

2019 年臺灣國際科學展覽會 優勝作品專輯

作品編號 190016

參展科別 電腦科學與資訊工程

作品名稱 運用 GAN 實現字體風格轉換

得獎獎項 大會獎：三等獎

出國正選代表

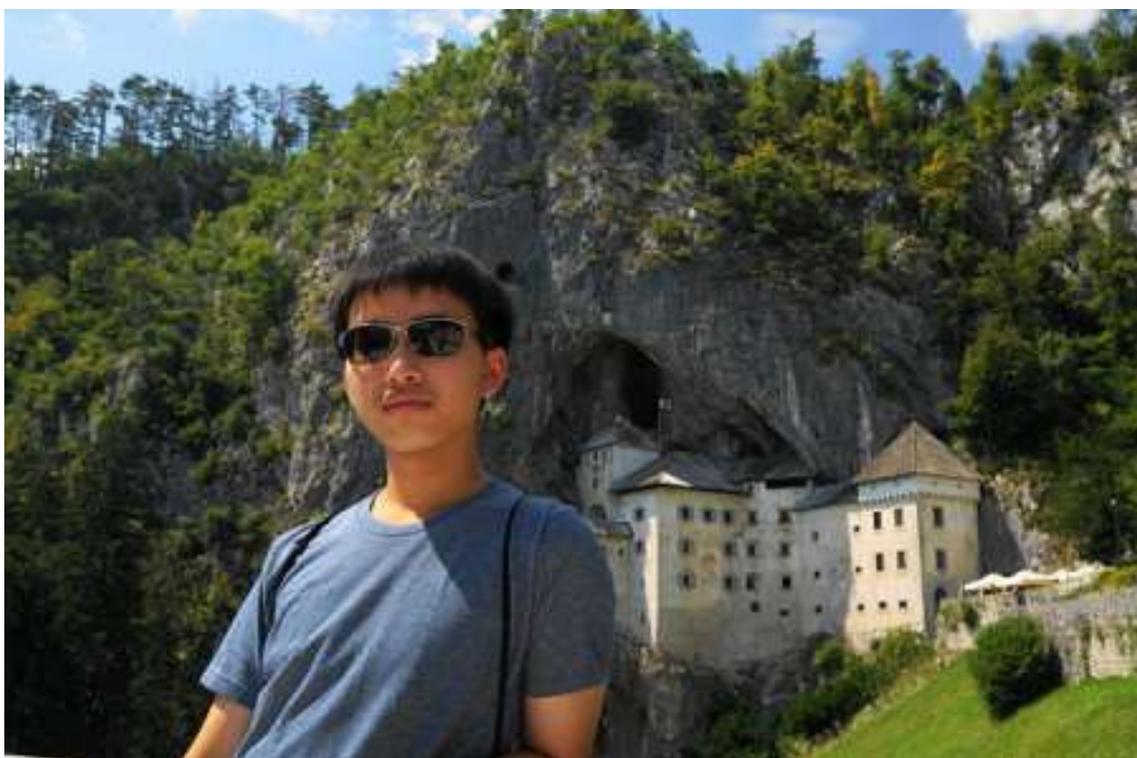
就讀學校 臺北市立建國高級中學

指導教師 許雅淳

作者姓名 邱泓翔

關鍵詞 GAN、pix2pix、CycleGAN

作者簡介



我是邱泓翔，喜歡做科學研究，偶然在看論文的過程中發現 GAN 的存在，覺得十分有趣，便將之用在字體風格轉換這個主題上。

深度學習的實驗是很辛苦的，每跑一次都需要花一天以上的時間，而在跑實驗之前的 debug 和一些繁瑣小事也總會讓我心煩不已，但正是這段辛苦的過程讓我從中認識到許多深度學習的知識，也學會了相關程式的實作細節，最終有了現在的研究成果。

摘要

本研究以實作字體風格轉換的生成對抗網路模型為動機，將 Conditional GAN 當作模型的基礎，探討 pix2pix 模型及其他研究的一些方法對模型會產生甚麼影響，以得出能最優化預測成效的深度學習模型。

首先進行的是前處理的步驟，將字體的 truetype 檔案轉換成模型輸入的 jpeg 檔，再以生成器 (Generator) 和判別器 (Discriminator) 建立 Conditional GAN 的基礎模型，然後探討加入 U-Net、Category Embedding 等方法，以及訓練資料集大小對模型造成的影響，最後實作整合的 pix2pix 模型和 CycleGAN 模型進行比較。

經過實驗後發現，U-Net 和 Category Embedding 都對模型的預測成果有所幫助。另外，對相似的字體而言，pix2pix 的效果較好，而對兩種風格差異較大的字體則需用 Category Embedding 的方式，融入更多字體進行訓練以達到更好的成效。

Abstract

This research intends to build a deep learning model for font style transfer. We use Conditional GAN as a primitive model and apply the pix2pix model and other methods to determine their effects for the sake of optimizing the output of our model.

The first step is to transfer the truetype file into jpeg to input into the model. Then we build the basic Conditional GAN model using generator and discriminator and apply several methods like U-Net and Category Embedding to determine their effects. Lastly, we compare the results of integral pix2pix model and CycleGAN model.

From the results, we find that U-Net and Category Embedding can both help the model to make better output. Besides, the pix2pix model does better for similar font styles. As for two disparate font styles, a better way is to use apply Category Embedding and more fonts to the model to gain better outputs.

一、 前言

(一) 研究動機

字體是每個人在文書處理上都會用到的工具，舉凡平常公文用的新細明體或標楷體，抑或簡報常使用的微軟正黑體，甚至是一些具有藝術性的字體，如行書體、碑體等等，都是我們在使版面整齊漂亮時經常使用的工具。

但我們所不知道的是，創造中文字體其實是一項困難且曠日費時的工作，根據統計，要創造一套符合標準（如：GBK）的中文字體，設計師要為超過 26000 個漢字作手工設計，通常需要好幾年的時間來完成。因此，若能使用深度學習的方式，先手工設計一部分的字，剩下的用深度學習的模型去作轉換，如此一來就能大幅減少創造字體所需的時間。因此，本研究的目的就是在建立一個字體轉換的深度學習模型（運用生成對抗網路 GAN 的方式實作），並對一些方法進行實驗比較及深入探討，期望能使字體風格轉換模型的效果達到最佳。

