

國立台北大學資訊工程學系專題報告

即時運動賽事虛擬動態廣告系統

Real-time Sport Event Virtual Advertising System

專題組員:劉兆翔、郭庭恩、王秉坤

專題編號:PRJ-NTPUCSIE-106-06

執行期間:106 年 09 月 至 107 年 06 月

1. 摘要

當國際賽事在國外播放時，部分在賽場上設置廣告看板的廠商，服務區域並不包含當地，導致轉播中的廣告對當地觀眾不起作用。本計畫”即時運動賽事虛擬動態廣告系統”旨在追求開發出一套軟體，利用影像處理、CNN 深度學習與辨識等技術，將賽事直播畫面中的看板內容，替換成當地廠商的廣告，使提升賽事轉播的商業利潤，使更多廠商受惠。

2. 簡介

隨著時代演進，各式各樣的國際賽事越來越多，廠商們紛紛在比賽場地設置廣告，藉由轉播、重播達到宣傳的目的。當賽事在國外播放時，因為地域問題，部分在賽場上設置廣告看板的廠商，服務區域並不包含當地，導致轉播中的廣告看板對當地觀眾不起作用，形同虛設。若能透過一套軟體，將賽事直播畫面中的看板內容，替換成當地廠商的廣告，便能提升賽事轉播的商業利潤。

為達成上述目的，所遇到的問題大致分成四種：看板定位、內容辨識、選手遮蔽、速度。在更換看板內容前，必須先判斷看板位置、形狀，而影片中看

板的形狀會隨著攝影機角度而有所改變。而在確認看板內容過程中，看板內容顏色會因光線、陰影產生變化，而和原看板有出入。在更換看板時，若選手恰巧遮蔽原看板，也必須盡量保持選手原貌。最後，到上述效果的時間不能太長，以利賽事直播的進行。計畫示意圖如圖 1、圖 2、圖 3。



圖 1、輸入轉播影像



圖 2、定位目標看板



圖 3、投影新的看板至原始影像

3. 專題進行方式

本系統之架構如下：第一步驟，透過 HSV 色彩空間設定閾值，FloodFill(漫水填充演算法)及 Morphology(形態學)進行看板之初步定位。第二步驟，利用找出最小外接矩形搭配 CNN 神經網路定位出確切的看板位置。第三步驟，利用 Geometric Transform(幾何變換)、Background subtraction(背景相減法)將虛擬廣告投影至原看板，並進行遮蔽物處理。系統架構圖如圖 4 所示，詳細研究方法及步驟說明如後。

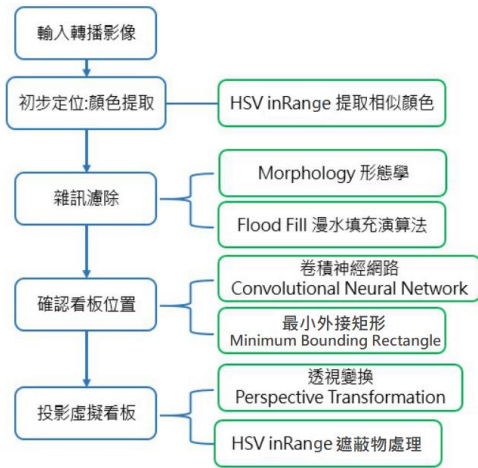


圖 4、系統架構圖

3.1 影像的輸入

轉播的影像(如圖 5)經由檔案或串流輸入後，利用電腦執行程式進行影像處理

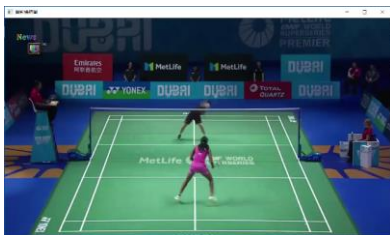


圖 5、讀入影像

3.2 看板定位

本節之目的是在於偵測與定位出影像中欲替換之看板的準確位置。首先利用 HSV 顏色提取法，以欲替換看板之背景色作為基準，將影像中可能為看板的部分分離出來。經過形態學、漫水填充等等處理消除雜訊後，再使用凸包運算找出該區域的最小外接四邊形。之後將該區域擷取下來，送入 CNN 神經網路進行辨識，判斷該區域是否真的為欲替換的看板。

