

# 草稿大師

<sup>1</sup> 蔡昕翰, <sup>1</sup> 葉安之, <sup>1</sup> 周楷蓉, <sup>1\*</sup> 李柏翰

<sup>1</sup>國立台灣師範大學附屬高級中學

通訊作者

E-mail: leepohan@gmail.com

## 摘要

解決不知如何下筆的困擾，就用草稿大師！作品結合了物聯網以及 AI。使用時選擇想畫的照片上傳到 app 伺服器運算，會產生線稿並顯示於手機螢幕。本作品解決不知從何畫起的煩惱，也可隨著使用者的喜好轉換圖片風格是一個融入於休閒娛樂的作品。

**關鍵字:** Fast Style Transfer, AnimeGANv2, YOLOv5, grabcut

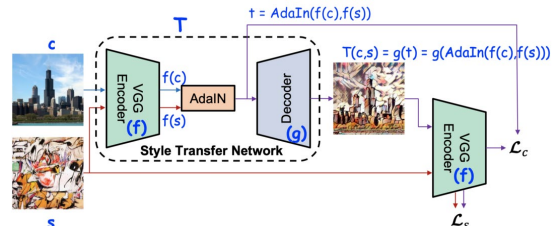


圖 2 Fast Style Transfer 效果圖(資料來源：  
<https://towardsdatascience.com/fast-and-arbitrary-style-transfer-40e29d308dd3>)

這個模型是一個 Encoder-Decoder 模型，Encoder 是一個預訓練好的 VGG-19 模型的前一部分，之後經由 AdaIN 將已經 encoded 過的內容與風格圖片融合起來變成他的特徵 map，最後透過 Decoder 將特徵 map 變成一張圖片。要訓練 Decoder 會將輸出的圖片與風格圖片經過 Encoder 再比對算出 loss。

我們的另一個風格轉換方法是 AnimeGANv2，使用時只需要輸入內容圖片就好了。他有三種模型，會讓輸入的圖片變成動漫效果。其效果及模型架構如圖 3。



圖 3 AnimeGANv2(資料來源：  
<https://tachibanayoshino.github.io/AnimeGANv2/>)

## 1. 前言

繪畫這工作不是每個人都擅長的(如圖 1)。

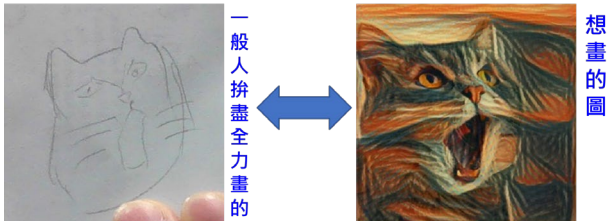


圖 1 構思草稿對一般人來講有些困難

為了解決繪畫不知如何下筆的困擾，AI 的機器學習目前非常熱門，且有一些演算法非常方便，於是在此氛圍下，我們構想了利用 AI 的機器學習方法，設計了一款以演算法方法或 grabcut 結合 YOLOv5 為藍圖的「草稿大師」，希望能幫助初學者利用 AI 演算法來找出快速學習途徑，學習如何加入使用者的情境，來建構出充滿情感的素描。

## 2. 相關技術

## 2.1. 風格轉換

我們的風格轉換方法，主要是當使用者想要畫得更有趣時用的。我們用了兩種 AI 深度學習模型達成，分別為 Fast Style Transfer [1]與 AnimeGANv2 [2]。Fast Style Transfer 需要輸入兩張圖片，分別為內容圖片(c)與風格圖片(s)，效果及模型架構如圖 2。

