

Style Master - Airbrush Application with Deep Learning

專題組員:何芷嫣、陳衍能、鄭弘詣、高廷瑜

專題編號:PRJ-NTPUCSIE-107-04

執行期間:107 年 09 月 至 108 年 05 月

1. 摘要

修圖軟體幾乎是大家手機裡的必備項目，當有新奇的濾鏡推出時，總是會造成一波熱潮。在本專題中，我們使用深度學習技術 CNN 將圖片轉換成特別的風格，以及 Mask R-CNN 達到物件分割的效果，藉此產生一款別具特色的修圖軟體。

2. 簡介

目前常見的修圖軟體內的濾鏡大多都大同小異，主要是針對顏色飽和做調整或者是加入雜訊產生雪花感等特殊效果，變化上都比較單調。因此，我們希望設計出一款修圖軟體，讓使用者可以將一張普通的照片轉換成類似畢卡索畫作等特定風格的照片，不僅如此，我們希望讓使用者可以輕鬆的變換多種照片風格，加入了物件分割的功能，以偵測主要人物以及背景的區塊，達到針對部分區域轉換風格的方法。

3. 專題進行方式

本系統架構圖如下圖 1，當輸入圖像後，主要分為 Transfer 以及 Paint 兩大區塊，而每個區塊皆有再細分為兩個項目，其中 Transfer 主要負責作風格的轉換，而 Paint 負責讓使用者

可以根據自己的喜好自由選擇上色區塊。詳細的研究方法說明如後。

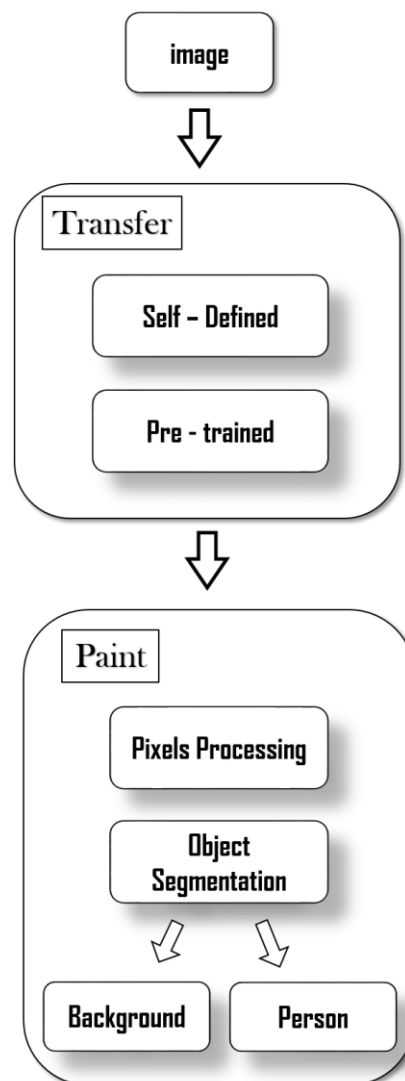


圖 1、系統架構圖

3.1 圖像的輸入

可利用相簿選擇圖片，亦或開啟相機拍攝相片。

3.2 Transfer

風格轉換的主軸是利用學習的方式，將原圖、風格圖以及結果圖(最初是雜訊圖)透過神經網路 VGG-network 取得特徵向量後，計算結果圖與原圖的 Loss 以及結果圖與風格圖的 Loss，藉此做梯度下降，一次一次的更新結果圖的像素，就能產生出類似於風格圖的原圖，但對於原圖又不會失真過多。但這種迭代更新的方法速度非常緩慢，而根據這種方法也衍生出了許多實時的風格轉換。

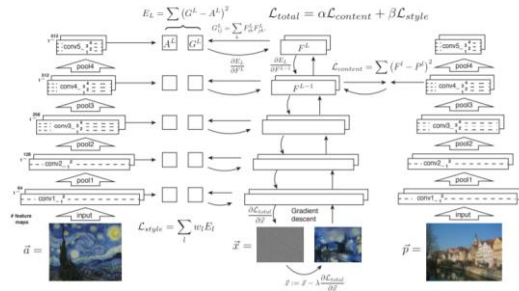


圖 2、風格轉換概念[1]

$$\mathcal{L}_{\text{content}}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2$$

$$\mathcal{L}_{\text{style}}(\vec{a}, \vec{x}) = \sum_{l=0}^L w_l E_l,$$

$$\mathcal{L}_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{\text{style}}(\vec{a}, \vec{x})$$

圖 3、三種 Loss function[1]

3.2.1 Johnson's Model

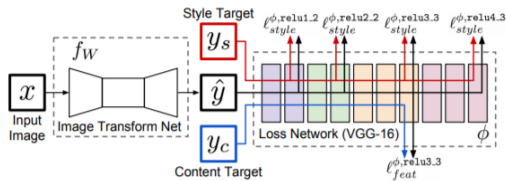


圖 4、Johnson 模型的架構圖[2]