(二) 研究目的

綜合上述討論，本研究目的歸納如下：

1. 用 python 撰寫將字體檔案（truetype 檔案）轉換成 jpeg 檔案的程式
2. 用 python 撰寫將 jpeg 檔案轉換成要輸入模型的 obj 或 csv 檔案的程式
3. 用 tensorflow 實作 pix2pix 模型，並探討以下變因對模型的影響：
 - (1) 探討 U-Net 對模型成效的影響
 - (2) 探討訓練資料集大小對模型成效的影響
 - (3) 探討 Category Embedding 對模型成效的影響
4. 用 tensorflow 實作 CycleGAN 模型
5. 分析 pix2pix 模型和 CycleGAN 模型對字體風格轉換的成效差異

二、 研究方法與過程

(一) 研究設備及器材

1. 硬體

(1) 筆記型電腦 (CPU : Intel Core i7-7700HQ ; GPU : GeForce GTX 1060)

2. 軟體及工具

(1) ubuntu 16.04 (作業系統)

(2) Python 2.7 (程式語言)

(3) CUDA 8.0 & cudnn (GPU 運算技術及深層神經網路原式函式庫)

(4) tensorflow 1.2.1 (深度學習框架)

(5) 相關套件 : Pillow(PIL) 、 numpy 、 scipy 、 imageio

(6) 字體的 truetype 檔案(本研究主要使用下列兩種

字體，如圖一所示，其餘字體就不一一列舉)

微風迎客，軟語伴茶
微風迎客，軟語伴茶

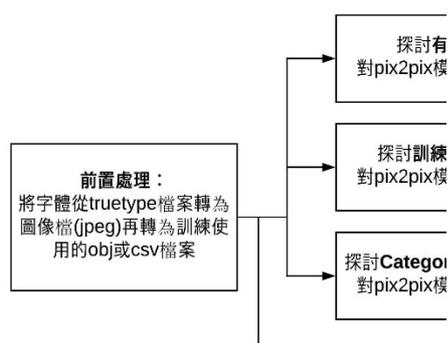
圖一：本研究所用字體示意圖

上方字體：文鼎細鋼筆行楷

下方字體：王漢宗中行書繁

使用此二字體的主要原因在於
兩字體有一定的相似特徵，卻又有
滿多不同之處，較能有效鑑別
兩種變因或方法的好壞

(二) 研究架構



圖二：實驗流程圖

本研究的架構是先將字體檔案轉為模型接受的 obj 或 csv 檔案，並實作出原始的 pix2pix 模型，然後探討不同方法及變因對 pix2pix 模型的影響，最後建立整合模型和 CycleGAN 模型進行比較。以下將分別說明詳細內容。

(三) 前置處理

1. pix2pix

(1) 將 truetype 檔案轉為 jpeg 檔案（程式名：font2img）：

I. 建 cjk.json 的字元集合檔案

II. 從 cjk.json 中隨機取出指定數量的字（例如：2000 字）

III. 運用 Pillow 套件將每個字的源字體和目標字體形態畫在圖片上（如圖三所示，左邊是目標字體，右邊是源字體）



圖三：圖片檔案示意圖
（左右兩字屬於同一張
圖片）

(2) 將 jpeg 檔案轉為訓練使用的 obj 檔案（程式名：package）：

I. 運用 cpickle 套件將 jpeg 檔案轉為 obj 檔案

2. CycleGAN

(1) 將 truetype 檔案轉為 jpeg 檔案（程式名：font2img）：

I. 程式同 pix2pix 的 font2img 程式，但因 CycleGAN 處理的是不成對資料集的問題，故配對的兩個字是不同的，而且是兩張分開的圖片



圖四：圖片檔案示意圖
（左右兩字是不同的兩
張圖）

(2) 將 jpeg 檔案轉為訓練使用的 csv 檔案（程式名：createCycleganDataset）：

I. 運用 click 和 csv 套件將 jpeg 檔案轉為 csv 檔案

(四) pix2pix

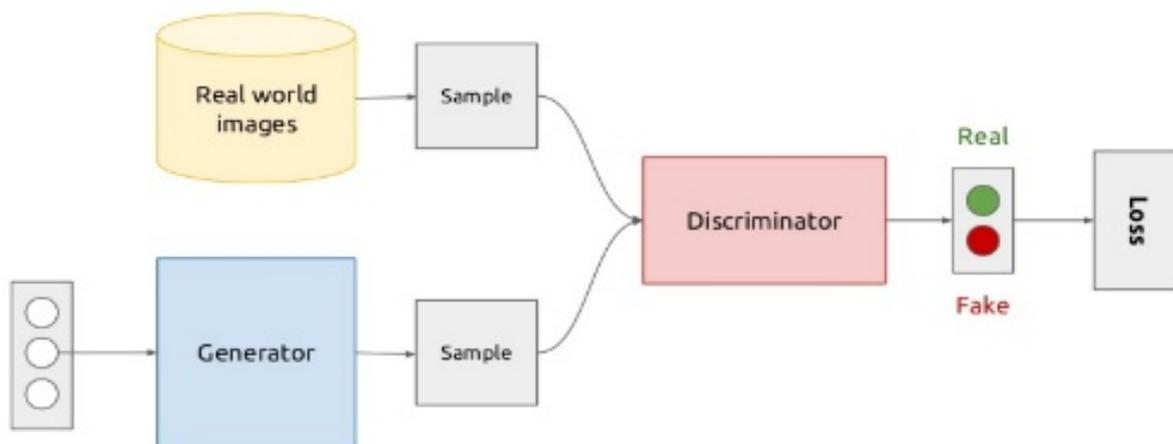
pix2pix 是針對圖像風格轉移領域的一種方法，其目標是在建立一個能處理絕大多數圖像風格轉移問題的大一統模型。pix2pix 的概念是以生成對抗網路（GAN）為基礎，並在模型上作些許調整所得出的一種方法。

1. 生成對抗網路 Generated Adversarial Network（GAN）

生成對抗網路簡稱 GAN，其啟發自博弈論中的二人零和博弈，GAN 模型中包含兩位博弈者：生成器（generator）和判別器（discriminator）。生成器 G 捕捉樣本資料的分布，用服從某一分布（均勻分布、高斯分布等等）的雜訊 z 生成一個類似真實訓練資料的樣本，目標是越像真實樣本越好；判別器 D 則是一個二分類器，估計一個樣本來自於真實訓練資料的機率，若其判斷樣本來自真實訓練資料，就輸出大機率，反之則為小機率，目標是能判斷每一個樣本是真是假。藉由生成器和判別器之間的對抗，可以讓生成器達到最佳的學習效果。

