

# 中華民國第 62 屆中小學科學展覽會

## 作品說明書

---

高級中等學校組 電腦與資訊學科

### 團隊合作獎

052504

字由字在—利用深度學習生成個人化字體

學校名稱：國立花蓮高級中學

|   |                  |
|---|------------------|
| 作者：<br><br>高二 周隆宇<br><br>高二 嚴得愷<br><br>高二 黃浩昕 | 指導老師：<br><br>黃楨喻 |
|---|------------------|

關鍵詞：深度神經網絡、風格矩陣、個人化字體

## 摘要

本研究旨在利用深度學習實作改變字體風格的模型，試圖讓生活中處處可見的文字變得活潑、生動，並省去許多設計字體所需的時間、精力。我們利用了風格矩陣、對抗式生成網絡(GAN)等概念設計了 3 種模型，並在實際運行後進行分析與比較。

VGG 模型運用了卷積神經網絡的概念，擷取特徵以生成風格矩陣並將其進行字體風格轉換。Bicycle-GAN 模型結合了 Conditional GAN 和 VAE 的優點，能夠生成多樣風格的字體。Zi2Zi 模型則是在 pix2pix 的方法改進部分缺點，能讓生成的字體更逼真，且具有顯著的成效。

經過多次的嘗試和失敗經驗，我們發現 VGG、Bicycle-GAN、Zi2Zi，3 種模型在字體風格轉換上具有相當的潛力，在經過適當的後處理之後，能夠在耗費少量時間和樣本的情況下，生成實際套用在電腦中的個人化字體。

## 壹、研究動機

文字是人類用以溝通的重要媒介，我們的生活處處都離不開文字。然而，隨著現代科技的發達，文字不再只出現在書本、白紙上，電腦、手機螢幕上更能看到密密麻麻的文字。然而，現今的我們所見的字體大多是新細明體、標楷體或微軟正黑體等制式化的字體，因此相較於以往的手寫書信，現代的電子郵件更顯得冷冰無情。

然而，設計一套字體又需要花費相當大量的時間和精力才能完成，想設計一套屬於自己的個人化字體相當困難。因此，我們希望透過深度神經網絡的方式省去繁瑣的字體設計過程，將個人化中文字體的實踐化為可能。這個模型不僅能將制式化的電腦字體化為獨特的個人化字體，還能應用在現今十分常見的電子文件中，應用層面可說是非常廣泛。

## 貳、研究目的

- 一、嘗試以 VGG-19 作為深度卷積網路的框架，透過風格矩陣生成個人化中文字體。
- 二、嘗試改良並利用複合式結構——Bicycle-GAN 作為模型生成個人化中文字體。
- 三、嘗試利用 Zi2Zi 模型作為條件式 GAN 的模型，生成個人化中文字體。
- 四、比較各方法的成果並分析成效，分析各種方法的優劣。
- 五、對生成的個人化字體進行後處理，使其符合做為電腦字體的樣式。

## 參、文獻回顧

### 一、Visual Geometry Group (VGG)

- (一) 簡介：VGG 是 Keras 收錄的模型(model)之一，模型使用了大量資料作訓練，且使用非常多層的處理，已使用 ImageNet 100 中萬張圖片進行訓練，使用者不需自行訓練模型，稱為預訓練模型(pre-trained models)
- (二) 架構：VGG 是經典的卷積神經網路(Convolutional Neural Network, CNN)模型，並添加了許多隱藏層，以便進行深度學習。
- (三) 特色：VGG 的主要方法是使用更多的隱藏層，使用大量的圖片訓練，進而提高準確率至 90%。VGG16/VGG19 分別為 16 層(13 個卷積層及 3 個全連接層)與 19 層(16 個卷積層及 3 個全連接層)

### 二、卷積神經網絡

- (一) 簡介：卷積神經網絡 (Convolutional Neural Network, CNN) 是一種前饋神經網絡，模擬人類的視覺處理過程來進行設計的模型，適合用來處理圖像資料進行訓練。
- (二) 原理：在普通的神經網絡架構中，添加數個甚至數十個特殊的隱藏層，包括：池化層、卷積層、完全連接層……。利用這些隱藏層進行特徵提取等作業，希望能模擬生物判斷影像的步驟。
- (三) 應用：現今卷積神經網絡的應用廣泛，包括：自然語言處理、視訊分析、影像辨識、探索藥物等等。

### 三、風格矩陣(Style Matrix)

- (一) 概敘：根據 A Neural Algorithm of Artistic Style(Leon A. Gatys, Alexander S. Ecker, Matthias Bethge,2015)所述，風格矩陣是表示一圖片特徵的方式，並可用於風格轉換之目的。
- (二) 名詞解釋
  1. Feature map：即資料的維度，例如黑白圖片只有 1 個 feature map，而彩色圖片則有 3 個 feature map。
  2. Gram Matrix：是指不同 Feature map 矩陣內積後的結果，可以用來表示兩維度之間的相關性。

3. Filter：是卷積層中的重要部分，負責提取部分範圍內的局部特徵。
4. Style Loss：表示一圖片之風格損失函數，可透過風格矩陣進行定義
5. Content Loss：表示一圖片之內容損失函數。內容是代表宏觀圖片的相似性，
6. 而非局部的特徵差異。可透過原始圖片數值資料進行定義。

### (三) 用途

1. Style Loss：利用預訓練之 VGG-19 模型對圖片進行特徵提取。可能存在多個 filter 進行不同類型的特徵提取，形成風格矩陣，再進行內積運算形成 Gram Matrix，最後再利用 Gram Matrix 進行單層損失函數的定義，以進行下一步的模型優化。
2. Content Loss：直接利用原始圖片的色塊數值進行數學運算求得。用以表達 2 張不同圖片內容的大略差異。

### (四) 實際應用：

用於進行圖片之風格轉換，例如將照片賦予孟克吶喊畫作之風格，可生成具有此畫作風格之圖片。另外，亦可使用多個畫作樣本，只須設定各樣本之間的權重即可。

## 四、Pix2pix

- (一) 概敘：Pix2pix 是一種 Conditional GAN(CGAN)，主要用於圖像和圖像之間的映射，稱為圖像轉譯。
- (二) 方法：pix2pix 有著 CGAN 的特色，運行 GAN 的過程中有輸入圖片，而不是單純透過雜訊向量進行訓練。因此 Pix2pix 能夠在輸入圖片的基礎上進行訓練。
- (三) U-Net 結構：是一個優化 Encoder-Decoder 的跳躍式連結。由於 Encoder-Decoder 結構具有對稱性，故第  $i$  層和第  $n-i$  層大小大致相同。U-Net 結構就是將第  $i$  層和第  $n-i$  層直接連接起來，目的是確保輸入圖片和生成圖片的相似性。
- (四) 特性：由於 Pix2pix 是圖像的一對一映射，我們在訓練時必須採用成對的(Paired)資料進行訓練。也因此，pix2pix 應用的範圍受到訓練集的限制，必須有充足的訓練集才能產生有意義的輸出結果。

## 肆、研究器材或軟體

### 一、硬體：

(一) 桌上型電腦(CPU：i7-10700；GPU：Nvidia Geforce GTX 1650)

(二) 手寫板

### 二、軟體：

(一) Windows 10 (作業系統)

(二) Ubuntu(作業系統)

(三) Anaconda(虛擬環境)

(四) python(程式語言)

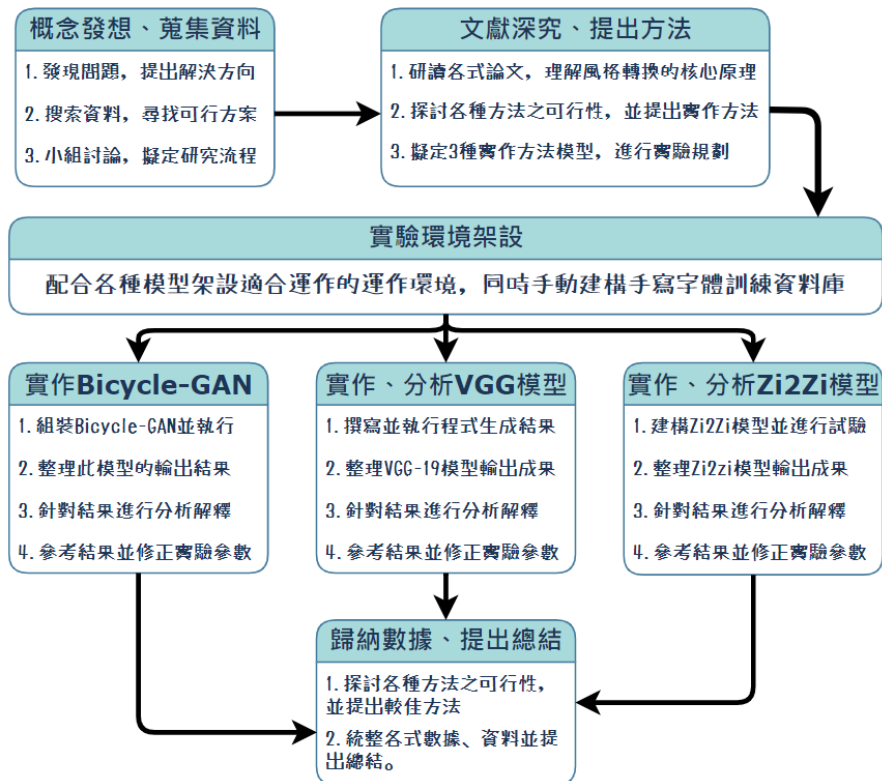
(五) Onenote(輸入手寫字體)

