TrackNet: 利用深度學習熱度圖追蹤轉播影片中之網球
TrackNet: Tennis Ball Tracking from Broadcast Video by Deep Learning Networks

Network Optimization Lab (NOL)
Department of Computer Science, College of Computer Science
National Chiao Tung University
1001 University Road, Hsinchu City 30010, Taiwan
Emails: andersen.cs05g@g2.nctu.edu.tw, cwyi@nctu.edu.tw

Abstract: 球和球員的軌跡可用於分析比賽策略及評估球員的表現，基於電腦視覺的物體軌跡追蹤技術已被用於註釋和增強運動影片，但某些球類運動（如：網球、羽球、棒球等）因球的體積小速度快，造成影片中的成像既小且模糊，使得飛行軌跡標記變得困難，需動用多台高速高解析攝影機的幫助。針對一般的比賽錄影，本論文提出一個以 CNN 為基礎的的深度學習架構，以數張連續幀產生網球的偵測熱度圖，以定位網球位置，進而計算飛行軌跡；這個提案模型不僅能從模糊的影像中定位網球，還能夠判斷受遮擋的網球位置。我們完整標記 2017 台北世大運男子單打的冠軍賽，共 20846 張相片；並另外標記 9 場比賽的片段，共 17968 張相片。10 場比賽各取約 2000 張相片，採 10 分法進行精確度驗證，模型的精確度（Precision）、召回率（Recall）及 F1 測量值（F1-measure）分別高達 98.9%、78.3%及 87.4%；利用傳統影像處理方法辨識 2017 世大運比賽影像，取得的精確度、召回率及 F1 測量值分別為 88.7%、74.4%及 80.9%，實驗結果顯示我們提出的演算法所優於傳統的方法。論文所使用的資料集可透過連結??下載取得。

Keywords: 軌跡追蹤、網球、深度學習、熱度圖、球賽廣播

1. 簡介

影片記錄了大量的視覺資訊，被視為重要的視覺感測器日誌，解譯影片內容是近年影像處理與深度學習領域的熱門研究。在運動學習和選手培訓的應用中，比賽錄像常用於賽後檢討與戰術分析。以高爾夫學習為例，攝影機可用於捕捉關鍵的揮桿姿勢，透過觀看回放影片或測量影像中肢體間的角度或相對位置，教練即可指出揮桿動作的問題。就籃球比賽而言，影片則常用來於評估球員的反應，透過大數據分析，甚至可預測採取攻擊或防禦戰術所造成的得失分變化期望值，在勢均力敵的比賽中，勝負將可能因採用不同戰術所帶來的比分變化而改變。

數據收集在大數據分析中是最基本的，在 NBA 的生態系中，專業公司利用高解析攝影結合影像處理來計算球的傳導與球員的跑位資料，但這樣的方案所需的財力與資源並不是一般個人或球隊所能負擔的。另外，像網球、羽球、棒球等運動，因球的體積小且移動速度快，會有物件影像小且模糊的問題，更提高問題的複雜度。在這個研究中，針對日常的網球比賽影片，我們設計一個深度學習網路架構，用來偵測並定位一般 3C 設備所錄製的比賽影片中的網球，除了可克服影像模糊及殘影等問題，甚至可定位被遮蔽的網球。這項技術未來可擴展到其他的球類運動，普遍應用在各級體育賽事，協助業餘及職業團隊進行資料收集及應用。

傳統影像物件辨識會根據物件的外顯特徵（如：形狀、顏色及尺寸等）或統計特徵（如：HOG 或 SIFT）進行偵測，但對於體積小且移動快的物件，因快門時間相對較長，會有影像殘留及模糊的現象，導致偵測錯誤及低識別率。針對網球追蹤，可利用飛行軌跡的特性進行補強，透過檢測幀與幀之間合理的物理模型，從眾多可能的候選物件中進行配對，從而找出到最可能的軌跡[1]。此外，針對模糊圖像問題，有透過融合數張圖像來生成清晰圖像的技術。基於上述的觀察，我們嘗試避免使用規則導向的影像處理技術，改採深度學習網絡學習球的外形，同時利用多張連續影像學習球的移動模式，以改善前述相關的問題。

