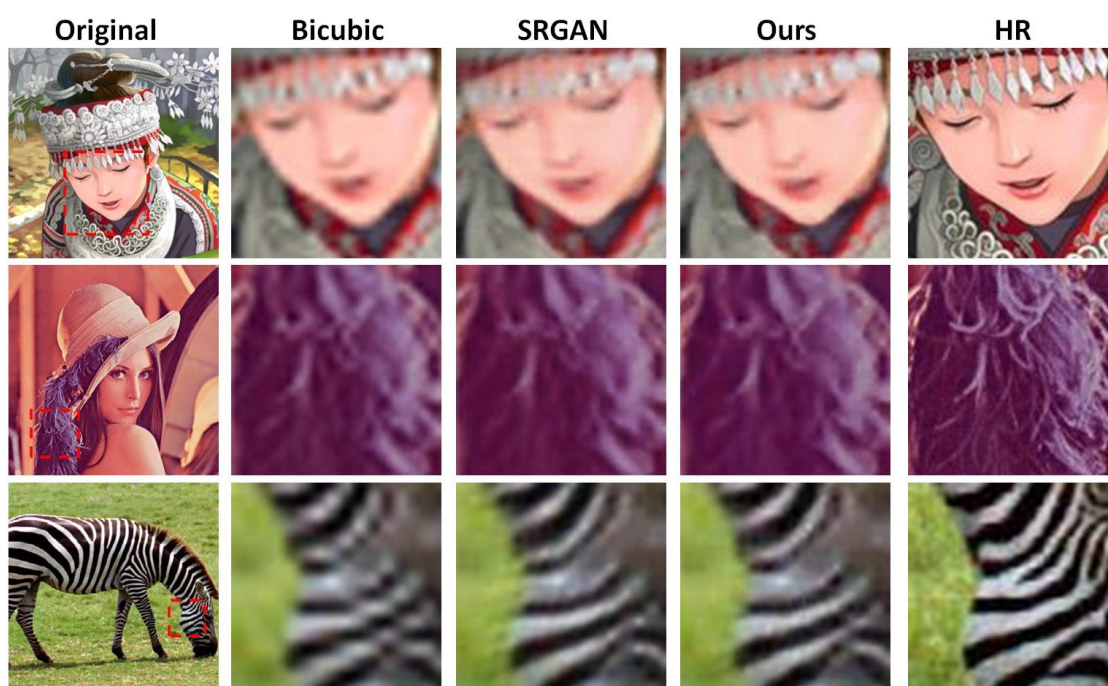


圖 4-12 : Set14 dataset 結果比較圖

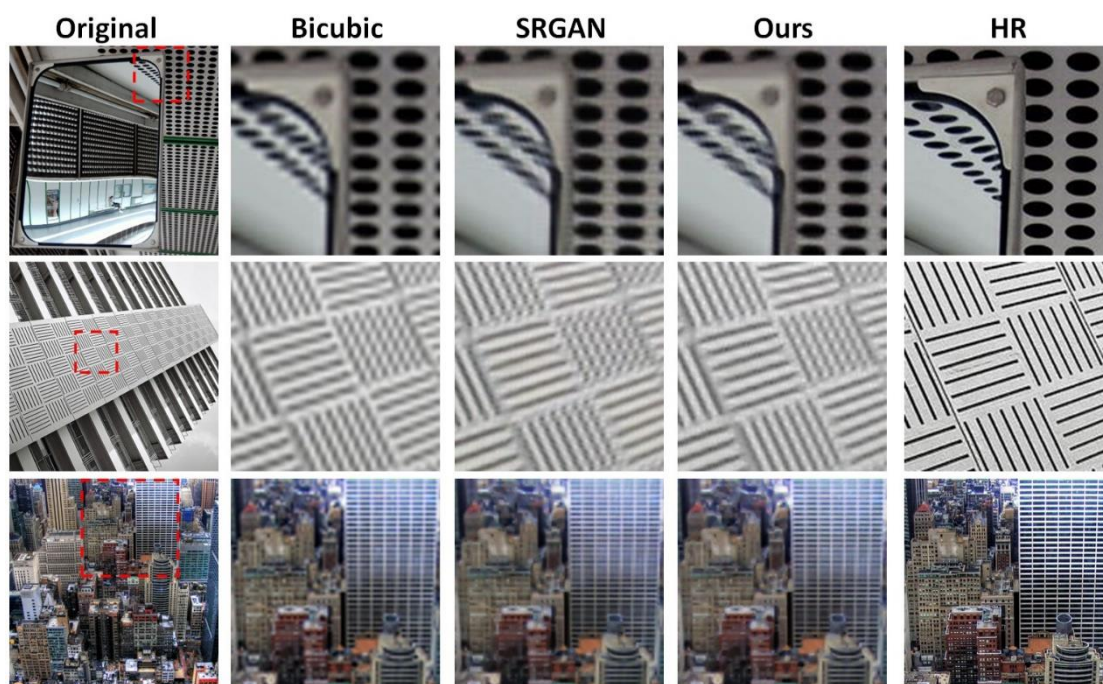


圖 4-13 : Urban100 結果比較圖

4.5 評估表格

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 4-1 : 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 4-2 : 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 4-3 : 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

4.6 訓練速度

由表 4-4 可得知，我們模型在 epoch 0、100、200、300 及 399 時，平均訓練時間是優於 SRGAN 的。

另外在紀錄訓練時間的部分，我們只有完整紀錄自身模型的時間，而 SRGAN 只有紀錄 epoch 0、100、200、300 及 399 的時間，但由於我們模型在訓練 400 次下所花費平均時間，每個 epoch 為 151s 與表 4-4 中 Average 的 152s 是差不多的，因此推估訓練 SRGAN 的總花費時間比我們模型長。

表 4-4：訓練速度比較表

模型 訓練次數	SRGAN	Ours
Epoch 0 ~ 1	about 156s	about 144s
Epoch 100 ~ 101	about 144s	about 163s
Epoch 200 ~ 201	about 156s	about 151s
Epoch 300 ~ 301	about 152s	about 147s
Epoch 399 ~ 400	about 172s	about 156s
Average	156s	152s

4.7 程式碼連結

[Ours Independent-study/ours_srgan_code at main · ZhangYaowen-](#)

[0107/Ours_Independent-study \(github.com\)](#)

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了 Inception module 在網路淺層、中層使用時，要避免網路太寬才能有效的訓練模型，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，網路需具有一定的深度，且網路寬度的設計隨著網路深度增加而變寬，這樣才能讓 Inception module 的效果發揮最好。至於參考 ESRGAN 的部分概念，主要影響在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，利用 Bicubic 和訓練 400 次的 SRGAN 與我們的模型相比，我們的模型提升解析度的效果較好，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。

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, [線上]. 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. [線上]. Available: <https://data.vision.ee.ethz.ch/cvl/DIV2K/>.
- [10] Google Brain, “TensorFlow,” Google, [線上]. 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, [線上]. 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, [線上]. 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.