(六) TensorFlow(深度學習框架)

(七) CUDA & cudnn (GPU 運算套件)

## 伍、研究過程或方法

### 一、研究流程圖

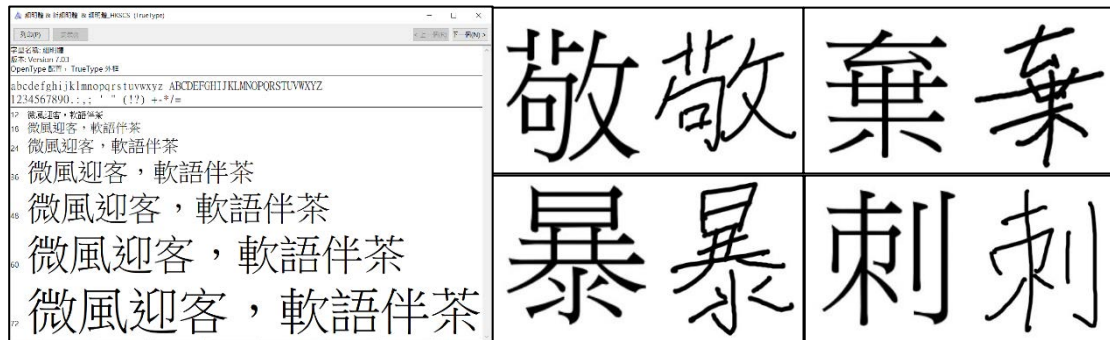


▲圖一 研究流程圖

## 二、架設模型訓練集(Dataset)

本研究用來作為模型學習的訓練資料主要有 2 種：

- (一) 利用中文字體檔案：truetype(.ttf)檔中的所有字體作為素材進行訓練。
- (二) 利用手寫板書寫個人化手寫字體，並和普通電腦字體組合成訓練集。其中，我們選用了 4 位同學的手寫字體作為樣本，並搭配新細明體、標楷體製作成成對的訓練集，資料數量有將近 2000 筆。



▲圖二 研究流程圖

## 三、VGG-19 模型

### (一) 損失函數的定義

1. 在本模型中，總損失函數是透過 Style Loss 和 Content Loss 共同定義的。
2. Style Loss :
  - (1) 將雜訊向量(生成圖片)分為紅、綠、藍三個維度並將樣本圖片及原始圖片調整尺寸進行預處理。
  - (2) 先利用 filter 對圖片提取特徵，將所有 feature map 的特徵矩陣進行內積，最後得到 Gram 矩陣，也就是風格矩陣。

$$G = A^T A = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_n^T \end{bmatrix} [\mathbf{a}_1 \ \mathbf{a}_2 \ \cdots \ \mathbf{a}_n] = \begin{bmatrix} \mathbf{a}_1^T \mathbf{a}_1 & \mathbf{a}_1^T \mathbf{a}_2 & \cdots & \mathbf{a}_1^T \mathbf{a}_n \\ \mathbf{a}_2^T \mathbf{a}_1 & \mathbf{a}_2^T \mathbf{a}_2 & \cdots & \mathbf{a}_2^T \mathbf{a}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_n^T \mathbf{a}_1 & \mathbf{a}_n^T \mathbf{a}_2 & \cdots & \mathbf{a}_n^T \mathbf{a}_n \end{bmatrix}$$

- (3) 接著利用以下公式合併同一隱藏層不同維度之損失函數：

$$L_{style}^{[l]}(S, G) = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^{[l](S)} - G_{ij}^{[l](G)})^2$$

(4) 再利用以下公式合併不同層之間的損失函數，得到風格損失函數(Style Loss)：

$$L_{style}(S, G) = \sum_l w^{[l]} \cdot L_{style}^{[l]}(S, G)$$

3. Content Loss：

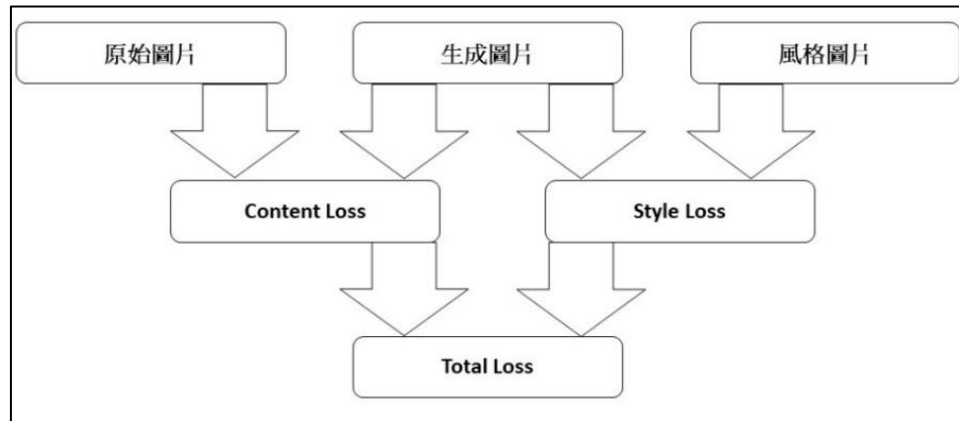
(1) 概敘：內容損失函數是比較兩圖片的宏觀內容差距，用以表示兩者內容的差異程度。

(2) 定義： $L_{content}(C, G) = \frac{1}{2} \sum_{i,j} (a_{ij}^{[l](C)} - a_{ij}^{[l](G)})^2$

(3) 用途：此處用以計算雜訊向量和原始圖片的差距，期望逐漸減少損失函數

4. Total Loss：總損失函數是 Style Loss 和 Content Loss 的加權平均：

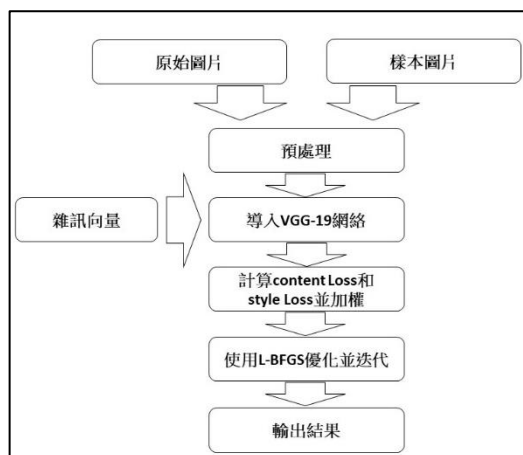
$$L_{total}(C, S, G) = a \cdot L_{content}(C, G) + b \cdot L_{style}(S, G)$$



▲圖三 VGG 模型損失函數計算

(二) 模型訓練之優化方式

1. 我們使用的優化方法是 L-BFGS 算法。L-BFGS 是解無約束非線性規劃問題最常用的方法，具有收斂速度快、內存開銷少等優點，在機器學習各類算法中常有它的身影。



圖四：VGG 模型運作流程圖

#### 四、Bicycle-GAN 模型

(一) 概敘：Bicycle-GAN 是一種一對多的圖像轉譯模型，在圖像轉譯的基礎上實現多種風格的生成。從模型結構來說，Bicycle-GAN 是由 cVAE-GAN 和 cLR-GAN 共同組成的複合結構。

(二) cVAE-GAN(Conditional Variational AutoEncoder GAN)：在 VAE 生成模型的基礎上添加條件輸入(Condition)，使得生成器生成的結果符合輸入的條件的對抗是生成網絡。意即輸入有 2 種：雜訊向量和參考圖片，在 VAE 可生成多樣風格的優點上搭配參考圖片的內容。

(三) cLR-GAN(Conditional Latent Regressor GAN)：將生成器生成出來的圖片輸入到 Encoder 網絡內，使最後輸出的圖片更加接近原本輸入的圖片，意即提升生成圖片的真實性。

(四) 損失函數設定：

1. 基礎的條件式生成器：輸入雜訊向量和條件圖片

(1) 生成器和鑑別器的損失函數為：

$$L_{GAN}(G,D) = E_{A,B \sim p(A,B)}[\log(D(A,B))] + E_{A \sim p(A), z \sim p(z)}[\log(1-(D(A,G(A,z))))]$$