3.2.1 HSV 顏色提取法

將初始影像由 RGB 色彩空間轉換為 HSV 色彩空間，設 (r, g, b) 分別是一個顏色的紅、綠和藍坐標，它們的值是在 0 到 1 之間的實數。設 $\max$ 等價於 r, g 和 b 中的最大者，設 $\min$ 等於這些值中的最小者。轉換公式[5]為

$$h = \begin{cases} 0^\circ & \text{if } \max = \min \\ 60^\circ \times \frac{g-b}{\max-\min} + 0^\circ, & \text{if } \max = r \text{ and } g \geq b \\ 60^\circ \times \frac{g-b}{\max-\min} + 360^\circ, & \text{if } \max = r \text{ and } g < b \\ 60^\circ \times \frac{b-r}{\max-\min} + 120^\circ, & \text{if } \max = g \\ 60^\circ \times \frac{r-g}{\max-\min} + 240^\circ, & \text{if } \max = b \end{cases}$$

$$s = \begin{cases} 0, & \text{if } \max = 0 \\ \frac{\max-\min}{\max} = 1 - \frac{\min}{\max}, & \text{otherwise} \end{cases}$$

$$v = \max$$

在 HSV 色彩空間下，依據欲偵測物件的顏色，將三個通道設置不同的閾值，可以得到原影像對特定顏色的遮罩(mask)，計算方式依據[6]：

$$dst(I) = lowerb(I) \leq src(I) \leq upperb(I)$$

在 HSV 色彩空間下，其不同顏色的敏感度及光影的適應能力，會比在 RGB 空間下設定閾值來提取顏色更為優異。



圖 6、原始影像



圖 7、對深藍色的遮罩

3.2.2 Morphology 形態學中的閉運算 (Close)

閉運算為先將影像作膨脹 (Dilation) 運算，再作影像之侵蝕 (Erode) 運算。其功用在於填補影像中的小洞、彌補狹窄的間斷、以及使物體輪廓變為平滑。[7]

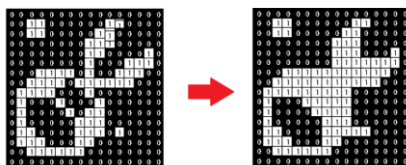


圖 8、閉運算的效果[8]

3.2.3 漫水填充演算法(Flood Fill)

漫水填充演算法是從一個區域中提取若干個連通的點與其他相鄰區域區分開的演算法，因為其思路類似洪水從一個區域擴散到所有能到達的區域而得名。

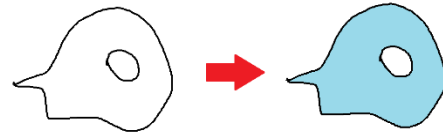


圖 9、漫水填充示意圖

在二值化(Binary Image)的圖像中，透過漫水填充演算法將圖像中若干的區域分群，再透過設定面積閾值，可以在不破壞目標區域的情況下，將二值化的雜訊濾除。



圖 10、二值化原影像

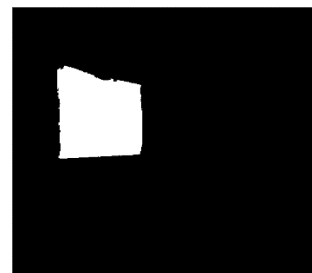


圖 11、漫水填充後

3.2.4 運用卷積神經網路 (Convolutional Neural Networks) 進行廣告辨識

傳統 DNN(Deep neural network) 網路的輸入為一維的資料，在分析圖像時，必須將三維的資訊轉為一維。若去除了這些形狀資訊，就代表丟失了一些重要的空間資料。為了解決這個問題，卷積神經網路(CNN)較傳統的DNN多了卷積(Convolutional)及池化 (Pooling) 兩層，用以保留原始影像的空間資訊。