上述過程如下列公式及下圖所示：

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log 1 - D(G(z))] \quad - \text{Eq}(1)$$



圖五：生成對抗網路（GAN）模型示意圖

生成器 G 學習生成和訓練資料相似的圖片，判別器 D 學習判斷圖片的真偽

2. 模型

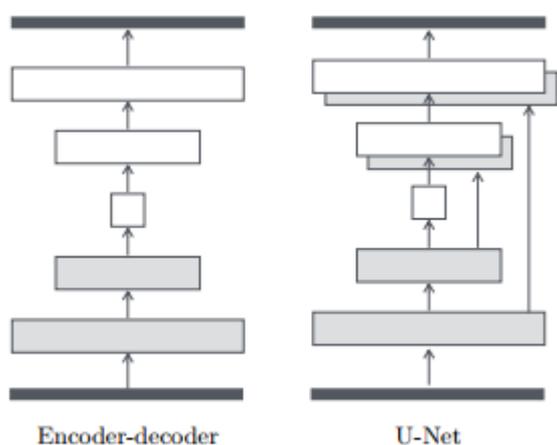
pix2pix 的模型與 Conditional GAN 大致相同，僅在幾處做了些許的改變：

(1) Conditional GAN (簡稱 cGAN) :

cGAN 與一般的 GAN 只有一點微小的差別，就是 GAN 的生成器是學習隨機雜訊 z 和輸出圖片 y 之間的映射函數 (即 $G: z \rightarrow y$)，但 cGAN 是學習輸入圖片 x 及隨機雜訊 z 和輸出圖片之間的映射函數 (即 $G: \{x, z\} \rightarrow y$)，其餘架構都與 GAN 相同。

(2) 改變一——加入「跳躍連結」:

大部分 cGAN 的生成器都是使用編碼—解碼器的網路架構 (encoder-decoder network)，在此架構中，輸入會通過一連串逐漸縮小的神經層，直到通過瓶頸層之後又逐漸變大 (如圖三左圖所示)。在這個網路中，每一項資訊都會流過所有的神經層，包括最小的瓶頸層。然而，對於圖像風格轉移問題，輸入與輸出的圖片勢必有些資訊是相同的，因此，若能讓這些資訊直接流過網路，而不必通過每一個神經層是較為合理的。實現此想法的具體作法是加入「跳躍連結」，設 n 為神經層的總數，在第 i 層和第 $n-i$ 層之間都加入一個「跳躍連結」，每一個跳躍連結可讓資訊直接流過。這個做法會讓模型變成「U-Net」的形式 (如圖六右圖所示)。



圖六：兩種生成器的網路架構

左邊是編碼—解碼器的網路架構，為一般 cGAN 所使用；右邊則是 pix2pix 模型做出的改變：在以瓶頸層為對稱的兩神經層中都加入「跳躍連結」，形成「U-Net」的網路形式

(3) 改變二——Markovian discriminator (PatchGAN) :

L1 loss 和 L2 loss 會造成生成出來的圖片十分模糊是已知的事實，但其實 L1 loss 固然在圖案較複雜的地方會造成模糊，在簡單的地方卻能準確的轉換，因此，在這裡只需要 GAN 的判別器去判斷圖案複雜處的真偽，剩下的只要依賴 L1 loss 即可。具體做法是設計一個判別器模型，它只會在 $N \times N$ 大小的方塊範圍內判別真偽，而此判別器會在圖片上循環遊走，進行判別。

3. 損失函數 (loss) 的設定

(1) Adversarial Loss

I. cGAN 的損失函數形式：

$$L_{CGAN}(G, D) = E_{x, y \sim p_{data}(x, y)}[\log D(x, y)] + E_{x \sim p_{data}(x), z \sim p_z(z)}[\log(1 - D(x, G(x, z)))] - Eq(2)$$

其中 G 的目標是讓 G(x) 看起來越像 y 訓練集的資料越好，而 D 則是要分辨假樣本 G(x) 和實際樣本，也就是說，G 想要最小化此損失函數，而 D 卻想要最大化它，以數學式表示則有以下形式：

$$G^* = \arg \min_G \max_D L_{CGAN}(G, D) - Eq(3)$$

II. 判別器的輸入沒有真實圖片的損失函數形式：

$$L_{GAN}(G, D) = E_{y \sim p_{data}(y)}[\log D(y)] + E_{x \sim p_{data}(x), z \sim p_z(z)}[\log(1 - D(G(x, z)))] - Eq(4)$$

(2) L1 loss

由於前述提到在模型中加入 L1 loss 可以幫助生成低複雜圖圖案處的圖片，故此處也附上 L1 loss 的數學形式：

$$L_{L1}(G) = E_{x, y \sim p_{data}(x, y), z \sim p_z(z)}[\|y - G(x, z)\|_1] - Eq(5)$$

(3) 整體損失函數

此模型的整體損失函數由上述的 adversarial loss 和 L1 loss 組合而成，由 λ 控制兩者的比重。而此模型的目標是要讓生成器 G 和 F 學習到讓生成出的圖片和實際資料越像的特徵，故可用以下數學形式加以表達：

$$G^* = \arg \min_G \max_D L_{CGAN}(G, D) + \lambda L_{L1}(G) - Eq(6)$$

4. 已做過的調整

(1) Category Embedding

根據設計師設計字體的經驗，讓神經網路同時學會多種字體風格是十分重要的，因為同時對多種風格建模可以讓編碼器接觸到更多漢字，且不僅限於一個目標字體，還包括所有字體的組合，此外，這種作法也能讓解碼器從其他自體學會同一種偏旁的不同寫法。

然而，這種作法會面臨的問題是同一個漢字可以出現在多種字體當中，而原本的 pix2pix 模型並沒有解決這種一對多的關係。因此，在此處需要使用「類別嵌入」(category embedding) 的作法，將不可訓練的高斯雜訊在資訊進入解碼器之前輸入神經網路，作為一種嵌入的風格 (style embedding)，如此一來，解碼器就會考慮原本的漢字以及此嵌入的風格來生成目標漢字。