(2) L1 損失可以代表重建後的圖像輪廓

$$L_1^{\text{image}}(G) = E_{A,B \sim p(A,B), z \sim p(z)} \|B - G(A,z)\|_1$$

(3) 由於希望生成器(Generator)能將損失函數降低、鑑別器(Discriminator)能增加損失函數，因此將上述式子綜合可得：

$$G^* = \arg \min(G) \max(D) L_{GAN}(G,D) + \lambda L_1^{\text{image}}(G)$$

2. cVAE-GAN：

(1) KL 損失是計算編碼器生成結果和預先準備的數據分布  $N(z)$  的差異：

$$L_{KL}(E) = E_{B \sim p(B)} [D_{KL}(E(B) || N(0,I))]$$

$$\text{而其中：} D_{KL}(p||q) = - \int p(z) \log \frac{p(z)}{q(z)} dz$$

(2) 而 GAN 部分的損失函數：

$$L_{GAN}^{VAE} = E_{A,B \sim p(A,B)} [\log(D(A,B))] + E_{A,B \sim p(A,B), z \sim E(B)} [\log 1 - D(A,G(A,z))]$$

(3) 總結後可得：



$$G^*, E^* = \arg \min(G, E) \max(D) L_{GAN}^{VAE}(G, D, E) + \lambda L_1^{VAE}(G, E) + \lambda_{KL} L_{KL}(E)$$

3. cLR-GAN :

(1) 首先是雜訊向量和 GAN 的反向矯正，利用 L1 損失實作：

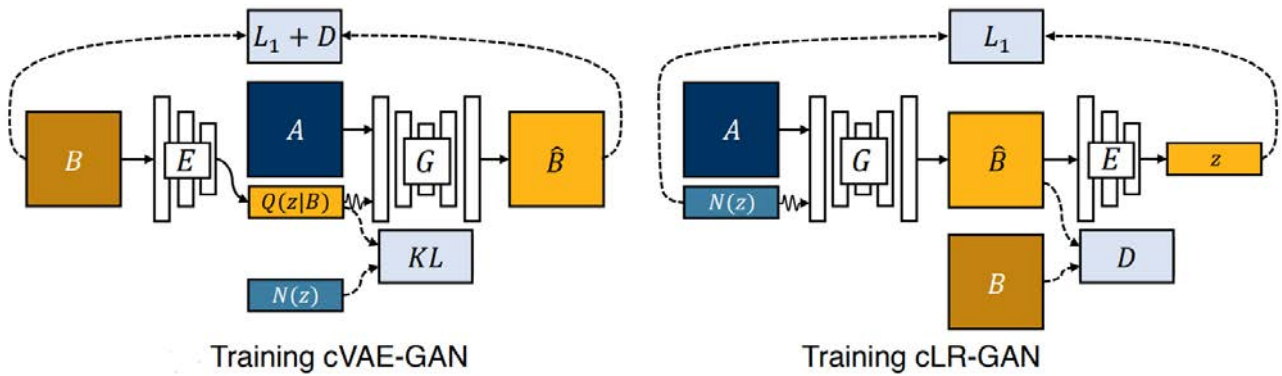
$$L_1^{latent}(G, E) = E_{A \sim p(A), z \sim p(z)} \|z - E(G(A, z))\|$$

(2) 接著再整合為：

$$G^*, E^* = \arg \min(G, E) \max(D) L_{GAN}(G, D) + \lambda_{latent} L_1^{latent}(G, E)$$

4. 統整：將所有損失函數整合在一起進行訓練

$$G^* E^* = \arg \min(G, E) \max(D) L_{GAN}^{VAE}(G, D, E) + \lambda L_1^{VAE}(G, E) + L_{GAN}(G, D) + \lambda_{latent} L_1^{latent}(G, E) + \lambda_{KL} L_{KL}(E)$$



▲圖五 Bicycle-GAN 模型運作示意圖

## 五、Zi2Zi 模型

(一) 概述：

Zi2Zi 模型，顧名思義就是進行字體和字體間風格轉換的模型。Zi2Zi 模型大致參考了 pix2pix 和 Conditional GAN 的作法，並解決了 pix2pix 無法進行一對映射的缺點。

(二) 類別嵌入法：

在 Encoder 和 Decoder 間插入不可訓練的雜訊向量並和隱編碼一起送入 Decoder 進行解碼，可達到生成多種風格字體的效果。

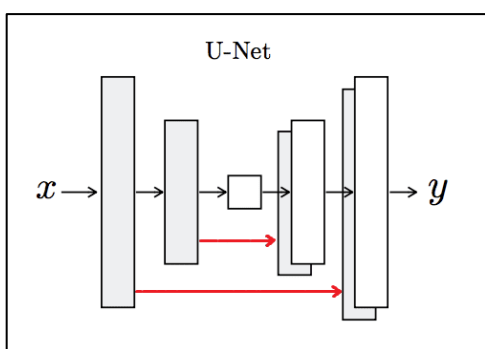
(三) Patch-GAN：

在一般的對抗式生成網絡中，判別器(discriminator)只輸出一個數字，代表判別為真或假的機率。而 Patch-GAN 是一種特別的判別器，輸出的是一個矩陣，代表圖片各部

分為真或假的機率。使用時再將矩陣的各個值進行平均，可以參酌各部分的相似度，藉以增加生成圖片的品質。

#### (四) U-Net 結構：

在 Encoder-Decoder 網絡中，輸入和輸出具有部分相同的資訊，是不需要透過模型進行轉換的。因此，我們需要一些跳躍性的連結，使資訊能夠直接地通過網絡。由於 Encoder-Decoder 結構具有對稱性，故第  $i$  層和第  $n-i$  層大小大致相同，U-Net 結構就是將第  $i$  層和第  $n-i$  層直接連接起來，確保輸入圖片和生成圖片的相似性。



圖六：U-Net 結構在 Encoder-Decoder 網絡示意圖

#### 六、輸出成果後處理

- (一) 為了讓輸出成果能夠真正應用在電腦字體中，我們必須消除輸出圖片的白色背景和雜訊並做相關後處理，以達成我們實際應用在電腦做為電腦字體的目的。
- (二) 我們會將所有成果輸出到同一個目錄下，批次利用 python 中的 Pillow 套件讀取圖片，並解析其像素值。我們會將「白色」和「接近白色」的區塊挖除，即可生成不具背景的个人化手寫字體圖片，以便合併成一套個人化字體。



圖七：輸出成果後處理前後比較圖

## 陸、研究結果

### 一、實驗 VGG 模型在生成字體之成效

在這系列的實驗中，我們使用「新細明體」圖檔作為原始圖片，並利用手寫板書寫個人化字體並匯出成圖檔作為樣本圖片，再導入程式中進行訓練與生成。在此實驗中，我們試驗使用的字是「永」字，所謂「永字八法」意味者「永」字中包含所有的筆順，可增加實驗的代表性。

實驗中，我們將圖片輸入模型進行優化 60 次。由於中間的優化次數並沒有顯著的差異，因此我們只放上優化 1 次、30 次、60 次的圖片，即下圖的 1 號、30 號、60 號輸出。

(一) 組別一：組別一是原始的一組，使用正常的手寫字體作為樣本，可作為實驗對照組

| 原始圖片   | 樣本圖片   | 1 號輸出  | 30 號輸出  | 60 號輸出   |
|--|--|--|---|--|
|  |  |  |  |  |

(二) 組別二：組別二以較粗的筆法書寫「永」字，操作變因為字體粗細

| 原始圖片  | 樣本圖片  | 1 號輸出   | 30 號輸出   | 60 號輸出  |
|---|---|---|--|---|
|  |  |  |  |  |

(三) 組別三：組別三使用和永幾乎無關的「談」字作為原始圖片，操作變因為原始字型

| 原始圖片  | 樣本圖片  | 1 號輸出   | 30 號輸出   | 60 號輸出  |
|---|---|---|--|---|
|  |  |  |  |  |

(四) 組別四：組別四使用一半和永字相關的「詠」字，操作變因為原始字型

| 原始圖片 | 樣本圖片 | 1 號輸出 | 30 號輸出 | 60 號輸出 |
|------|------|-------|--------|--------|
| 詠    | 永    | 詠     | 詠      | 詠      |

(五) 組別五：組別五使用黃色的樣本字體作為樣本圖片，操作變因為字體顏色

| 原始圖片 | 樣本圖片 | 1 號輸出 | 30 號輸出 | 60 號輸出 |
|------|------|-------|--------|--------|
| 永    | 永    | 永     | 永      | 永      |


(六) 組別六：組別六使用扭曲的手寫字體作為樣本字體，操作變因為線條形狀

| 原始圖片 | 樣本圖片 | 1 號輸出 | 30 號輸出 | 60 號輸出 |
|------|------|-------|--------|--------|
| 永    | 永    | 永     | 永      | 永      |

(七) 組別七：組別七使用黑底白字進行訓練，操作變因為背景顏色

| 原始圖片 | 樣本圖片 | 1 號輸出 | 30 號輸出 | 60 號輸出 |
|------|------|-------|--------|--------|
| 永    | 永    | 永     | 永      | 永      |


(八) 第八組：組別八使用火焰作為特效樣本，色系鮮明

| 原始圖片 | 樣本圖片  | 1 號輸出 | 10 號輸出 | 20 號輸出 |
|------|---|-------|--------|--------|
| 永    |  | 永     | 永      | 永      |