## 2.2. 計算輪廓圖

我們使用 Canny 演算法計算圖的邊緣，但在進行 Canny 運算前，我們會先將圖片轉換為灰度。我們拿圖 1 來做示範，效果如圖 4。

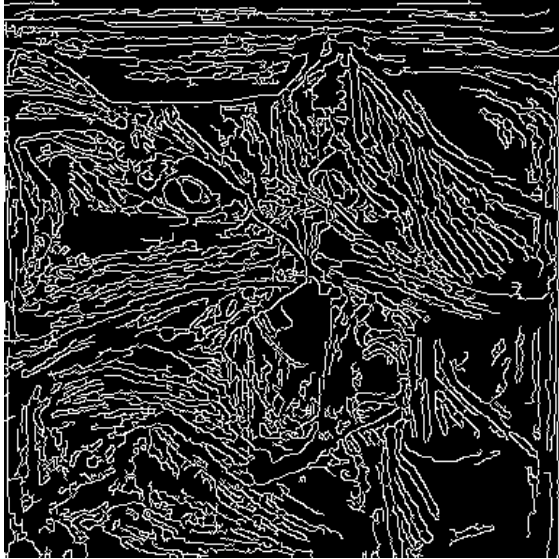


圖 4 效果示意圖

可以看到這張圖的筆劃太過複雜，因此要先將圖片的某些較不重要的邊緣柔和化，我們用高斯模糊達目的，如圖 5。



圖 5 效果示意圖

我們使用 OpenCV 進行計算輪廓。

### 2.3. 分隔輪廓圖

我們在製作完輪廓圖後會將其分成筆畫我們的純演算方法流程圖如圖 6。

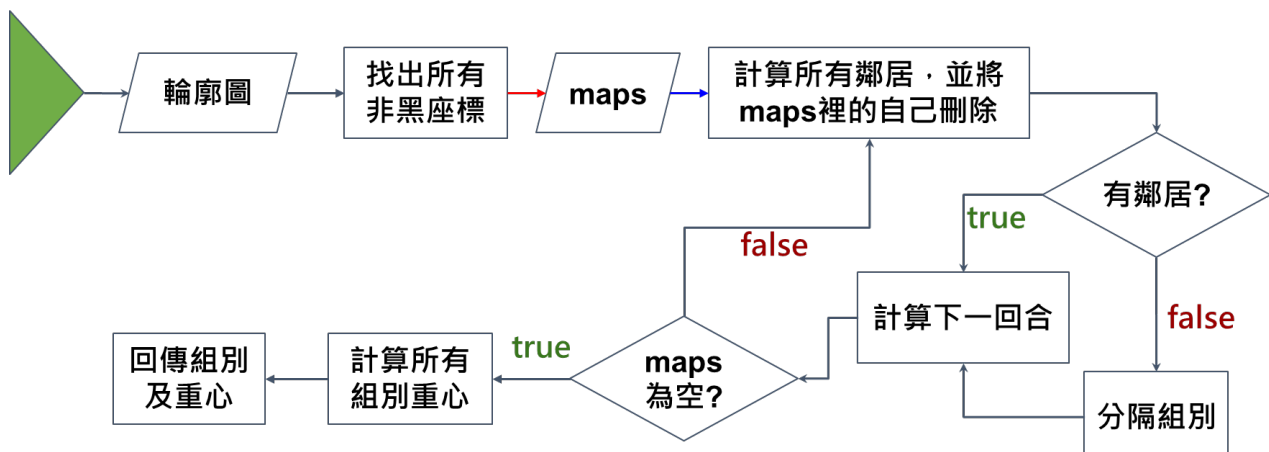


圖 6 計算筆畫順序純演算方法流程圖

簡單來說，若有一個輪廓圖如圖 7。

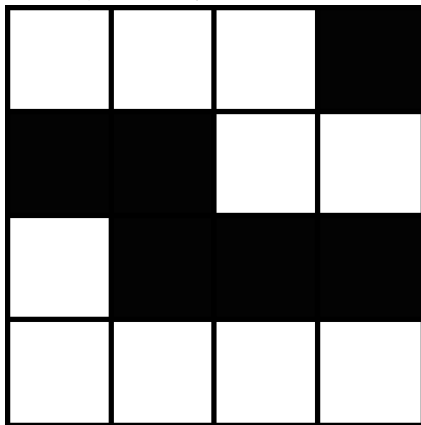


圖 7 示範輪廓圖

演算法會先找到所有白色空格，然後記起來，將選到的格子找到其鄰居。假設圖 8 中的紅格子是被選到的，藍格子是他的鄰居。

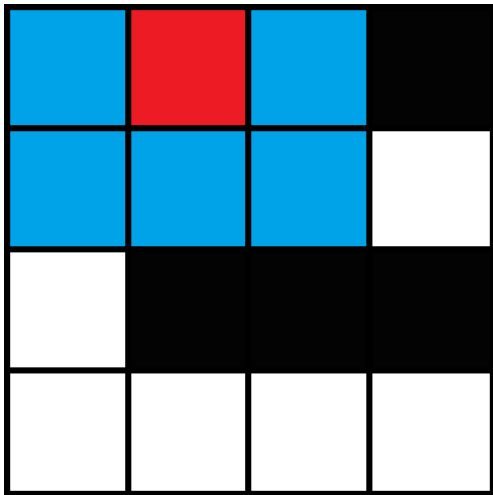


圖 8 演算法示意圖

但其中兩個是黑的，所以下一回會將那兩個排除後選中，並將原本選中的加入組別，之後的圖就直接省略黑色鄰居。如圖 9，綠色為已加入組別的，紅色的為選中的，藍色的為鄰居。

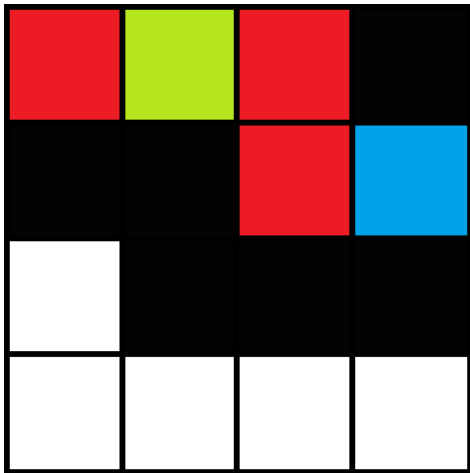


圖 9 演算法示意圖

之後重複上步驟。