物件分類與偵測是深度學習最早探索的問題之一，其中 VGG-16 [6]的影像特徵圖編碼網絡，是後續常被使用或參考的設計。因圖片中可能同時存在多個物件且物件大小不固定，R-CNN 家族[2][3][4]結構化地檢測畫面，從中先找出多個可能包含物件的區域，稱為 Region of Interests (RoIs)，再對這些區域進行仔細的物件偵測與分類，但因計算速度較慢無法達到即時應用的需求。為了加快處理速度，YOLO 家族[5]使用一階段的方法，只在有限的搜尋空間中偵測物體，顯著提高處理速度，精簡版本 Tiny YOLO 甚至可以在 Raspberry Pi 上運行。相對於以區塊為單位的算法，完全卷積網路（Fully Convolutional Networks, FCN）則是以像素為單位進行分類，為了還原特徵圖編碼過程所減損的尺寸，上取樣（upsampling）及逆卷積網（DeconvNet）常被用在將特徵圖解碼生成對應原尺寸大小的資料陣列，如：熱度圖（heatmap）可用在指示物件的存在。

綜合上述的陳述，我們設計如下的軌跡追蹤深度學習網絡。首先，採用 VGG-16 為原型進行特徵圖編碼，但不同其他的深度學習網絡，我們容許一次輸入多張連續的幀，可從中不只學習球的影像特徵，也學習球的軌跡特性，以期達到特徵加成的效果。其後，仿效 FCN 的生成階段，還原特徵圖編碼的池化層造成的下取樣，解碼生成用於偵測及定位球的熱度圖。最後，接在深度學習網絡之後，我們依據熱度圖計算畫面可能存在的網球，為了符合影片的特性，我們假設畫面中最多只有一顆球來進行計算及評估。

為了訓練及評估深度學習網絡，我們完整標記了 2017 世界大學運動會網球男子單打決賽的廣播影片[7]，共計 20846 張圖片。為了驗證同時輸入多張連續幀的效果，我們實作兩種網絡，分別標示為模型 I 及模型 II：模型 I 僅以單一圖像作為輸入，而模型 II 則以三張連續幀為輸入。實驗結果顯示，模型 I 的 Precision 和 Recall 分別為 95.2%和 89.2%，而模型 II 則可高達 99.8%和 96.1%。在定位的精確度上，模型 I 和模型 II 的平均誤差分別為 2.0 像素和 1.3 像素。可發現使用連續圖像的模型 II 的準確性得到顯著改善，甚至可偵測到被遮擋的球。與傳統圖形識別演算法[1]相較，根據我們的實作其 Precision 及 Recall 分別為 88.7%和 74.4%，不論是模型 I 或模型 II 皆大幅提昇識別的準確性。為了避免及評估可能的過度擬合（overfitting），我們另外挑選了 9 場不同場景的比賽進行標記，包含紅土球場、草地球場、黑白影片等不同場景，每場比各取約 2000 張畫面，再從原有資料集亦取出約 2000 張畫面，再以影片為組別進行 10-fold 驗證，最終的 Precision、Recall 及平均定位誤差分別為 98.9%、78.3%及 5.4 點像素。

針對如網球體積小速度快的物件偵測與定位，本論文提出一個可輸入連續圖像的深度學習網絡框架，學習連續圖像間物件的特性與相關性，實驗顯示這個方法可有效從日常廣播影片中偵測與定位網球，從而計算網球的軌跡。本論文的其餘部分安排如下：第二節提供相關的研究文獻簡介和卷積神經網絡的介紹；第三節提供本研究中使用的資料集的介紹；第四節介紹研究所提出