(九) 第九組：組別九以閃電作為特效樣本，線條對比清楚

| 原始圖片 | 樣本圖片  | 1 號輸出 | 10 號輸出 | 20 號輸出 |
|------|---|-------|--------|--------|
| 永    |  | 永     | 永      | 永      |

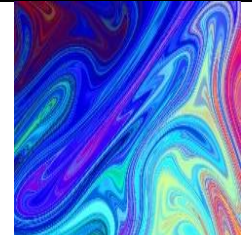
(十) 第十組：組別十以霓虹漩渦作為特效樣本，形狀及色彩十分特別

| 原始圖片 | 樣本圖片  | 1 號輸出 | 10 號輸出 | 20 號輸出 |
|------|---|-------|--------|--------|
| 永    |  | 永     | 永      | 永      |

(十一) 第十一組：組別十一以雙色火焰作為特效樣本，其形狀和焰色十分獨特

| 原始圖片 | 樣本圖片  | 1 號輸出 | 10 號輸出 | 20 號輸出 |
|------|---|-------|--------|--------|
| 永    |  | 永     | 永      | 永      |

(十二) 第十二組：組別十二以扭曲圖案作為特效樣本，形狀獨具一格

| 原始圖片 | 樣本圖片  | 1 號輸出 | 10 號輸出 | 20 號輸出 |
|------|---|-------|--------|--------|
| 永    |  | 永     | 永      | 永      |

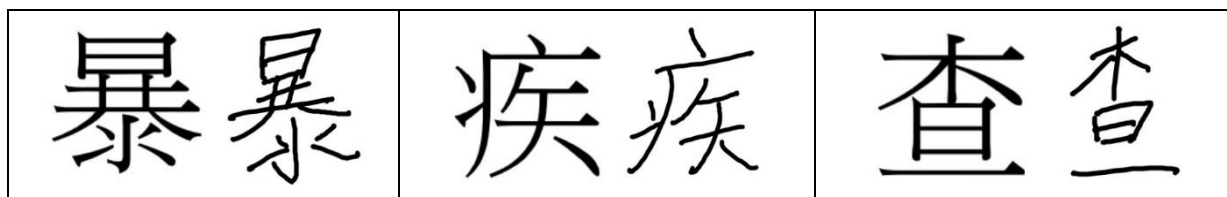
(十三) 第十三組：組別十三以紫色閃電網作為特效樣本，圖片色調有強烈對比性。

| 原始圖片 | 樣本圖片  | 1 號輸出 | 10 號輸出 | 20 號輸出 |
|------|---|-------|--------|--------|
| 永    |  | 永     | 永      | 永      |

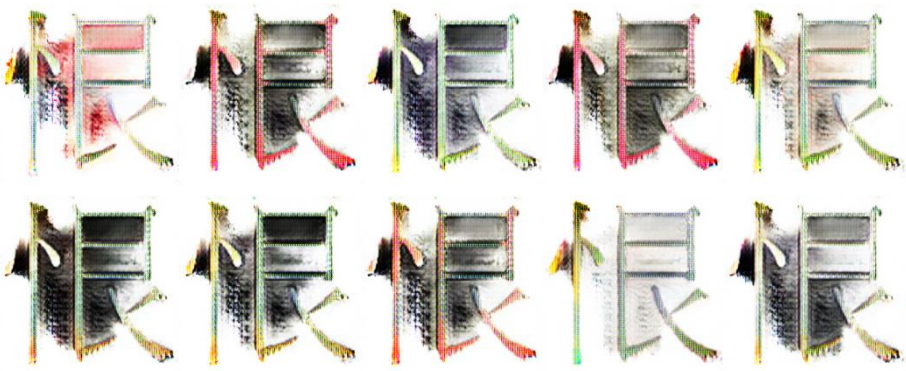
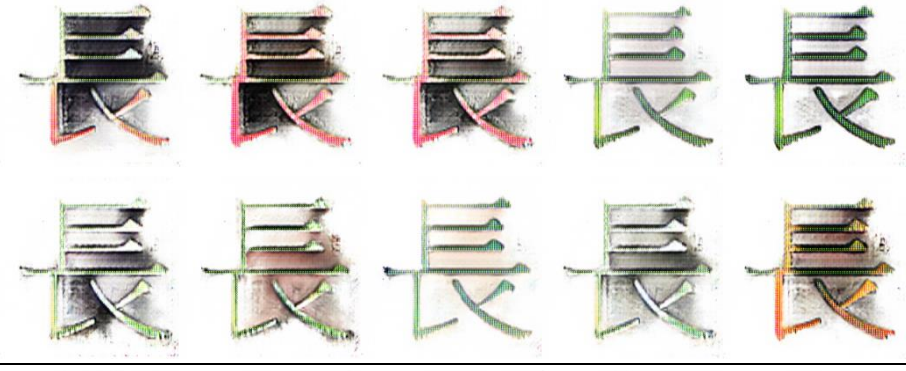

## 二、實驗 Bicycle-GAN 模型在生成字體之成效

在這系列實驗中我們使用先前建立的成對(paired)資料(手寫文字圖片和電腦文字圖片)庫隨機挑選樣本並使用架設好的 Bicycle-GAN 模型進行訓練並輸出 10 個結果。表格中的原始圖片就是上述 Bicycle-GAN 模型中的條件圖片。

如下圖所示，我們使用的訓練資料是由電腦中文字體和個人手寫字體所組成的成對圖片所組成。



▲成對資料集形式

| 原始圖片 | 輸出圖片   |
|------|--|
| 恨    |   |
| 長    |  |
| 刺    |  |

|   |  |
|---|--|
| 他 |  |
| 皮 |  |
| 帥 |  |
| 氣 |  |
| 騎 |  |

### 三、實驗 Zi2Zi 模型在生成字體之成效

在這系列的實驗中我們使用了不同中文字體的 truetype 字體檔案作為資料庫，將 truetype 字體檔案轉換成如實驗二的成對資料形式後再輸入 Zi2Zi 模型進行訓練。我們使用的是成對的(paired)訓練資料。在實驗中，我們將 90%的資料集作為訓練資料，10%的資料集做為訓練資料。

(一) 第一組實驗使用了 500 筆成對資料進行訓練，此實驗的操作變因是訓練次數，我們使用了較具代表性的組別進行展示，即下表中的第 5、30、45、50 號(次)輸出。

| 5號 | 30號 | 45號 | 50號 |
|----|-----|-----|-----|
| 茶  | 茶   | 茶   | 茶   |
| 戍  | 戍   | 戍   | 戍   |
| 疾  | 疾   | 疾   | 疾   |
| 差  | 差   | 差   | 差   |
| 午  | 午   | 午   | 午   |
| 瑯  | 瑯   | 瑯   | 瑯   |
| 空  | 空   | 空   | 空   |
| 彼  | 彼   | 彼   | 彼   |
| 己  | 己   | 己   | 己   |
| 壬  | 壬   | 壬   | 壬   |
| 怕  | 怕   | 怕   | 怕   |
| 開  | 開   | 開   | 開   |
| 庚  | 庚   | 庚   | 庚   |
| 六  | 六   | 六   | 六   |
| 辛  | 辛   | 辛   | 辛   |
| 殺  | 殺   | 殺   | 殺   |



(二) 第二組實驗是針對不同數量的資料集進行測試，我們首先擴充資料集的數量至 1000 筆，再分別使用 100、500、1000 筆資料進行實驗。

| 100張 | 500張 | 1000張 |
|------|------|-------|
| 煩 煩  | 雜 雜  | 禮 禮   |
| 山 山  | 甫 杜  | 彼 彼   |
| 一 一  | 殺 殺  | 勢 勢   |
| 寅 寅  | 杜 茶  | 冑 冑   |
| 白 白  | 白 遣  | 劇 劇   |
| 獅 獅  | 開 開  | 袖 袖   |
| 十 十  | 進 十  | 傳 傳   |
| 科 科  | 插 丑  | 乞 乞   |
| 禮 禮  | 十 處  | 澄 澄   |
| 進 進  | 丑 甫  | 敘 敘   |
|      | 處 心  | 髮 髮   |
|      | 空 空  | 梳 梳   |
|      | 心 白  | 儒 儒   |
|      | 瑯 瑯  | 擴 擴   |
|      | 茶 茶  | 渾 渾   |
|      | 差 差  | 午 午   |