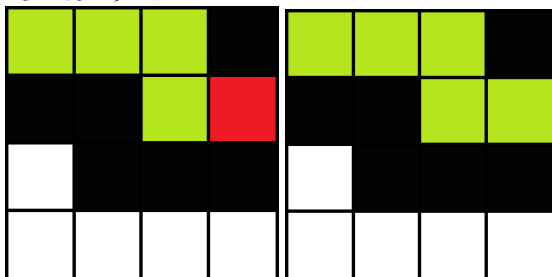


圖 10 第一組完成

最後當沒有選中的格子時，就完成了一個組別。接著就來看看還有沒有剩餘的白色格子，若有，則選定其中一個重複剛剛的步驟，這次使用紫色作為組別。

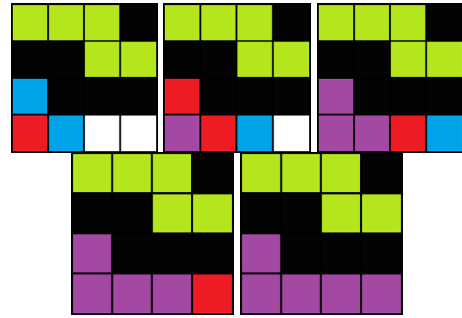


圖 11 全部完成

到這裡就沒有白色的空格了，所以我們完成了分組。但可以看到在流程圖的最後有一個計算每組重心，其用途會在之後的計算筆畫順序演算法中。

## 2.4. 筆畫順序計算方法

我們的筆畫順序演算方法有分兩種，第一種是純演算方法，第二種是用 YOLOv5 結合 grabcut。其中第一種包含四種演算法：

1. upper  
此演算法就是以一條線上最高的點做由高到低排序，排越前面的越先畫。  
畫出來的效果就像是從上面畫到下面，如圖 12。
2. nearest\_neighbor  
此演算法會先選擇最上面的線，之後選擇重心離上一個選擇的重心最近的線，重複直到全部畫完。  
畫出來的效果就是筆需要移動的比較少，如圖 13。
3. start\_from\_g  
此演算法會將所有線的重心的重心計算出來，並依照每條線的重心到那點的距離由小到大排列每條線的順序。  
畫出來的效果像從中間畫到外面，如圖 14。
4. far\_from\_g  
此演算法跟剛剛的很像，會將所有線的重心的重心計算出來，並依照每條線的重心到那點的距離由大到小排列每條線的順序。  
畫出來的效果會像是外面畫到中間如圖 15。