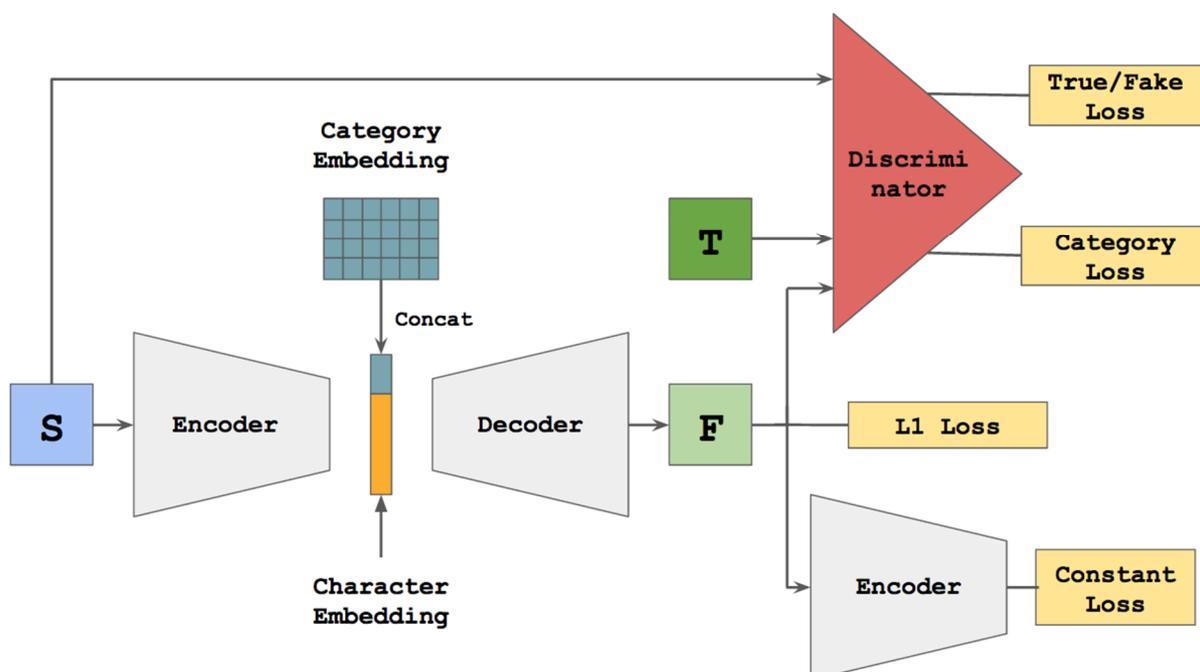
(2) Multi-class Category Loss

利用 Category Embedding 就能同時對多種風格的字體建模，然而此作法會衍生一個新問題：模型開始將各種風格混淆在一起，導致生成的字體甚麼都不像。

因此針對這個問題，在此處要引入 AC-GAN 模型中的 multi-class category loss，將此損失函數加到判別器上，以此去「懲處」風格混淆的情況，就能有效地保存每一種風格。

(3) Constant Loss

在模型中也引入了 DTN 神經網路中使用的 constant loss。此損失函數的基本原理就是：原本的字元和生成的字元對應的應該是同一個漢字，所以他們在圖片空間中的對應位置應該要十分相似。而結果證實 constant loss 能藉由讓編碼器保留生成漢字的識別性，縮小搜尋範圍，大大改善了收斂速度。

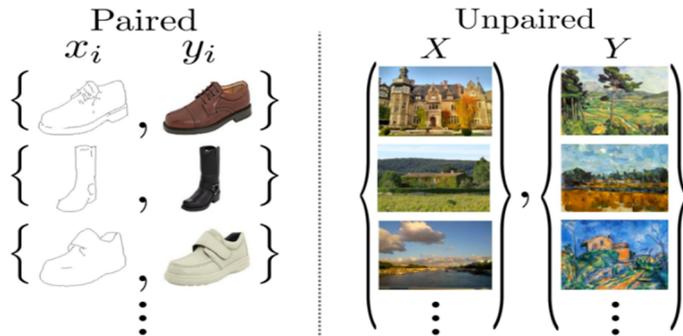


圖七：經調整後的 pix2pix 模型示意圖

整體架構與 cGAN 大致相同，而上述的三個調整：category embedding、category loss 和 constant loss 都加入到模型中，分別發揮各自的功能

(五) Cycle-Consistent Adversarial Networks (CycleGAN)

圖像風格轉移是計算機視覺領域的一個重要分支，常見的作法是使用成對的訓練圖片集，讓機器去學習輸入圖片與輸出圖片之間的映射關係。然而，對於許多任務，成對的訓練圖片集可能不易或無法取得，CycleGAN 即為針對此一問題的一個解決方法。



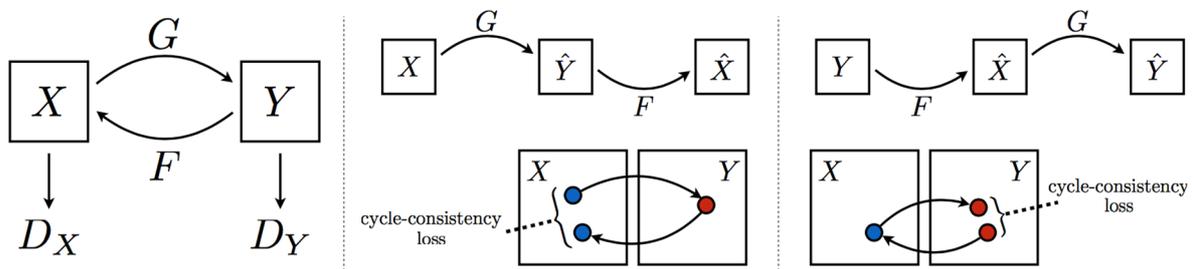
圖八：成對及非成對圖片示意圖

成對圖片間具有類似的特徵，如輪廓等等，而非成對圖片則否。

1. Cycle Consistency

CycleGAN 和 pix2pix 一樣都是由 GAN 衍生而來，但由於訓練圖片集並非成對，故無法保證 GAN 訓練出的模型會學習到如預期中的映射關係，因此模型中必須加入一些其他架構。在這裡會應用到一種關係：一張圖片經過一次變換及一次逆變換之後應該要變回原本的圖片，也就是說，若這裡有兩個轉換器 $G : X \rightarrow Y$ 和 $F : Y \rightarrow X$ ，則 G 和 F 互為彼此的逆變換，會有 $F(G(x)) \approx x$ 和 $G(F(y)) \approx y$ 的情形。此即 Cycle Consistency 的精髓，並應用到 CycleGAN 模型中得到很好的效果。