## 柒、討論

### 一、實驗 VGG 模型生成手寫字體之成效

#### (一) 手寫字體生成：

1. 樣本粗細：由組別一、二可以發現，改變的程度並不高，我們推測是因為較粗的

字體和原始字體過於相近，導致損失函數過低，因而不會造成太大的改變。因此在生成個人化手寫字體時，不建議使用字體過粗的筆。

2. 樣本字體相似度：由組別三、四可以得到推論，字型結構和原始字體越相近者，此模型較能做出更多的改變，越能完成個人化的目標。
3. 樣本顏色(模型影響範圍)：此實驗可用來檢測模型的運作範圍，不同顏色字體有「顯影劑」的功能，我們可以發現在組別五中，字體的輪廓附近有許多黃色的色塊，這些色塊可以證明此模型有助於改變字體的輪廓，達到個人化的目標。
4. 樣本線條形狀：由組別六可清楚看到其成果之變化遠大於第一組的成果，因此我們可以推知：樣本字體越彎曲越有特色，越容易改變原始字體的特徵，越有助於達到個人化的目標。
5. 背景顏色：由第七組實驗可知：由於背景是黑色，照理說原始圖片背景應該全部都要變色，但卻不然，在接近輪廓的地方有些許白色，此結果也可證明此模型對字型輪廓的影響是實際存在的。
6. 小結：從各項實驗我們可以知道此模型的手寫字體：
  - (1) 樣本字體越細，越能增加字型改變幅度，越有助於生成個人化字體。
  - (2) 樣本字體和原始字體越相近，越能增加字型改變幅度，越有助於生成個人化字體。
  - (3) 樣本字體形狀越獨特，越能增加字型改變幅度，越有助於生成個人化字體。
  - (4) 此模型的功能在字體輪廓附近有實際的功效，確實有改變字體風格之潛力。
  - (5) 然而，儘管能在字體輪廓附近產生改變，改變的形狀結果並不理想。若能繼續往下發展，我們將把研究著重在優化模型改變字體的線條品質以及嘗試改變字體主架構上的 s 差異，希望能生成更逼真的手寫字體。
  - (6) 儘管生成字體之線條與品質並不如預期，但是部分實驗卻告訴了我們這套模型的功能是實際存在的，我們將吸收這些寶貴的經驗，進行下

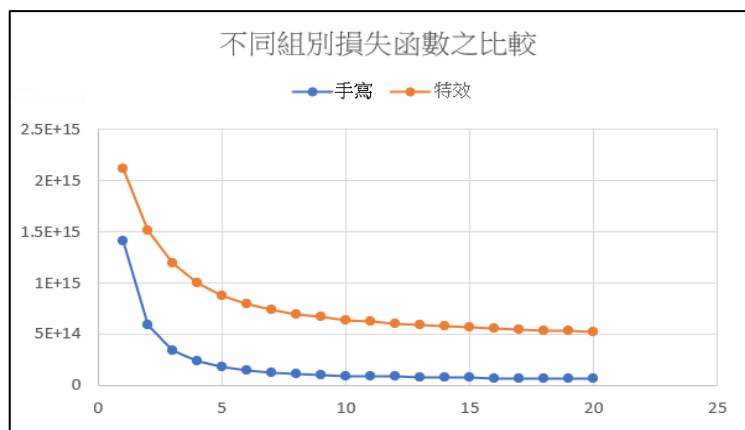
一階段的實驗。

## (二) 特效字體生成：

1. 由第八組和第十一組的火焰特效系列圖可以發現，成果圖的字型上有部分轉換成橘紅色，且存在不換火焰輪廓之風格，因此我們可發現特效圖對生成字體的色彩改變有卓越的成效。
2. 由第九組跟第十三組的閃電圖片可以明顯的發現：此模型不但能模擬樣本圖片的色彩變化，亦能模擬參考圖片的直線線條特徵，這兩組生成字體上的電弧可作為證明。
3. 由第十組和第十二組可以發現，此模型亦可以模擬樣本圖片中的曲線線條及弧線，從這兩組字體輪廓外圍的線條可以得到證明。
4. 小結：使用特效圖片生成藝術字體可說是十分成功，不論是色彩、輪廓都能大致模擬樣本圖案，字體也具相當程度的可辨認性。然而，由於背景也會參與訓練，導致背景也會出現變化而無法維持白底，因此在未來我們可能需要搭配程式進行去背，方可作為生成完整特效字體集的模式。

## (三) 損失函數趨勢

從下圖可以得知：此模型在訓練初期時就能快速降低損失函數的數值，而一次訓練約只需一分鐘的時間。訓練大約十次即可完成大部分的轉換，剩下的次數則是希望能讓成果圖片更臻完美。這是由於我們使用的優化算法—L-BFGS 算法，一個用於快速逼近函數極值且常出現在深度學習的算法。



▲圖七 VGG 模型損失函數趨勢圖

## 二、實驗 Bicycle-GAN 模型在生成手寫字體之成效

- (一) 如研究結果，我們可以明顯的發現：一個輸入能夠生成截然不同的數種成果，因為 Bicycle-GAN 模型有別於其他種生成模型，能夠支援多種風格的輸出。能達成這種成果最主要是受到 Bicycle-GAN 中 cVAE-GAN 結構的影響，因為此結構能夠加入雜訊向量的輸入，增加生成結果的多樣性。
- (二) 從研究結果中可看出，我們的輸出結果和我們所預期的手寫字體有相當的落差，我們推測問題是出在模型的訓練上，才會導致模型輸出有顏色的字體。
- (三) 由於輸出之結果和預估得到的字體的差異，我們推測應是損失函數的影響，因此可能需要對損失函數的定義進行微調，讓模型能夠更加貼近我們的需求。
- (四) 我們認為 Bicycle-GAN 模型在字體風格轉換上具有可行性，但是必須將目前模型的參數、設定上加以修改，才能順利運行並達到我們的期望的結果。

## 三、實驗 Zi2Zi 模型在生成手寫字體之成效

- (一) 根據我們的實驗結果可看出 Zi2Zi 模型輸出的結果，在少數訓練資料的情況下仍然能生成線條清晰的字體。
- (二) 從實驗中我們可以發現訓練次數越多，生成個人化中文字體的效果越顯著。
- (三) 從實驗中我們可以發現訓練集張數在 500 張時有最好的成效，我們推測資料集數量為 1000 張時，模型沒有辦法精確的擷取如此大量的字體特徵。因此，我們未來的發展方向就是期望能夠提升模型特徵擷取的精確度。
- (四) 在這次實驗中我們能清楚的看到生成結果有手寫字體風格，但仍可看到生成圖片中有些許雜訊，我們希望能透過增加訓練資料的數量，再進行模型訓練，達到更好的字體風格轉換功能。

## 四、未來展望

- (一) VGG 模型的線條精緻度：我們希望能夠優化字體特徵的擷取程式，讓模型能夠精確的改變字體架構，持續探究 VGG 模型生成手寫字體的成果細緻度。
- (二) VGG 模型特效字體：未來我們希望能撰寫程式解決特效字體背景的問題，讓 VGG 模型生成特效字體的成效更顯著。
- (三) Bicycle-GAN 的字體架構：我們希望能夠調整 Bicycle-GAN 的損失函數集部分參數，

希望此模型能夠在維持優點的情況下提升個人化字體生成的品質。

(四) 提升 Zi2Zi 模型特徵擷取的精確度：我們希望能夠透過調整模型來提升模型特徵擷取的精確度，以提升生成字體的品質、生成更加細緻的個人化中文字體。

(五) Zi2Zi 模型的細部優化：我們希望能透過擴充資料集和調整部分參數的方法消除輸出成果中的雜訊，以達到更完整的字體轉換成效。

(六) 向量圖可行性：目前大部分模型都使用點陣圖，未來我們想嘗試實驗向量圖在生成手寫字體上之可能性。

(七) 我們希望能夠找到一種能將字體風格量化的評估方法，而不是單純用肉眼判斷字體的相似度，才能客觀的探討實驗成效。

## 捌、結論

一、根據 VGG 模型生成手寫字體之實驗，我們發現 VGG 模型在中文字輪廓附近具有修飾的功效，確實具有細微調整字型的功能。

二、根據 VGG 模型生成手寫字體之實驗，特效字體能夠完整模擬樣本圖的線條、色彩，完整呈現具藝術線之字體風格，只需進行後續圖片處理就能作為可使用的完整字體。

三、根據 Bicycle-GAN 生成手寫字體之實驗，輸出成果能夠表現出 Bicycle-GAN 的模型特色，惟須調整模型參數，使其能夠生成更好的手寫中文字體。

四、根據 Zi2Zi 生成手寫字體之實驗，Zi2Zi 模型具有優秀的個人化字體生成能力，若能更精確的擷取筆畫細部的特徵，本模型將更臻完美。

五、輸出結果經過我們設計の後處理程式運算後可做為製作一套電腦字體的實際素材，實現了本研究的目的——產生一套個人化手寫字體。

## 玖、參考文獻資料

### 一、參考資料

(一) Leon A. Gatys, Alexander S. Ecker, Matthias Bethge (2015), A Neural Algorithm of Artistic Style。

(二) Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, Eli Shechtman(2017), Toward Multimodal Image-to-Image Translation。

(三) Bo Chang, Qiong Zhang, Shenyi Pan, Lili Meng(2018) , Generating Handwritten Chinese Characters using CycleGAN 。

(四) 柯博文(2020) 。《Python+TensorFlow 2.x 人工智慧、機器學習、大數據 | 超炫專案完全實戰》。台北：碁峯出版社。

## 【評語】 052504

1. 本作品以利用三種深度學習模型，實驗不同的個人化字體產生結果。
2. 雖然作品說明書有提供解說結果用的少數範例，建議還是要有完整的實驗規劃及實驗結果，用以說明或比較不同模型的優劣差異。另外應建立客觀的字體風格相似度（或定義評量 metrics），而不是以肉眼判斷好壞。
3. 建議進一步闡述作品的實用性，例如若用個人化字體所產生文件寄給並無該字體的其他人，似乎也是會以其他字體取代，如此一來似乎也只有自己可以使用新字體。