由圖 12 到圖 15 可觀察到，他們畫出來的筆畫順序較不符合人性，以人的角度來講就是一個太過刻意想要達成某件目標了，不自然。人類通常會將一張圖內的重點先畫出來，再畫背景。

因此我們就有了第二種計算筆畫順序方法，使用 YOLOv5 先將物件偵測出來，再將偵測出來的矩形部分透過 grabcut 將非背景部分取出，其效果如圖 16，運作圖解如圖 17。

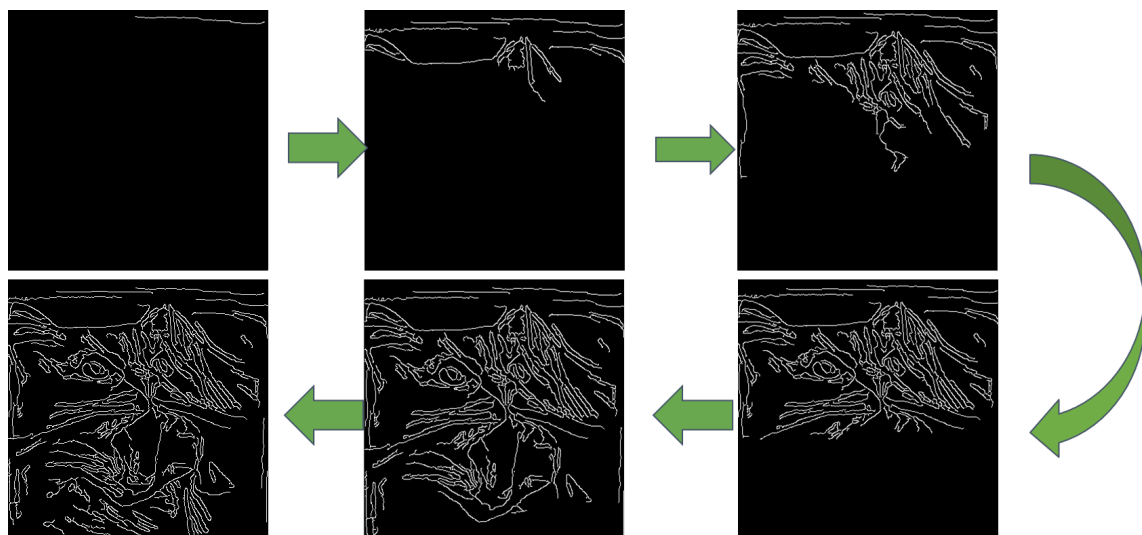


圖 12 upper 演算法示意圖

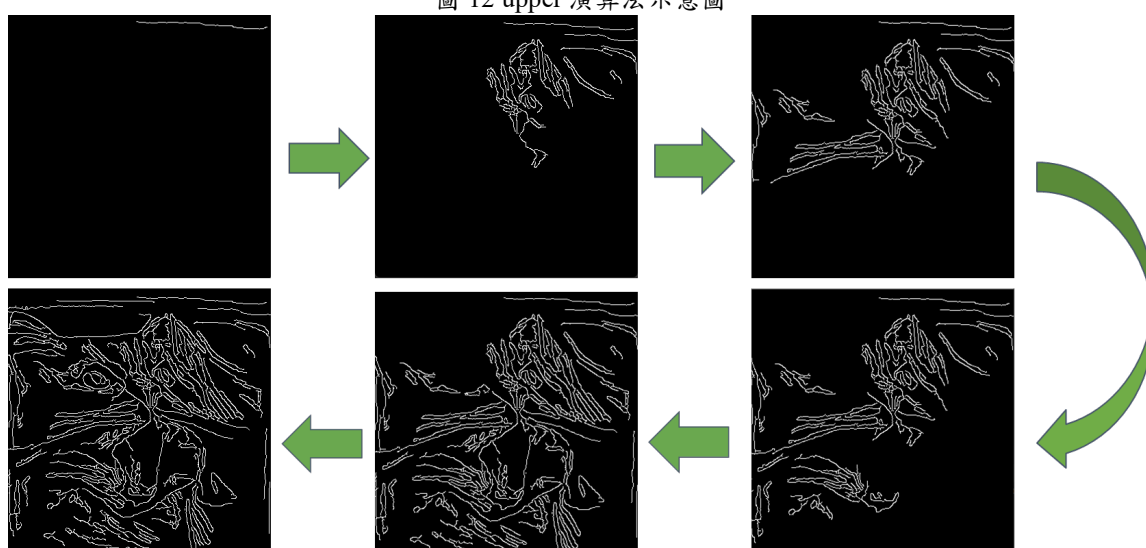


圖 13 nearest neighbor 演算法示意圖

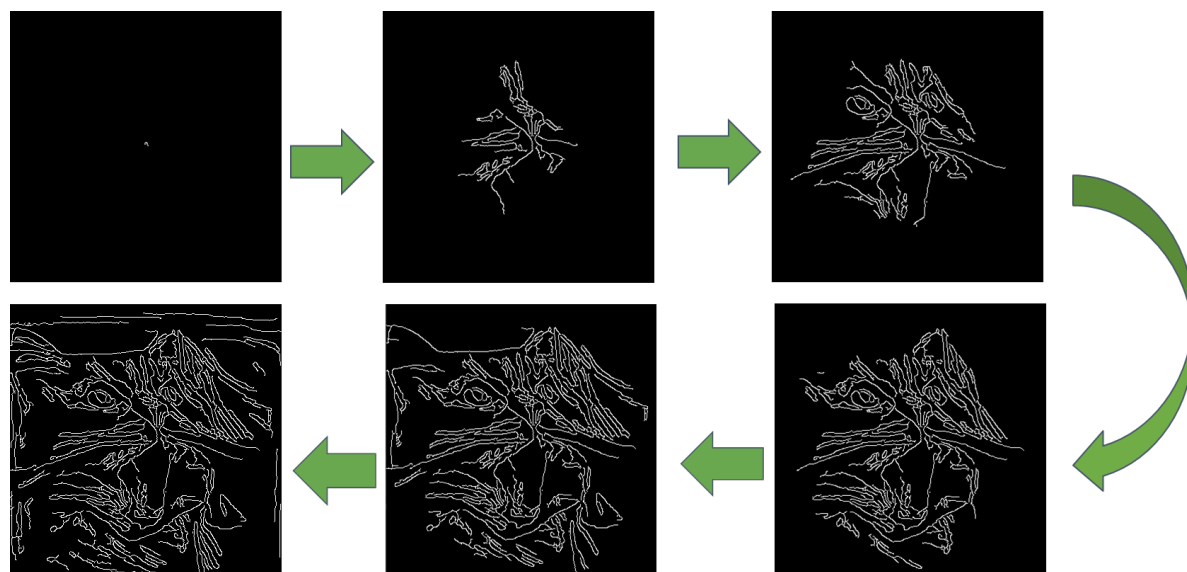