[11]

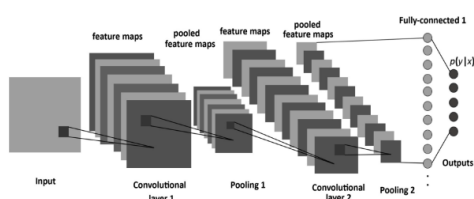


圖 13、CNN 網路架構[11]

卷積層的原理是透過一個指定尺寸的 Filter，在 Filter 的每一格給予不同的權重，由上而下依序滑動取得圖像中各局部特徵，計算加權總合，作為下一層的輸入。

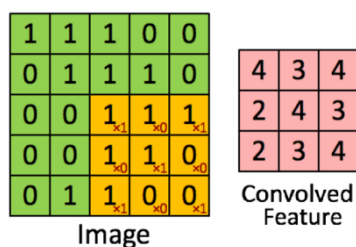


圖 14、卷積層示意圖[12]

池化層的功能則是將卷積後的特徵尺寸縮小，其好處包括減少運算量以加快模型運算的效率；當圖像中某些像素在鄰近區域有微小偏移或差異時，增加抗干擾的能力；減少過度擬合 (overfitting) 的情形。[13]

以池化層最常使用的最大化 (Max-Pooling) 為例，池化會在圖片上選取不同窗口 (window)，並在這個窗口範圍中選擇一個最大值，藉以降低特徵的維度。

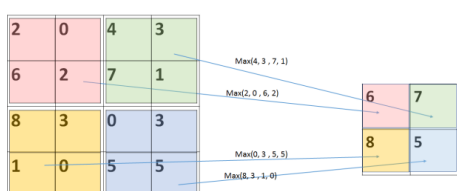


圖 15、池化層示意圖[11]

經過多次卷積與池化的過程後，將最後所得到的特徵輸入至全連接層 (Fully Connected Layer) 進行分類及訓練，此即為 CNN 的典型作法。

因為單純使用顏色定位看板，將無法分辨出兩個底色相似的看板之差別 (如圖 16、17)。故本計畫預期使用 CNN 神經網路來解決此種情形。



圖 16、原始影像圖



圖 17、顏色分離後的結果

首先將顏色分離後的結果區域使用 Canny 邊緣檢測演算法 (Canny edge detector) 提取邊緣 (edge) (如圖 18)，再將此二值化的圖像送入一個輸出為兩類的 CNN 二元分類 (Binary Classification) 模型。



圖 18、Canny 邊緣檢測

在訓練此模型時，第一類的輸入為欲替換的看板，也就是正樣本 (positive example)，第二類為我蒐集的其他不相關的看板，也就是負樣本 (negative

example)。

當訓練完成後，當輸入為欲替換的看板區塊，CNN 網路會將他分類為第一類，反之則分為第二類。如此即可判斷該區塊是否為欲替換的看板。

為了解決正樣本數不足的問題，這裡使用到一個數據增強(Data Augmentation)的技巧，透過對原始影像進行如旋轉(Rotation)、縮放(zoom)、平移(shift)等等幾何變換，來增加輸入樣本的量，以解決正樣本難以大量蒐集的問題。

3.3 投影虛擬看板

在定位出欲替換看板的位置後，程式要將新的看板投影到原先的看板上。首先通過幾何變換中的透射變換(Perspective Transformation)將虛擬的看板投影至欲替換的看板上，再使用 HSV inRange 的手段處理運動員遮蔽看板的問題。

3.3.1 幾何變換 (Geometric Transformation)