## 作品簡報



# 字由字在—

## 利用深度學習生成個人化字體

組別：高級中等學校組 科別：電腦與資訊科 作品編號：052504

# 研究動機

- 生活中充斥著制式化的電腦字體 → 有溫度的**個人化**手寫字體
- 設計一套電腦字體耗時又費力 → 利用深度神經網絡**節省時間、人力**
- 期望能設計出兼具**字體獨特性**和**製作便利性**的字體生成模型

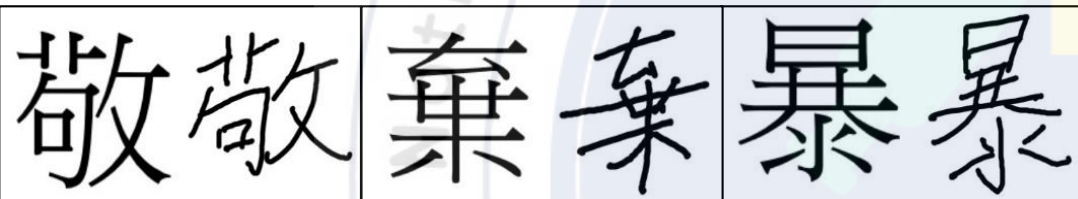
- 以**VGG-19**作為深度卷積網路的框架，透過風格矩陣生成個人化字體。
- 利用複合式結構——**Bicycle-GAN**作為模型生成個人化字體。
- 嘗試利用**Zi2Zi**模型作為條件式GAN的模型，生成個人化中文字體。
- 比較各方法的成果並分析成效，分析各種方法的優劣。
- 對生成的個人化字體進行**後處理**，使其符合做為電腦字體的樣式。

# 研究目的

|      | VGG-19模型 | Bicycle-GAN模型    | Zi2Zi模型       |
|------|----------|------------------|---------------|
| 模型原理 | 卷積神經網路   | CLR-GAN、CVAE-GAN | Pix2pix2、CGAN |
| 模型特色 | 風格矩陣     | 一對多圖形轉譯          | 針對文字架構設計      |

# 訓練集建置

在實驗中我們使用的是由**電腦字體(新細明體)**和**手寫字體**共同組成的成對(paired)資料集。電腦字體可從內建字體檔案(truetype檔)中提取，而手寫字體則由手寫板直接書寫。我們製作了超過1000組的訓練資料，並輸入各個模型進行訓練。

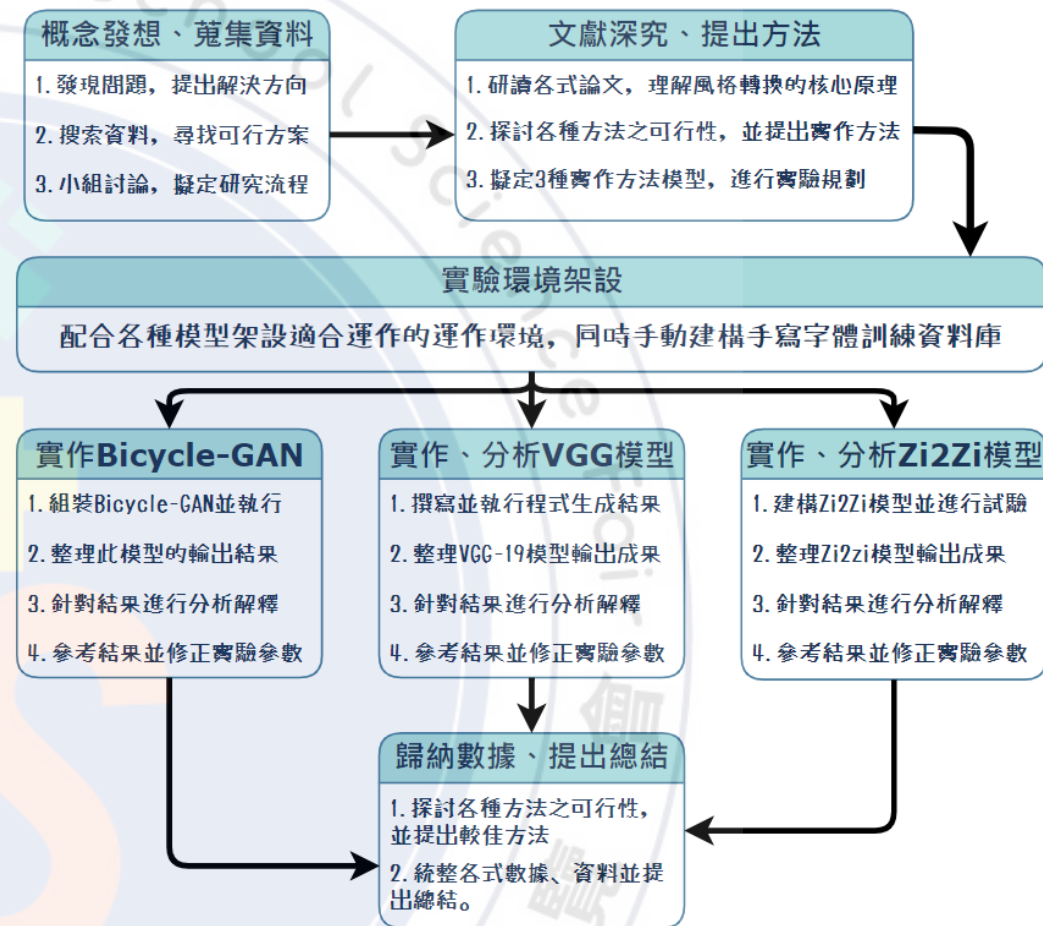


▲訓練用成對訓練集示意圖

# 圖片後處理

為了讓輸出成果能**實際應用在電腦字體**中，必須對輸出成果進行後處理。我們使用Python的Pillow套件批次進行**像素解析**，並對將「白色」和「接近白色」的區塊挖除。最後再把字體和中文字的**編碼**結合，即可製作成個人化字體檔案。

# 研究流程圖



▲輸出圖片後處理前後比較圖

# VGG 模型

在此模型中我們會將雜訊向量根據2種輸入字體進行特徵擷取並迭代，期望能產出：具有「**原始字體輪廓**」和「**樣本字體風格**」的成果字體。每一次的優化可分為2個部分：

(一)**內容損失函數**：將目前的成果和原始字體進行圖片像素分析，並計算出「內容損失函數」，代表圖片內容的宏觀差異。

$$L_{content}(C, G) = \frac{1}{2} \sum_{i,j} (a_{ij}^{[l](C)} - a_{ij}^{[l](G)})^2$$

(二)**風格損失函數**：利用filter對目前成果和樣本圖片進行特徵擷取，產生「風格矩陣」並計算出風格損失函數。

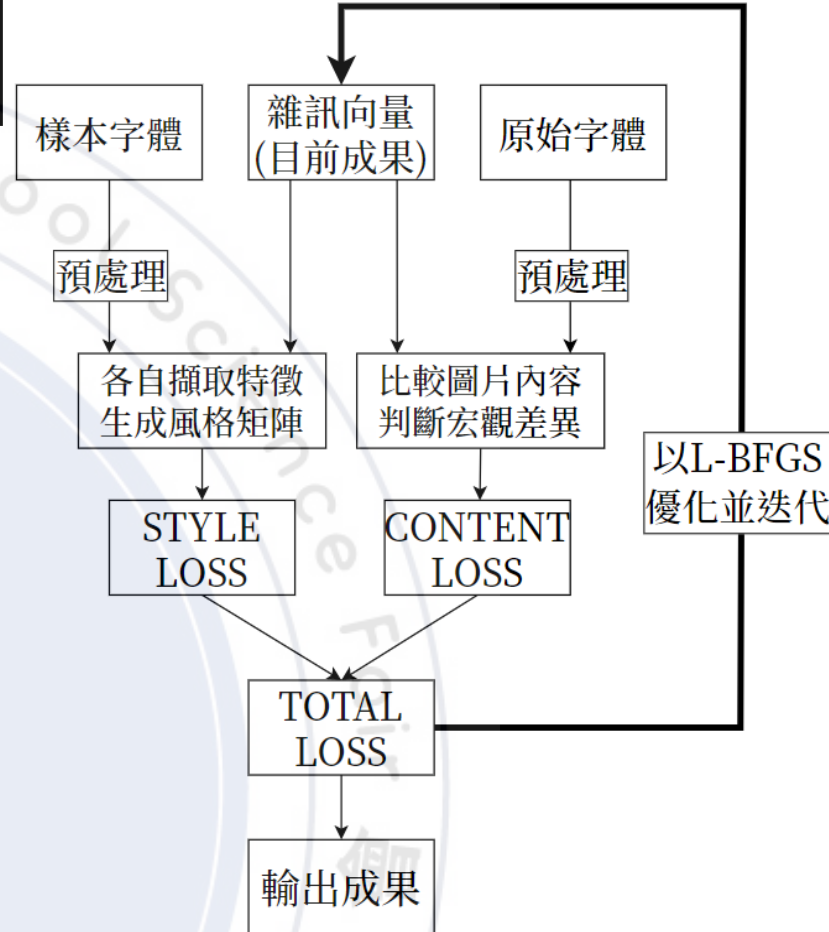
$$G = A^T A = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_n^T \end{bmatrix} [a_1 \ a_2 \ \dots \ a_n] = \begin{bmatrix} a_1^T a_1 & a_1^T a_2 & \dots & a_1^T a_n \\ a_2^T a_1 & a_2^T a_2 & \dots & a_2^T a_n \\ \vdots & \vdots & \ddots & \vdots \\ a_n^T a_1 & a_n^T a_2 & \dots & a_n^T a_n \end{bmatrix}$$

$$L_{style}^{[l]}(S, G) = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^{[l](S)} - G_{ij}^{[l](G)})^2$$