的 TrackNet 深度學習網絡和高斯熱圖的處理技術；第五節提供相關效能評估數據及說明；最後，第六節是我們的結論。

2. 研究背景及文獻探討

近年來，根據球員及球的軌跡來分析球員表現及比賽戰術的研究受到越來越多的關注[8][9][10][11]，許多的追蹤演算法及系統已被提出來。商業解決方案依賴高解析的高速攝影機進行比賽錄影，所需的硬體投資及營運成本相當高。例如，Hawk-Eye 系統[12]已廣泛用職業比賽，用來計算球的飛行軌跡，並透過視覺化的 3D 描繪，協助裁判澄清有爭議的判決，但該系統需在選定的位置和角度部署攝影機並搭配專人操作，對於大多數人而言過於昂貴。

嘗試從普通影片中定位球亦有多年的研究，但一來因球體積小，影像所占面積小，容易與畫面中相似顏色或形狀的景物混淆，常有錯誤偵測（false positive）發生；二來因球移動快，會造成影像模糊，會有失敗偵測（false negative）發生。透過探索連續幀之間球的軌跡模式，可有效改善網球的辨識與定位。此外，飛行軌跡本身即是一個重要的資訊，也是一些研究的主體[13]，如：使用多攝影機結合 3D 技術進行網球偵測和追蹤[14]；採用粒子濾波器（particle filter）在低質量的影片中追蹤網球[15]；以雙層資料關聯法從逐幀影像處理含有錯誤及失敗偵測的結果中，算出最有可能的球軌跡[16]。

深度學習在圖像分類問題上的成功[6][17]，鼓勵越來越多的研究人員採用這些技術來解決各種問題，如：物體偵測和截取[4][5][18]、電腦遊戲、網絡安全、活動識別[19][20]、文字/圖片語義理解、及智能商店等。深度學習網絡的基礎架構是藉由大量數據來訓練結構化但龐大的卷積類神經網絡（CNN），層與層之間最常見的運算包括：convolution、rectifier、pooling/downsampling 和 deconvolution/upsampling，最後再結合完全連接層或 Softmax 層，例如：被廣泛使用的 VGG-16 [6]，主要由 convolution 層、maximum pooling 層和 ReLU 層所組成。概念上來說，前端的層學習識別簡單的幾何特徵，而後端的層則被訓練識別物件特徵。

在卷積類神經網絡中，每層是$W \times H \times D$的數據陣列。W、H和D分別是數據的寬度、高度和深度。卷積運算是由一個$w \times h \times D$大小的運算核在橫跨$W \times H$範圍上計算的濾波器；步幅（stride）參數s在許多情況下常設為 1；為避免邊界附近的資訊流失或維持輸出數據陣列的行數或列數，可透過填充（padding）參數p為數據陣列四周添加數值為 0 的列和行來達成，Figure 1 描繪卷積運算的相關參數。若W'和H'分別表示下一層的寬度和高度，則有如下的等式：