2. 模型



圖九：CycleGAN 模型示意圖

- (1) 此模型包含兩個映射函數 $G : X \rightarrow Y$ 和 $F : Y \rightarrow X$ ，以及相關的判別器 D_Y 和 D_X ， D_Y 會激勵 G 生成近似於 Y 訓練集中資料的樣本，而 D_X 則會激勵 F 生成近似於 X 訓練及資料

的樣本。

(2) 為了更有效的訓練映射函數，模型中也應用了 Cycle Consistency 的原理而放入了兩個 cycle consistency loss：

I. 正向的 cycle consistency loss： $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$ （如上中圖）

II. 反向的 cycle consistency loss： $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$ （如上左圖）

3. 損失函數（loss）的設定

(1) Adversarial Loss

對於映射函數 $G: X \rightarrow Y$ 和其判別器 D_Y ，其 adversarial loss 表示如下：

$$L_{GAN}(G, D_Y, X, Y) = E_{y \sim p_{data}(y)}[\log D_Y(y)] + E_{x \sim p_{data}(x)}[\log(1 - D_Y(G(x)))] - Eq(7)$$

其中 G 的目標是讓 $G(x)$ 看起來越像 Y 訓練集的資料越好，而 D_Y 則是要分辨假樣本 $G(x)$ 和實際樣本，也就是說， G 想要將此損失函數最小化，而 D_Y 卻想要最大化它，以數學式表示則有以下形式：

$$\min_G \max_{D_Y} L_{GAN}(G, D_Y, X, Y) - Eq(8)$$

此損失函數也以同樣的方式應用在 $F: Y \rightarrow X$ 和其判別器 D_X 上。

(2) Cycle Consistency Loss

對於前述所提的 cycle consistency 也定義了其損失函數，形式如下：

$$L_{cyc}(G, F) = E_{x \sim p_{data}(x)}[\|F(G(x)) - x\|_1] + E_{y \sim p_{data}(y)}[\|G(F(y)) - y\|_1] - Eq(9)$$

(3) 整體損失函數

此模型的整體損失函數由上述的 adversarial loss 和 cycle consistency loss 組合而成，形式如下：

$$L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F) - Eq(10)$$

其中 λ 控制了兩種損失函數的比重。而此模型的目標是要讓生成器 G 和 F 學習到讓生成的圖片和實際資料越像的特徵，故可用以下數學形式加以表達：

$$G^*, F^* = \arg \min_{G, F} \max_{D_X, D_Y} L(G, F, D_X, D_Y) - Eq(11)$$

(六) 數據類型

1. 主觀測試

(1) 方式：讓 30 人對模型轉換出的字體與目標字體的實際模樣進行比較並打分數(1~9 分)，分數越高表示預測結果和目標越相似，最後將 30 人所打的分數取平均即為主觀測試的結果

(2) 定義：

I. 少許特徵：明顯不像

II. 細節處模糊：筆畫的邊邊有點糊糊的或是其中一兩筆畫是糊的但沒有整個糊掉的感覺

III. 多數特徵：感覺蠻像的

IV. 有多數特徵但可明顯看出和目標的區別：感覺像但是很明顯覺得有地方不一樣

(3) 分數標準：

1 分：完全無法辨識出是甚麼字

2 分：很模糊但仔細看可看出是甚麼字

3 分：可一眼辨識出是甚麼字，但只具有目標字體的少許特徵且有多處模糊

4 分：可一眼辨識出是甚麼字，但只具有目標字體的少許特徵且細節處模糊(每一筆畫的邊邊糊糊的)

5 分：可一眼辨識出是甚麼字且十分清晰，但只具有目標字體的少許特徵(明顯不像)

6 分：可一眼辨識出是甚麼字且具有目標字體的多數特徵(感覺蠻像的)，但細節處模糊

7 分：可一眼辨識出是甚麼字、具有目標字體的多數特徵且十分清晰，但可明顯看出與目標的差別

8 分：可一眼辨識出是甚麼字、具有目標字體的多數特徵且十分清晰，但細看才可看出與目標的差別

9 分：與目標字體完全相同

2. 客觀測試：準確度測量

(1) 符號定義：

n_{ij} ：屬於類別 i ，但被預測為類別 j 的像素數

n_{cl} ：類別總數

t_i ：類別 i 總像素數

$$(2) \text{ Pixel accuracy} = \frac{\sum_i n_{ii}}{\sum_i t_i}$$

意義：預測正確的像素占總像素的比例

$$(3) \text{ Mean accuracy} = \frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{t_i}$$

意義：對圖片進行分類，分別對每一類計算 Pixel accuracy 之後再平均

$$(4) \text{ Mean IU} = \frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \quad (\text{全名：Mean Intersection over Union})$$