相較於上述採用更新像素，也就是 pixel-to-pixel 的方式，Johnson 提出的模型是利用計算 perceptual loss，也就是只針對高度特徵做計算，來加速訓練時間。主要方法是將訓練好的神經網路(Loss Network)，提取出高級特徵後得到 feature reconstruction loss 以及 style reconstruction loss(圖 5)。

$$\ell_{feat}^{\phi,j}(\hat{y}, y) = \frac{1}{C_j H_j W_j} \|\phi_j(\hat{y}) - \phi_j(y)\|_2^2$$

$$G_j^{\phi}(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'}$$

$$\ell_{style}^{\phi,j}(\hat{y}, y) = \|G_j^{\phi}(\hat{y}) - G_j^{\phi}(y)\|_F^2$$

圖 5、兩種 reconstruction loss[2]

結合上述兩個 loss function，可以產生出預估結果圖，而只要透過更新權重的方式，就可以保證訓練完後得到的 Image Transform Network 是一個擁有高度特徵的風格轉換模型。

相較於 Gatys 的方法，Johnson 的模型將訓練時間壓縮到 2 到 3 小時，不僅如此，訓練完的模型可以隨意套入不同圖片，都可以快速地產生轉換後的結果圖，但一種模型就只代表一種風格。

$$\hat{y} = f_W(x) = \arg \min_y \lambda_c \ell_{feat}^{\phi,j}(y, y_c) + \lambda_s \ell_{style}^{\phi,j}(y, y_s) + \lambda_{TV} \ell_{TV}(y)$$

$$W^* = \arg \min_W \mathbf{E}_{x, \{y_i\}} \left[\sum_{i=1} \lambda_i \ell_i(f_W(x), y_i) \right]$$

圖 6、更新 fw 的權重[2]

3.2.2 AdaIN

雖然 Johnson 的模型效果比較好且訓練時間縮短，訓練完後的模型不管套入什麼圖片都可以即時產生結果，但還是不得免除要訓練 2-3 小時

的時間，因此我們只預先訓練了 10 種風格，其餘風格則讓使用者自行選擇，於是我們另外採用了此方法。

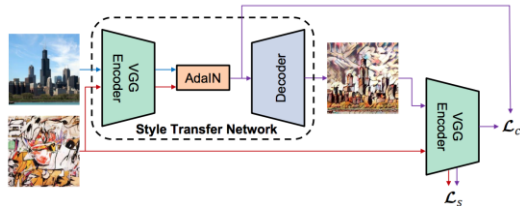


圖 7、AdaIN 的架構圖[3]

一樣是透過 VGG-network 取得特徵向量，AdaIN 的方法是直接將原圖與風格圖的特徵向量做線性轉換得到融合後的向量，之後再使用解碼器將融合後的向量轉換成圖片輸出。由於只需要經過演算的時間，而不是訓練的結果，也不須經過 gradient descent，因此可預期是即時的結果。

$$\text{AdaIN}(x, y) = \sigma(y) \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y)$$

圖 8、AdaIN 演算法。[3]



圖 9、左上：原圖 右上：風格圖
左下：Johnson 模型 右下：AdaIN

從上圖可看出 Adain 在形狀上有比較大的誤差，且效果沒有 Johnson 訓練好的模型好。。

3.3 Paint

本節目的在於讓使用者根據自己喜好決定上色區塊，此外，也包含基

本修圖軟體所有的功能，例如：橡皮擦、清除全部、全部上色等等……。

3.3.1 Pixel Processing

類似於小畫家筆刷的繪圖方式，當使用者選擇好風格後，可選擇筆刷大小，再根據自己的喜好，對任意 pixel 上色。而根據使用者所觸碰之像素位置，記錄後將相對應的風格結果圖即時呈現在畫布上，就能達成這種筆刷式的繪圖效果。



圖 9、pixel processing 示意圖

3.3.2 Object Segmentation

相較於手動上色，我們希望讓使用者以更輕鬆的方式選擇上色區塊，在此我們使用了物件分割技術，選取出主要人物以及背景兩大區塊，可直接針對部分區域上色。

因此我們使用 Mask R-CNN，他能分辨圖片中將近 90 種物體，但由於時間方面考量，我們只針對主要的人物以及背景做遮罩。Mask R-CNN 使用了 Faster R-CNN 提取 Region Proposal 的方法，先將圖片經過 CNN 取得 Feature Maps，再使用 Region Proposal Network 產生出最有可能包

含物體的 bounding boxes 以及其機率，再利用這些 boxes 做下一步分類預測。

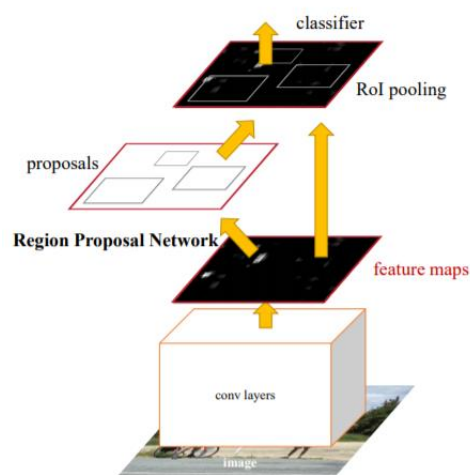


圖 10、取得 proposals[6]