計算完損失函數之後，我們使用L-BFGS算法進行模型的優化，不斷逼近最符合需求的輸出成果。

## 實驗設計：

1. 手寫組：利用**手寫字體**作為樣本字體進行實驗，操作變因有：字體粗細、結構、顏色、彎曲程度等等
2. 特效組：利用各式**特效圖片**作為樣本字體進行實驗，探討模型對色彩的風格轉換成效。







# VGG模型實驗討論

## 變因探討

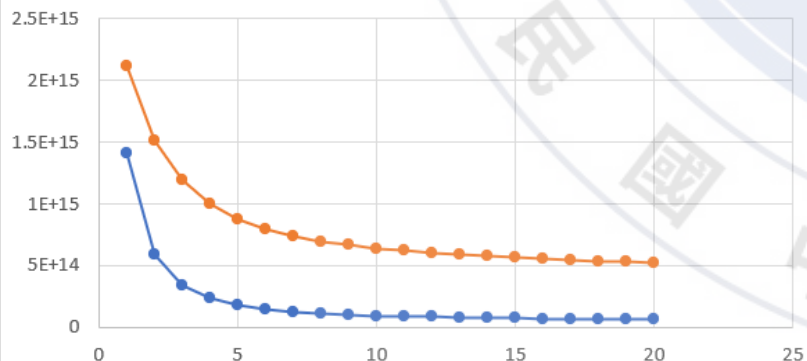
從**手寫組**實驗我們可以發現：樣本字體越細、越彎曲、有特色或是原始字體和樣本字體的架構越相近，會有越好的字體轉換成果。

從**特效組**的實驗我們可以發現：此模型較擅長抓取顏色特徵鮮明的風格，故特效組的輸出成果較佳，風格重現較完整、細膩。

| 原始字體 | 特效圖片   | 輸出成果   | 原始字體 | 樣本字體   | 輸出成果   |
|------|--|--|------|--|--|
| 永    |    |    | 永    |   |   |
| 永    |   |   | 永    |   |   |
| 永    |   |   | 談    |   |   |
| 永    |   |   | 詠    |   |   |
| 永    |   |   | 永    |   |   |
| 永    |  |  | 永    |  |  |

不同組別損失函數之比較

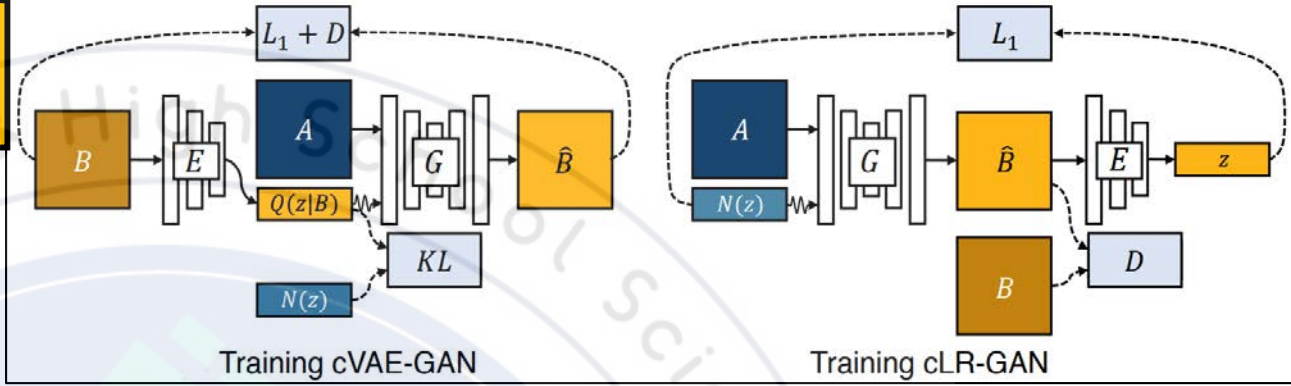
● 手寫 ● 特效



左圖是手寫組和特效組實驗過程的**損失函數趨勢圖**，從圖中我們可以發現：模型在**初期**的優化就有顯著的成效，能夠快速降低損失函數。而特效字體和手寫字體的差距則是因為圖片**內容差距**很大，是無法進行優化的部分。

# Bicycle-GAN 模型

Bicycle-GAN能夠支援多樣風格的生成，是由編、解碼器和GAN等結構所組成的複合式模型。主要分為2個部分。如右圖所示。



(一)**cVAE-GAN**：在VAE生成模型的基礎上添加條件輸入，在多樣風格的優點上生成具條件的內容。

(二)**cLR-GAN**：將生成器生成出來的圖片輸入到Encoder網絡內，期望能讓最後輸出的圖片更加接近原本輸入的圖片，有助於提升生成圖片的真實性。

## 實驗設計

在此模型的實驗中我們使用先前建立的**成對(paired)資料集**，隨機挑選樣本並使用架設好的Bicycle-GAN模型進行訓練並**輸出10個結果**。表格中的原始圖片就是上述Bicycle-GAN模型中的條件圖片。