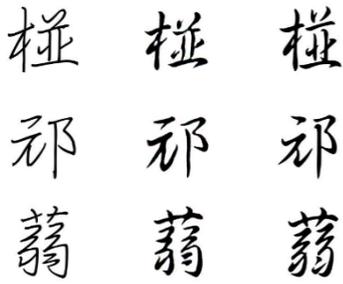
意義：對圖片進行分類，分別對每一類計算其像素的交集與並集之比值之後再平均

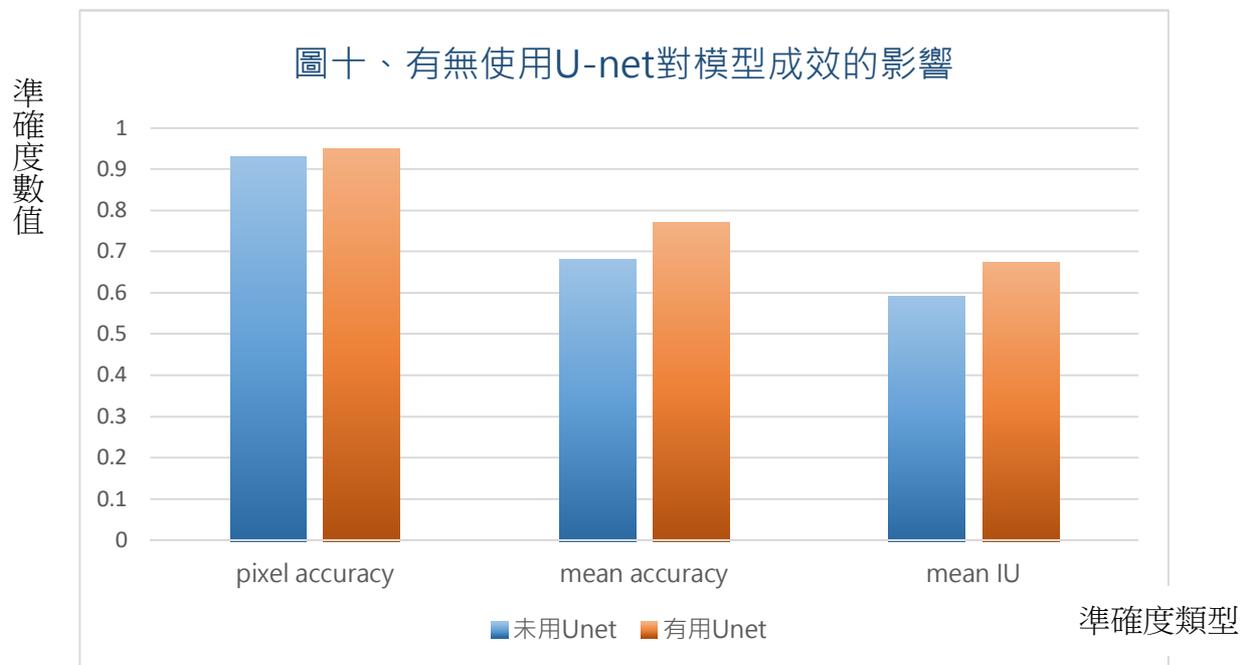
三、 研究結果與討論

以下表格中的圖片，若未特別說明，皆為以文鼎細鋼筆行楷為源字體、王漢宗中行書繁為目標字體訓練到第 50 個 epoch 的結果，且左側文字為源字體的模樣，中間文字為模型預測的結果，而右側文字則是目標字體的實際模樣。

(一) 探討 U-Net 對模型成效的影響

表一：實驗一結果圖片表格

| 有無使用 U-Net | 無 | 有 |
|----------------|---|---|
| 主觀測試結果 (分) | 1.7 | 7.8 |
| Pixel accuracy | 0.926473 | 0.9481 |
| Mean accuracy | 0.677112 | 0.769558 |
| Mean IU | 0.58786 | 0.671492 |
| 結果圖片 |  |  |



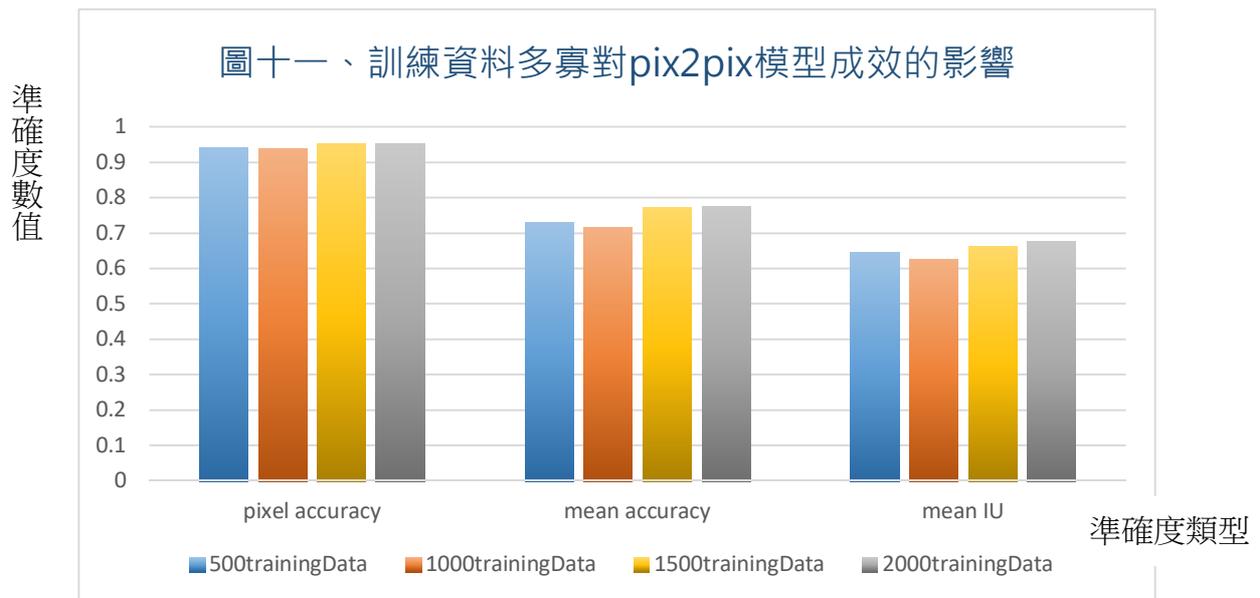
(二) 探討訓練資料集大小對模型成效的影響

用來訓練的十種字體：

王漢宗中行書繁、文鼎中特廣告體、華康流葉體、華康彩帶體、王漢宗鋼筆行楷繁、王漢宗細圓體繁、王漢宗粗鋼筆——標準、文鼎 POP-4、文鼎行楷碑體、文鼎細鋼筆行楷

表二：實驗二結果圖片表格

| 訓練圖片數 (張) | 500 | 1000 | 1500 | 2000 |
|----------------|----------|----------|----------|----------|
| 主觀測試結果 (分) | 4.2 | 4.0 | 6.1 | 7.8 |
| Pixel accuracy | 0.938293 | 0.935054 | 0.947661 | 0.9481 |
| Mean accuracy | 0.728955 | 0.71383 | 0.76927 | 0.769558 |
| Mean IU | 0.642259 | 0.621774 | 0.660242 | 0.671492 |
| 結果圖片 | | | | |



(三) 探討 Category Embedding (簡稱 CE) 對模型成效的影響

1. 源字體與目標字體形態差異較小的情況

源字體：文鼎細鋼筆行楷

目標字體：王漢宗中行書繁

表三：實驗三-1 結果圖片表格

| 有無使用 CE | 無 | 有 |
|----------------|----------|----------|
| 主觀測試結果 (分) | 7.8 | 6.9 |
| Pixel accuracy | 0.9481 | 0.943245 |
| Mean accuracy | 0.769558 | 0.748432 |
| Mean IU | 0.671492 | 0.660203 |
| 結果圖片 | | |