圖 14 start from g 演算法示意圖



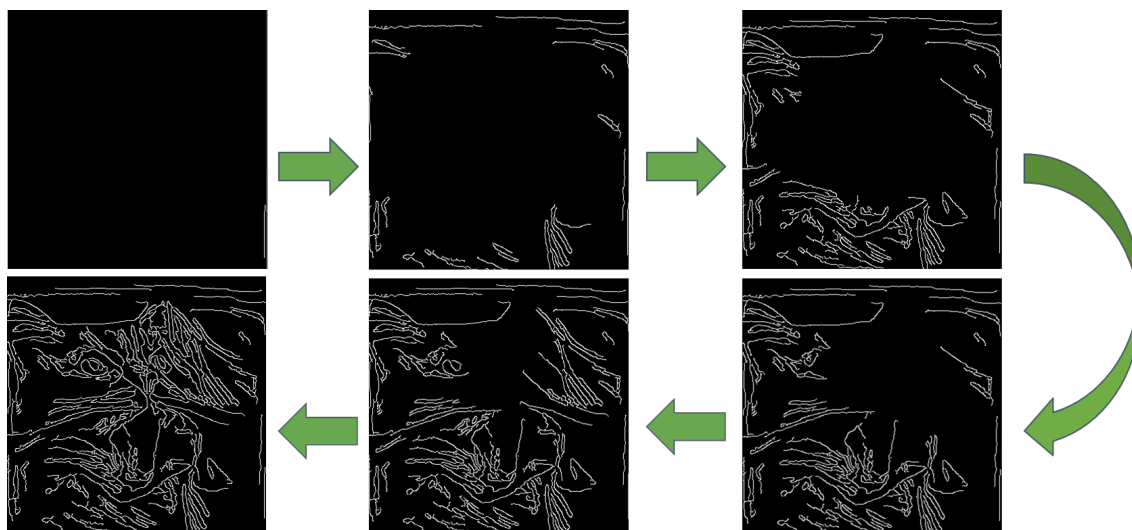


圖 15 far from g 演算法示意圖

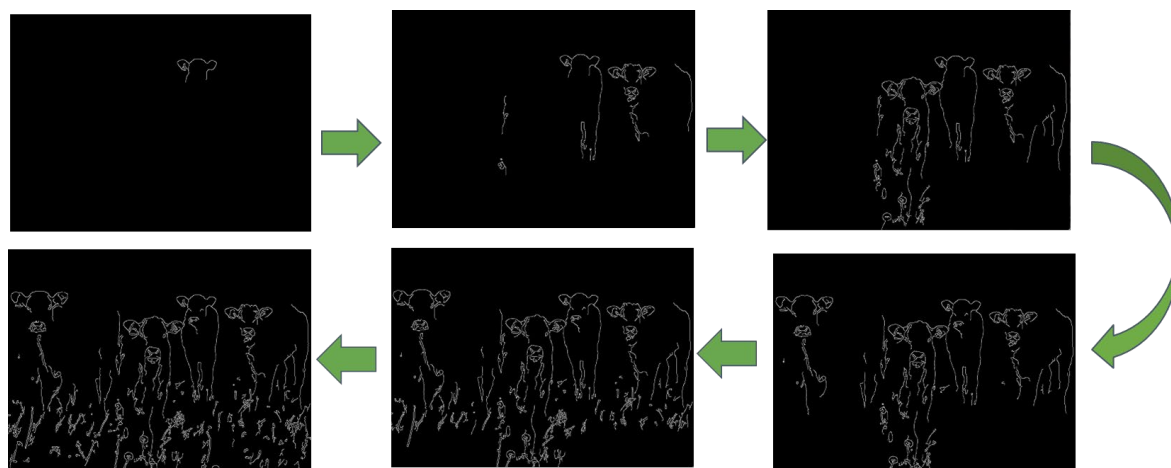


圖 16 YOLOv5+grabcut 演算法示意圖(圖中為四頭牛)