幾何變換是指從具有幾何結構之集合至其自身或其他此類集合的一種對射，以其操作集合的維度來分類，可分類出平面變換與空間變換等。幾何變換亦可依據其保留其性質來分類[12][13]。

透過幾何變換，能夠將一個固定形狀的看板，投射到任意一個四邊形上，達到替換看板的效果。

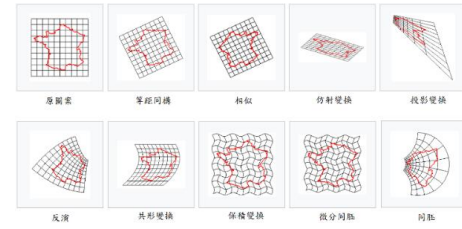


圖 19、幾何變換的分類[14]

3.2 透視變換(Perspective Transformation)

透視變換是將圖片投影到一個新的視平面(Viewing Plane)，也稱作投影映射(Projective Mapping)。通用的變換公式為[16]：

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

u, v 是原始圖片，對應得到變換後的圖片座標 x, y ，其中

$$x = x' / w', y = y' / w'$$

$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ 表示線性變換， $\begin{bmatrix} a_{31} & a_{32} \end{bmatrix}$ 用於平移， $\begin{bmatrix} a_{13} & a_{23} \end{bmatrix}^T$ 用於產生透射變換。

將變換公式整理可得：

$$x = \frac{x'}{w'} = \frac{a_{11}u + a_{21}v + a_{31}}{a_{13}u + a_{23}v + a_{33}}$$

$$y = \frac{y'}{w'} = \frac{a_{12}u + a_{22}v + a_{32}}{a_{13}u + a_{23}v + a_{33}}$$

因此，已知變換對應的幾個點可以求取變換公式，而特定的變換公式也能求得新的變換後的圖片。

以一個正方形變換成四邊形為例：變換的 4 組對應點可表示為

$$(0,0) \rightarrow (x_0, y_0), (1,0) \rightarrow (x_1, y_1), (1,1) \rightarrow (x_2, y_2), (0,1) \rightarrow (x_3, y_3)$$

根據變換公式可得

$$\begin{aligned} a_{31} &= x_0 \\ a_{11} + a_{31} - a_{13}x_1 &= x_1 \\ a_{11} + a_{21} + a_{31} - a_{13}x_2 - a_{23}x_2 &= x_2 \\ a_{21} + a_{31} - a_{23}x_3 &= x_3 \\ a_{32} &= y_0 \\ a_{12} + a_{32} - a_{13}y_1 &= y_1 \\ a_{12} + a_{22} + a_{32} - a_{23}y_2 - a_{23}y_2 &= y_2 \\ a_{22} + a_{32} - a_{23}y_3 &= y_3 \end{aligned}$$

求解出的變換矩陣就可以將一個正方形變換到四邊形。反之，四邊形變換到正方形亦同，因此，欲將一四邊形變換到另一四邊形，只要先將其變換為正方形，再變換為目標的四邊形即可 [17][18]。

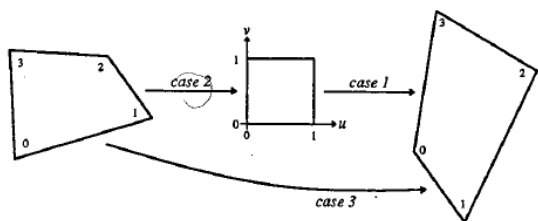


圖 20、透視變換運算原理[19]



圖 21、透視變換結果圖

3.3.2 遮蔽物處理

當將虛擬的影像投影至原始影像中，有可能會遇到運動員遮蔽看板的情形。如沒有進行處理，會造成新投影上去的看板將運動員遮蔽住。透過再進一步對看板內之顏色進行前述之 HSV inRange 處理，即可得到對於完整看板的遮罩(mask)。當遮蔽物進入看板範圍時，便可得到該遮蔽物之輪廓。如此便可以處理運動員遮蔽看板的問題。