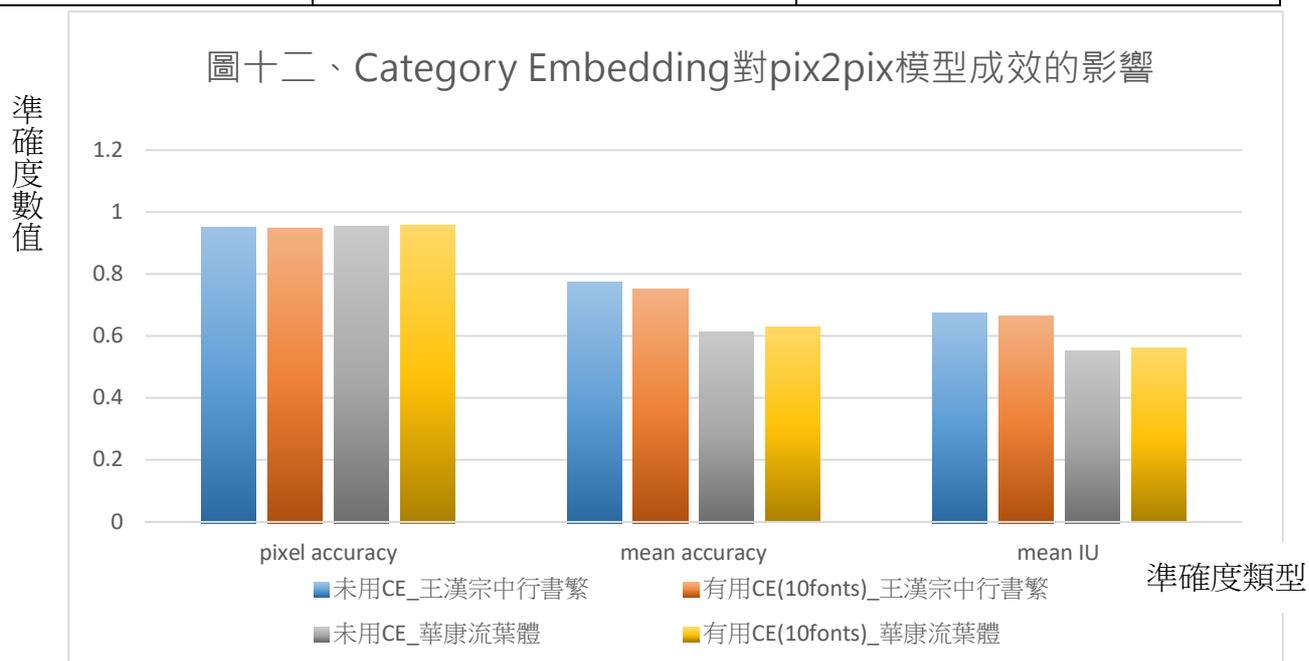
2. 源字體與目標字體形態差異較大的情況

源字體：文鼎細鋼筆行楷

目標字體：華康流葉體

表四：實驗三-2 結果圖片表格

| 有無使用 CE | 無 | 有 |
|----------------|---|---|
| 主觀測試結果 (分) | 5.6 | 6.3 |
| Pixel accuracy | 0.951246 | 0.954766 |
| Mean accuracy | 0.609385 | 0.625985 |
| Mean IU | 0.549814 | 0.559693 |
| 結果圖片 |  |  |



(四) 比較 pix2pix 模型及 CycleGAN 模型在字體風格轉換方面的成效

1. 源字體與目標字體形態差異較小的情況

源字體：文鼎細鋼筆行楷

目標字體：王漢宗中行書繁

表五：實驗四-1 結果圖片表格

| 模型種類 | pix2pix | CycleGAN |
|----------------|---|---|
| 主觀測試結果（分） | 7.8 | 5.8 |
| Pixel accuracy | 0.9481 | 0.911678 |
| Mean accuracy | 0.769558 | 0.624787 |
| Mean IU | 0.671492 | 0.517275 |
| 結果圖片 |  |  |

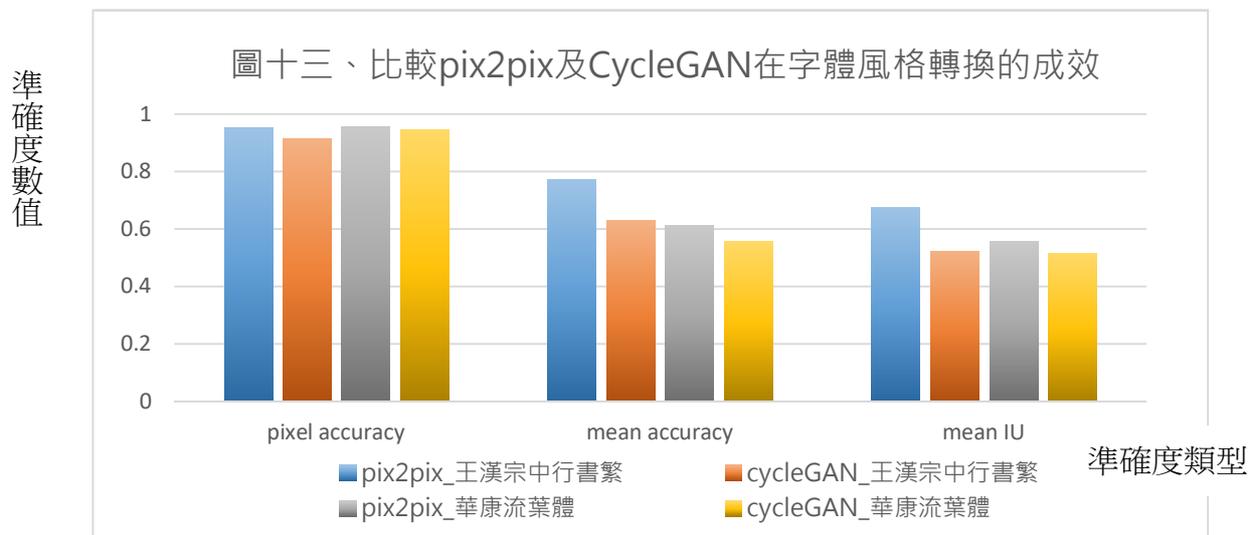
2. 源字體與目標字體形態差異較大的情況

源字體：文鼎細鋼筆行楷

目標字體：華康流葉體

表六：實驗四-2 結果圖片表格

| 模型種類 | pix2pix | CycleGAN |
|----------------|---|---|
| 主觀測試結果（分） | 5.6 | 5.2 |
| Pixel accuracy | 0.951246 | 0.939175 |
| Mean accuracy | 0.609385 | 0.551501 |
| Mean IU | 0.549814 | 0.509404 |
| 結果圖片 |  |  |