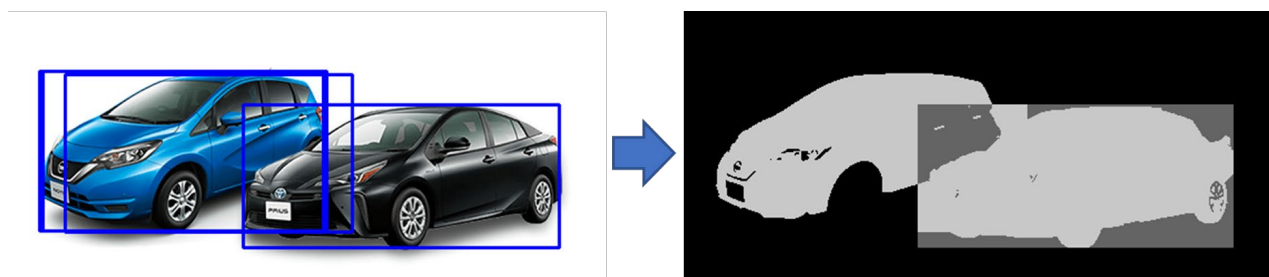


圖 17 先將圖片用 YOLOv5 擷取物件所在矩形，再進行 grabcut 找出有意義處，右圖中越亮越先畫。

### 3. 系統操作

#### 3.1. 系統架構圖

圖 18 為本作品整體系統架構圖，可分為伺服器端與用戶端。伺服器端主要做的事就是複雜的運算，用戶端主要就是給使用者操作的地方。首先會讓用戶選擇要畫哪張照片，此操作可以為照相或是選擇本機照片，應用程式會將使用者選擇的照片以 base64 形式存在手機上。之後會問使用者需不需要進行轉

換風格，若要，則依使用者選擇的轉換方法，要求使用者提供風格圖片。最後給使用者確認後就能計算筆畫了。計算筆畫的方法會給使用者選擇。選擇的計算筆畫順序的方法若為 YOLOv5+grabcut 方法，則會傳輸原圖+轉換過的圖給伺服器，若為純演算法，則只會傳輸原圖給伺服器。最後用戶端接收筆畫的順序並儲存起來，就讓使用者開始畫畫。

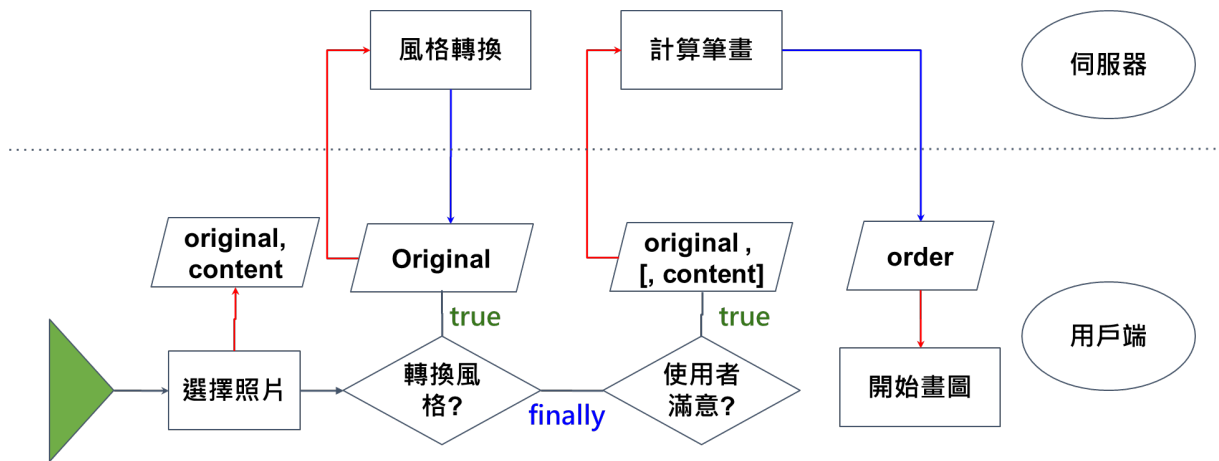


圖 18 系統架構圖

### 3.2. 伺服器實作

當伺服器啟動時，他會有如表 1 的 route。我們的用戶端與伺服器溝通時，就會呼叫表中的 route。在本作品中，我們會用 flask 實作伺服器。

表 1: 伺服器 route 說明

route	method	description
/	GET	檢查是否正在運作
/get_outline	POST	計算色塊邊框
/style_transfer	POST	轉換圖片風格
/gen_step	POST	計算畫筆順序

### 3.3. 用戶端操作

當用戶端啟動時，我們會讓使用者輸入伺服器的網址。操作畫面如圖 19。



Drawing Teacher  
A project by HSNU students

圖 19 啟動畫面

之後會進到照相的畫面，可以選擇要拍照或是選擇照片。操作畫面如圖 20。



圖 20 選擇照片

選完照片後會問使用者要不要轉換風格，若選擇 FST(Fast Style Transfer for arbitrary types)則需要選擇檔案上傳。操作畫面如圖 21。