RPN 運作方式是透過取 sliding window，事先準備好不同尺寸比例的 box 去計算出含有物體的機率最高的 bounding boxes。

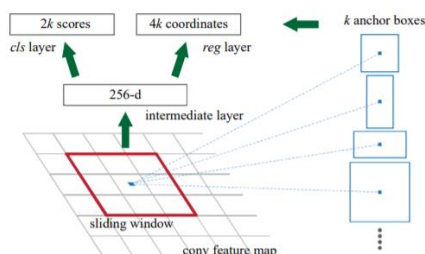


圖 11、RPN[6]

由於 CNN 做回歸分析或分類預測時大部分都會限定尺寸，因此再做預測前，先使用 ROIAAlign 的方法，將 bounding boxes 變成固定尺寸，再通過全連接層進行 bounding boxes 分類預測，預測完後再透過迴歸分析以便更精確的定位。此外，ROIAAlign 的方法就是做雙線性內插法。

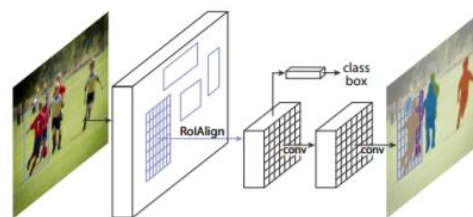


圖 12、Mask R-CNN 取得 proposals 後的步驟[7]

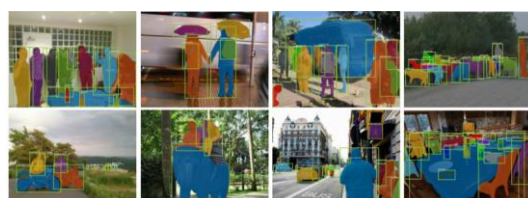


圖 13、Mask R-CNN 結果示意圖[7]

從上圖可以看出，相較於 faster R-CNN 只能選取矩形區塊，Mask R-CNN 可以將遮罩逼近整個物體的形狀，不僅如此，也能分辨很多類物件。



圖 14、結果示意圖

4. 主要成果與評估

此次成果有兩種呈現方式，第一種是 IOS 系統的手機應用程式，第二種是電腦版的應用程式。

手機版的執行流程如下圖所示：

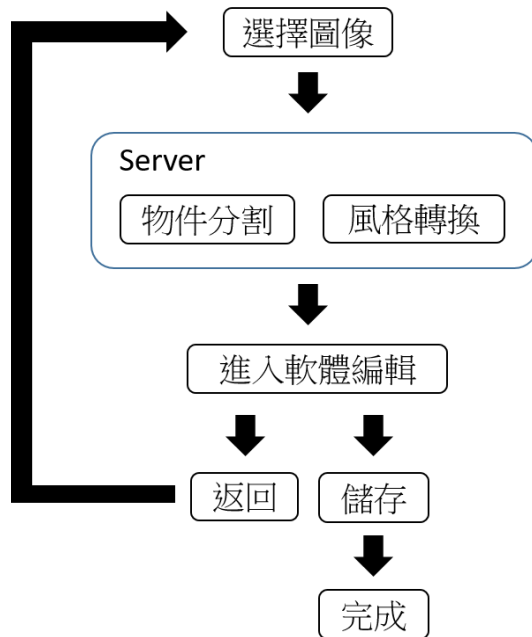


圖 15、執行流程圖

由於手機硬體不能支持龐大的神經元網路運算，我們將大量運算過程放到 Server 端，由 Server 運算完成後再傳回手機裡。進入編輯後可自由選擇變換之區域與風格，再將最後成果儲存，或者返回選擇圖片。

電腦版的操作雷同，但不需要 Server，直接透過電腦運算，不同的是，我們使用開啟按鈕，才將畫布開啟而不是直接呈現在畫面上。

5. 結語與展望

本次專題當中，我們成功做出一款修圖軟體。利用大量深度學習技術，產生出特別且新奇的風格，是目前市面上很罕見的應用程式。

由於龐大的神經網路，讓運算時間增加，雖然使用 server 做運算，一

次回傳近 30 張圖片還是需要一定的時間，未來希望可以針對時間性去做優化，讓此應用程式更便於使用者。只要時間可有效壓縮，未來也可以針對更多物體做遮罩，增加此應用程式的運用範圍。

6. 銘謝

感謝指導教授以及實驗室的學長提供各式資源與指導，一步步帶領我們向前進。也感謝組員們的努力才能使我們的成品如期完成。

7. 參考文獻

- [1] Gatys, et al., "A Neural Algorithm of Artistic Style.", 2015.
- [2] Justin Johnson, Alexandre Alahi, Li Fei-Fei., "Perceptual Losses for Real-Time Style Transfer and Super-Resolution.", 2016.
- [3] Xun Huang, Serge Belongie, "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization", 2017
- [4] Alex "Imagenet classification with deep convolutional neural networks", 2012.
- [5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation.", 2004.
- [6] Ren, S., He, K., Girshick, R., Sun, J., "Faster R-CNN: Towards real-time object detection with region proposal networks", 2015.
- [7] Kaiming He, et al., "Mask R-CNN", 2018.