四、 結論與應用

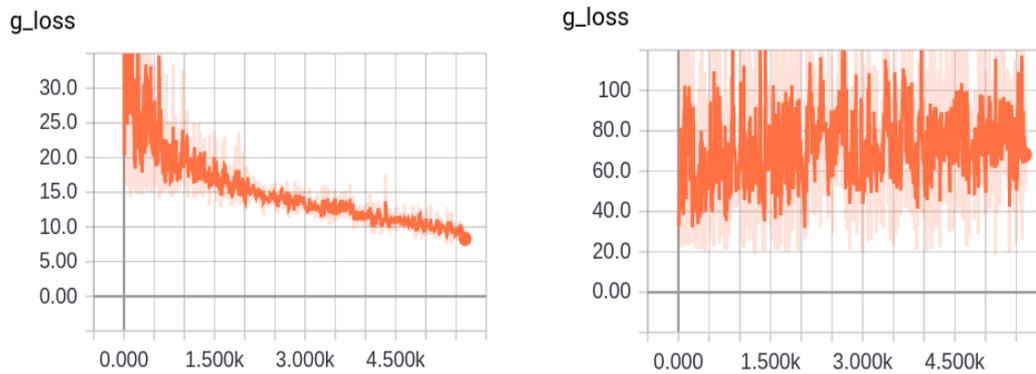
(一) 總結與應用

1. U-Net 能顯著改善模型的成效。
2. 訓練圖片數越多，模型轉換成效越好。
3. Category Embedding 的成效在源字體和目標字體形態差異小時並不明顯，但在兩字體差異大時就有不錯的成效。
4. 不論在源字體與目標字體差異小或大的情況，pix2pix 的轉換結果都比 CycleGAN 來的優異。

(二) 問題與討論

1. 探討 U-Net 對模型成效的影響

由實驗數據及結果圖片可看出，使用 U-Net 的結果明顯比未使用來的好很多，不僅字型的結構較為完整，預測的圖片也較為清晰。此外，觀察兩實驗生成器 loss 的趨勢圖（圖八）也可發現，有加入 U-Net 可讓 loss 逐漸變小，未加入則會讓 loss 不斷來回擺盪，無法有效降低 loss 值，達到較好的優化。由此我們可以推論，在 pix2pix 模型中，生成器在 encode 和 decode 的過程裡會使原有的一些特徵消失或模糊化，故在編碼器及解碼器之間加入 skip connection 使資訊連通能有效改善模型的成效。



圖十四：U-Net 實驗的生成器 loss 趨勢圖

圖左是有使用 U-Net 的實驗，圖右則是未使用的，能明顯看出有加 U-Net 能使 loss 逐漸降低，未加 U-Net 則無法

2. 探

討訓練資料集大小對模型成效的影響

由結果圖片及圖十三可看出，訓練圖片數越多，模型轉換的效果越好。訓練圖片只有 500 及 1000 張時，訓練出來的模型轉換效果是十分模糊的；而當訓練圖片有 1500 張時，我們比較可以看出轉換出的是甚麼字了，但有些字的細節部分還是不甚清楚；而用 2000 張圖片進行訓練時，模型預測的成效變得非常好，轉換出來的字與目標十分相似。

3. 探討 Category Embedding 對模型成效的影響

由實驗數據及結果圖片可看出，當源字體與目標字體的形態差異較小，是否使用 Category Embedding 對結果並沒有很大的影響；但當源字體與目標字體的形態差異較大，使用 Category Embedding 就有較大的幫助：在未使用 Category Embedding 的情況時，我們無法輕易辨識轉換出來的是哪個字，但在有使用的情況下，這種現象有明顯的改善，雖然成效也並非很好，但至少轉換出來的字型在架構上更接近實際字體，所測得的主觀測試及客觀測試數據也較佳。

4. 比較 pix2pix 模型及 CycleGAN 模型在字體風格轉換方面的成效

由實驗數據結果圖片可看出，在源字體與目標字體差異小時，pix2pix 模型的效果比起 CycleGAN 模型而言是較優的，因為 pix2pix 轉換的文字在圖片中的位置較精準，文字各方面的細節也更接近目標。我們推測出現此現象是因為 CycleGAN 的輸入缺乏與源字體一對一對應的

圖片，導致源字體與目標字體的映射關係較難建立，因此模型預測效果較差。此外，CycleGAN模型的訓練結果有與訓練資料過於相似的情形，例如「喫」這個字的「共」部分（如圖十五），文鼎細鋼筆行楷有多了一撇，而實際上王漢宗中行書繁是沒有那一撇的，但預測結果卻是有那一撇的，這是待改善的問題。另外，在源字體與目標字體差異大時，可以發現兩種模型轉換出來的效果都不是太好，預測出來的字體的筆劃會在不正確的位置，導致整個字難以辨識，在這種情況下仍需以 Category Embedding 融入多種字體進行訓練會有較好的成果。



圖十五：CycleGAN 問題示意圖

(三) 未來展望

1. 模型在源字體於目標字體差異大時成效仍不佳，希望能想出其他方法進行改善

五、 參考文獻

- [1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio(2014). Generative Adversarial Networks. stat.ML. arXiv:1406.2661v1.
- [2] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Imagenet-to-image translation with conditional adversarial networks. In CVPR, 2017.
- [3] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In CVPR, 2017.
- [4] Augustus Odena, Christopher Olah, Jonathon Shlens. Conditional Image Synthesis with Auxiliary Classifier GANs. In stat.ML.
- [5] Yaniv Taigman, Adam Polyak, Lior Wolf. Unsupervised cross-domain image generation. In cs.CV.
- [6] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat(2017). Google's Multilingual Neural Machine Translation System: Enabling

Zero-Shot Translation. cs.CL. arXiv:1611.04558v2.

[7] kaonashi-tyc(2017). zi2zi: Master Chinese Calligraphy with Conditional Adversarial Networks.

kaonashi-tyc.github.io/2017/04/06/zi2zi.html

[8] Sword York(2016). Generative Adversarial Networks.

blog.slinuxer.com/2016/10/generative-adversarial-networks

【評語】 190016

此作品運用生成對抗網路 (GAN) 的 neural networks (NN) 來產生不同風格的字體，藉由先提供約 2000 筆符合新風格的字體來訓練此 GAN NN，之後就可以讓此 GAN NN 來自動產生符合此風格字體的另外兩萬六千多字。

此作品用不同的方式來產生較想要的風格字體，在實驗中比較這些不同方式的優缺點，並用不同的效能指標來進行客觀和主觀上的比較，建議未來可以應用在不同的語言文字或領域。