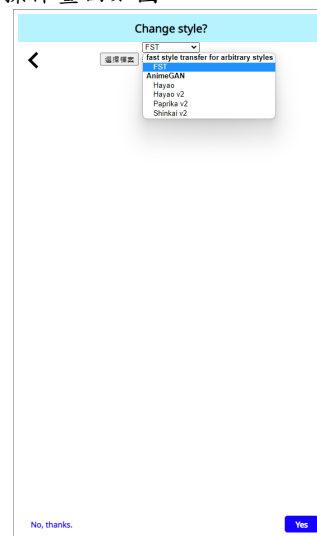


圖 21 詢問轉換風格

接著會將運算完的圖片回傳給用戶端，並問使用者好不好，操作畫面如圖 22。



圖 22 滿意畫面

接著會問使用者要用哪種筆畫順序演算法。操作畫面如圖 23。



圖 23 選擇筆畫順序演算法

最後就可以開始畫圖。我們有兩種用法，第一種是直接畫在螢幕上的方法，就只要用手指按住或是滑鼠按住就能畫了。第二種方法是連到攝影機，對著空白的紙，每一步跟著描，讓使用者能邊畫邊比對有沒有畫錯，並有一個實體成果。

畫面上有一些可調的東西。左上角那兩個滑桿可以調整畫布離頂部的高度(margin top)還有畫布的大小(magnification)。若在螢幕上畫覺得攝影機很煩可以

按 camera on 關掉攝影機。如果覺得畫太爛了可以按 clear 清空剛剛畫的。若想要給別人看看自己的大作，可以按 download 下載你剛剛畫的。如圖 24。



圖 24 沒調過的畫面

如圖 25，藍色的是使用者畫的，黑色是系統畫過的，紅色是準備要畫的。

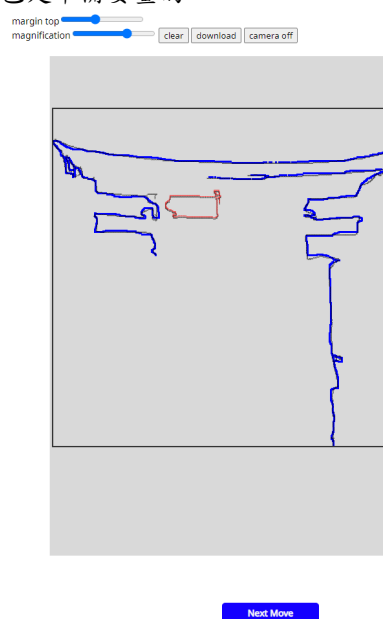


圖 25 使用者操作示意

#### 4. 討論

我們能利用重心找出筆畫順序，但跟人類筆順差太多了，但優點是速度快。利用結合 DL 技術能更貼近人性，但 YOLOv5 常常會有重複的框框，能不能有好方法去除框框？

此作品的宗旨是讓人人都能簡單的畫畫。我們希望未來能開發小孩的繪畫才能，讓他們在藝術的薰陶下有更多的人文氣息。

## 5. 結論

我們成功開發「神之一筆」 $\alpha_0$ 版。並成功使用 cordova 將網頁分享到網路上(未開放在公開網域)，能用 flask 來讓 app 與伺服器連接。而伺服器能轉換風格、計算草稿線條及筆畫順序、app 可以讓使用者容易操作。作品結合生活創意，是極富教育意義及深遠的應用。

## 參考文獻

- [1] “magenta/arbitrary-image-stylization-v1-256.” Tensorflow Hub. <https://tfhub.dev/google/magenta/arbitrary-image-stylization-v1-256/2> (accessed Feb 19, 2023).
- [2] Xin Chen and Gang Liu. “AnimeGANv2.” AnimeGANv2. <https://tachibanayoshino.github.io/AnimeGANv2/> (accessed Feb 19, 2023)