| 原始圖片 | 輸出圖片 |
|------|------|
|      |      |
|      |      |
|      |      |

# Bicycle-GAN 模型實驗討論

## 風格多樣性

從實驗結果我們可以發現：一個輸入能夠生成數種**不同風格**的成果，主要是因為Bicycle-GAN模型的**結構特色**，能夠支援多種風格的輸出。

不過依照實驗成果「多樣風格」主要作用在**色塊變化**上，若要改變**字形的架構**，可能需要調整參數調整高斯噪聲的作用。

## 未來調整方向

- 損失函數計算：我們希望能透過改變損失函數的定義來對模型輸出進行調整。
- 高斯噪聲計算：我們希望能將高斯噪聲作用的對象從「圖片色塊」改為「線條架構」。

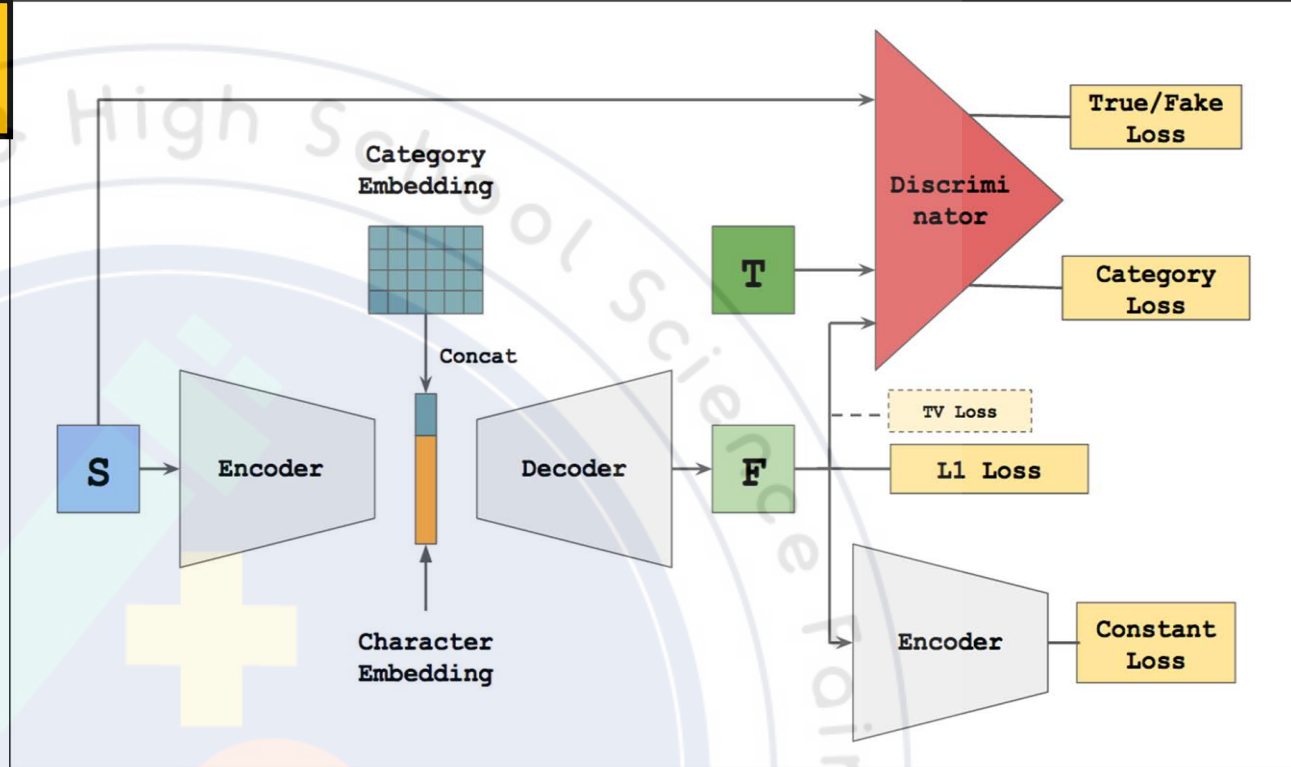


# Zi2Zi 模型

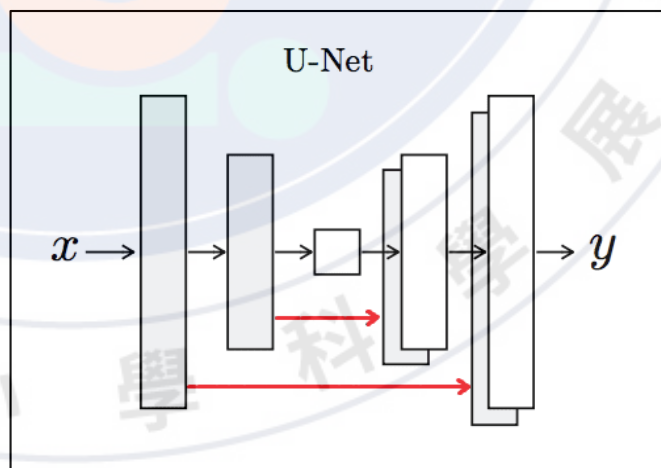
Zi2Zi模型是一種結合了**AutoEncoder**和**GAN**的模型，能進行字體間的風格轉換。Zi2Zi參考了pix2pix和Conditional GAN的原理，並加入一些結構，提升字體轉換的效果。

(一)**類別嵌入法**(Category Embedding)：在編碼器生成的隱編碼上嵌入未經訓練的**高斯噪聲**，再輸入解碼器生成圖片，進而實現模型對多種風格字體的相容性。

(三)**U-Net結構**：由於在編、解碼器網絡中，輸入和輸出具有部分**相同的資訊**，因此需要一些**跳躍性的連結**，使資訊能夠直接地通過網絡。U-Net結構就是將第*i*層和第*n-i*層直接連接起來，確保輸入圖片和生成圖片的相似性。



(二) **Patch GAN**：為了讓**鑑別器**更精確的解析字體，鑑別器會對圖片**分區進行鑑別**，形成一個**矩陣**，並依據各部分的評比平均進行判斷，提高辨識字體的準確性。



U-Net結構示意圖



# Zi2Zi模型的損失函數

- **GAN**：利用**生成器和鑑別器互相對抗**並優化模型，是對抗式生成網絡的核心概念。
- **L1 Loss**：是用來計算輸出和輸入圖片**差距**的損失函數，有助於生成構造較簡單部分的圖片。
- **Category Loss**：在能輸出多種風格的模型中，會發生「**風格混淆**」的情況。故有此損失函數來避免此現象發生。
- **Constant Loss**：將輸出圖片再次輸入Encoder，並和輸進Decoder前的隱編碼比對，計算損失函數。由於輸入和輸出是同一字，因此2種**隱編碼應要十分相似**。故此損失函數有助於**提升模型收斂速度**。
- **TV Loss**：此損失函數主要功能為**降噪**，能夠降低相鄰像素值的差異，讓圖形邊界更柔和。

# Zi2Zi模型的實驗設計

Zi2Zi模型的實驗主要分為2組，操作變因為**資料集筆數**和**訓練次數**。另外，在這系列的實驗中，90%的資料集會做為訓練資料、10%的資料集會作為測試資料，和輸出成果進行比對、評估。

1. 第一組實驗使用了500筆訓練資料，操作變因為訓練次數。
2. 第二組實驗訓練了50次，操作變因是資料集數量。

# Zi2Zi模型的實驗結果

- 根據實驗結果可看出Zi2Zi模型在**少數訓練資料**的情況下仍然能生成線條清晰的字體。儘管細節部分並非完美，仍可清楚的辨識字體並分辨出風格的差異
- 在這次實驗中能清楚看到生成結果有**手寫字體風格**，但仍存在些許雜訊，我們希望能提升**特徵擷取的精確度**，達到更好的字體風格轉換功能。
- 未來我們希望能在不影響品質的情況下**減少所需的訓練集**，以提升生成個人化字體的便利性和實用性。

| 5號 | 30號 | 45號 | 50號 |
|----|-----|-----|-----|
| 茶  | 茶   | 茶   | 茶   |
| 戊  | 戊   | 戊   | 戊   |
| 疾  | 疾   | 疾   | 疾   |
| 差  | 差   | 差   | 差   |
| 午  | 午   | 午   | 午   |
| 瑯  | 瑯   | 瑯   | 瑯   |
| 空  | 空   | 空   | 空   |
| 彼  | 彼   | 彼   | 彼   |
| 己  | 己   | 己   | 己   |
| 壬  | 壬   | 壬   | 壬   |
| 怕  | 怕   | 怕   | 怕   |
| 開  | 開   | 開   | 開   |
| 庚  | 庚   | 庚   | 庚   |
| 六  | 六   | 六   | 六   |
| 辛  | 辛   | 辛   | 辛   |
| 殺  | 殺   | 殺   | 殺   |

▲實驗一之輸出結果

| 100張 | 500張 | 1000張 |
|------|------|-------|
| 煩    | 雜    | 禮     |
| 山    | 甫    | 彼     |
| 一    | 殺    | 黎     |
| 寅    | 杜    | 暴     |
| 白    | 白    | 劇     |
| 獅    | 開    | 袖     |
| 十    | 進    | 馬     |
| 科    | 插    | 怵     |
| 禮    | 十    | 澄     |
| 進    | 丑    | 遠     |
|      | 處    | 遠     |
|      | 甫    | 遠     |
|      | 心    | 豎     |
|      | 空    | 豎     |
|      | 心    | 極     |
|      | 瑯    | 儒     |
|      | 茶    | 擴     |
|      | 差    | 渾     |
|      |      | 午     |
|      |      | 午     |

▲實驗二之輸出結果

# 結 論

1. 根據VGG模型生成**手寫字體**之實驗，我們發現VGG模型在中文字輪廓附近有**修飾**的功效，確實具有細微調整字型的功能。
2. 根據VGG模型生成**特效字體**之實驗，特效字體能夠完整模擬樣本圖的線條、色彩，完整呈現具**藝術性之字體風格**。
3. 根據Bicycle-GAN生成手寫字體之實驗，輸出成果能夠表現出Bicycle-GAN**多樣化風格**的模型特色，惟須調整模型參數，使其能改變高斯噪聲的作用方式，生成更細緻的中文手寫字體。
4. 根據Zi2Zi生成手寫字體之實驗，Zi2Zi模型具有**優秀的個人化字體生成能力**，若能更精確的**擷取筆畫細部的特徵**，本模型將更臻完美。
5. 輸出結果經過我們設計的**後處理程式**運算後可做為製作一套**電腦字體**的實際素材，實現了本研究的目標——產生一套**個人化手寫字體**。

# 未來展望

- VGG模型的線條精緻度：我們希望能夠優化字體**特徵的擷取程式**，讓模型能夠精確的改變**字體架構**，持續探究VGG模型生成手寫字體的成果細緻度。
- Bicycle-GAN的字體架構：我們希望能夠調整Bicycle-GAN的**損失函數及部分參數**，希望此模型能夠在維持優點的情況下提升個人化字體生成的品質。
- Zi2Zi模型的細部優化：我們希望能透過**擴充資料集**和調整部分參數的方法提升**特徵擷取的精確度**，並消除輸出成果中的**雜訊**，以達到更完整的字體轉換成效。
- 向量圖可行性：目前大部分模型都使用**點陣圖**，我們想嘗試實驗**向量圖**在生成手寫字體上之可能性，並將其和點陣圖的成果做比較。
- 資料集多元性：我們的資料集主要以新細明體作為電腦字體，我們希望能使用其他字體進行實驗，如：**標楷體**，並比較利用不同字體做為資料集的差異，找出成效最好的方法。

# 參考資料

- Leon A. Gatys, Alexander S. Ecker, Matthias Bethge (2015), A Neural Algorithm of Artistic Style。
- Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, Eli Shechtman(2017), Toward Multimodal Image-to-Image Translation。
- Bo Chang, Qiong Zhang, Shenyi Pan, Lili Meng(2018), Generating Handwritten Chinese Characters using CycleGAN。
- 柯博文(2020)。《Python+TensorFlow 2.x人工智慧、機器學習、大數據 | 超炫專案完全實戰》。台北：碁峯出版社。