圖 22、原始影像



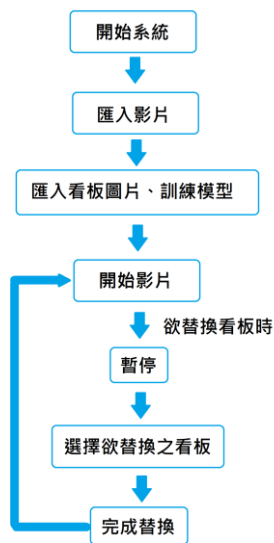
圖 23、對於遮蔽物的遮罩



圖 24、處理後的結果

4. 主要成果與評估

本系統之執行流程圖如下



首先將影片匯入，再選擇各置換看板之模型與欲替換圖片。當影片正在執行時，如欲替換看板，可將影片暫停，再點選影片中之看板，即可完成替換。

圖 25 為本計畫結果示意圖，左上格為原始輸入賽事轉播畫面，右上格為韓國地區置換為收視者所看到的廣告，左下格為日本地區置換為收視者所看到的廣告，右下格為台灣地區置換為收視者所看到的廣告。透過這樣子的置換，可增進置放球場廣告的效益。



圖 26、結果示意圖

程式開發環境為在 Linux 下使用 Anaconda 與 OpenCV 進行開發。

在 1080 x 607 之解析度下，替換最高四個看板時，影片執行之每秒幀數仍能保持在約 23，符合主題即時替換之需求。

5. 結語與展望

本次的專題研究中，成功在 Linux 系統之下，以各種影像處理之演算法搭配 CNN 類神經網路，成功實踐了即時的虛擬廣告系統，使得廣告商在置放廣告時有更大的彈性。未來希望可以在投影上廣告時，根據球場現場的狀況，對更新上去的看板作額外的處理，使虛擬的看板更為融入背景，呈現更為自然的效果。

6. 銘謝

感謝指導教授與實驗室的學長姐們提供各式的資源以及指導，帶領我們走過專題的每一個流程。也謝謝專題的夥伴們，在開發的過程中都給予莫大的幫助，讓我們能順利完成這次的專題。

7. 參考文獻

- [1] The University of Edinburgh School of Informatics: Affine Transformation
- [2] 曲建仲：翻轉人類未來的 AI 科技：機器學習與深度學習
- [3] 李怡欣：以特徵圖資訊為基礎的網球影片虛擬廣告植入方法，2011
- [4] 陳韋亘：半自動二維廣告置入影像之研究，2011
- [5] Wikipedia: HSL and HSV
- [6] OpenCV 3.4.0 tutorial: Changing Colorspaces
- [7] Wikipedia: Closing (morphology)

- [8] School of Informatics: Image Processing Learning Resources, close
- [9] 線代啟示錄：凸組合、凸包與凸集
- [10] Wikipedia: Convex Hull
- [11] CH.Tseng: 初探卷積神經網路
- [12] CDSN 閱微草堂 123: Deep Learning——CNN (Convolution Neural Network)
- [13] Data Science and Robots Blog: How do Convolutional Neural Networks work?
- [14] Wikipedia: Geometric transformation
- [15] Public Lab: GSoC Update: Leaflet.draw and Non affine transformations
- [16] P. S. Heckbert, Fundamentals of Texture Mapping and Image Warping, 1989.
- [17] J. Hu; D. Q. Zhang; H. Yu; C. W. Chen, MULTI-OBJECTIVE CONTENT PRESERVING WARPING FOR IMAGE STITCHING
- [18] 【圖像處理】透視變換 Perspective Transformation：透視變換
- [19] OpenCV 3.4.0 tutorial: Perspective Transformation
- [20] 鄧玉祥：基於高斯混合模型背景影像建立之研究，2010
- [21] OpenCV 3.4.0 tutorial: How to Use Background Subtraction Met