$$W' = \frac{W + 2p - w}{s} + 1 \text{ and } H' = \frac{H + 2p - h}{s} + 1$$

而下一層的深度則是濾波器的數量。

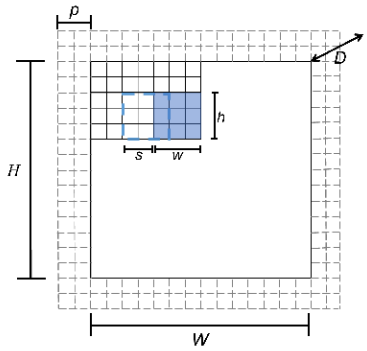

Figure 1 深度學習網絡中的卷積運算

因卷積運算是線性的，無法有效捕捉非線性行動，因此引進稱為 rectifier 的激活函數（activation function）導入非線性行為。線性整流函數（Rectified Linear Unit; ReLU）是深度學習模型中最常用的激活函數，當輸入值為負時，則該函數輸出 0，否則將輸入值直接輸出。ReLU 可表為$f(x) = \max(0, x)$，其函數圖如 Figure 2 所示。maximum pooling 具有 downsampling 及特徵融合的功能，一個區塊的資料會以區塊中最大數值取代，Figure 3 是 2x2 maximum pooling 的示意圖。Pooling 會減少數據陣列的大小，因此在逐像素的分類網絡[21][22]需要進行 upsampling，以建構對應原圖像素數量的輸出陣列，Figure 4 是 2×2 upsampling 的例子。為了限制數據的大小範圍並進行公平的比較，batch normalization 是一個廣泛被使用可加速訓練過程的技術，每片$W \times H$數據陣列將會被獨立地標準化為正規分布。

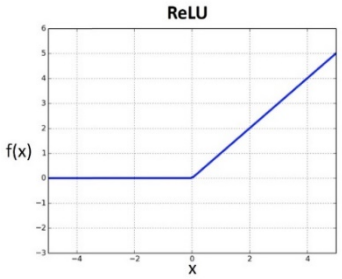

Figure 2 ReLU 函數圖形

Figure 3 2x2 maximum pooling 的範例

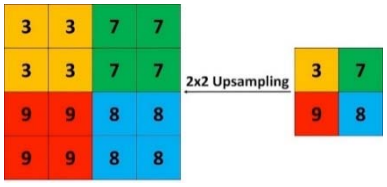

Figure 4 2x2 upsampling 的範例

反向傳播算法（backward propagation）廣泛用於類神經網絡的訓練，更新濾波器及連結的權重，先以損失函數（loss function）評估當前模型的好壞，再由梯度下降法（Gradient Descent）據

以更新權重，過程中以鏈鎖律（chain rule）分層依序計算損失函數的梯度。損失函數的設計是影響訓練效率及網絡模型效能的重要因素，常用的損失函數包括方均根誤差（Root Mean Square Error, RMSE）和交义熵（cross-entropy）等。

在這個論文中，我們提出了一個名為 TrackNet 的深度網絡框架，可從一般網球比賽影片中偵測網球。TrackNet 可讀入數張連續幀，不僅利用是外觀來偵測網球，還可學習網球的軌跡模式，最後生成一張檢測網球的熱度圖。熱度圖的想法已被用於一些研究[23][24]，TrackNet 所生成的熱度圖是以網球圖像為中心的類高斯分佈。

為了比較與評估 TrackNet 的效能，我們實作了[1]中以傳統影像處理技術來偵測網球的演算法，並稱之為 Archana's。Archana's 會先將每一幀的圖像以中值濾波器（median filter）進行平滑處理以去除細小的雜訊；計算背景模型（background model）後，進行背景去除（background subtraction）以取得前景；因網球的移動速度快，可從前後兩幀的差異中找出可能是網球的部分；再用邏輯 AND 運算前兩者得到是前景且移動相對較快的，以形狀、大小和長寬比來判斷，從中找出可能是網球的物件；應用擴張和侵蝕來找到候選網球影像。最後，為了過濾錯誤的候選影像，使用類神經網絡進行真球和假球分類，以最高機率者為最終的答案。

3. 資料集

資料集的第一部分是基於 2017 年夏季世大運網球男子單打決賽的廣播影片，該影片可在 YouTube [7]上獲得，解析度為 1280×720，幀速率為 30 fps，總長度約為 75 分鐘。該影片在資料集中命名為 TennisVideo.mp4，大小為 1.06 GB。在移除非比賽影像（例如僅拍攝觀眾）之後，從視頻中分割出 81 個剪輯，每個剪輯是記錄從發球開始直到死球的過程。每個剪輯的資料被儲存在名為 Clip1，Clip2 等的單獨文件夾中，被剪輯的影片名稱為 Clip1.mp4，Clip2.mp4 等。除了被剪輯的影片之外，還有從影片中所存下的圖片檔案，名為 0000.jpg，0001.jpg。最重要的是，名為 Label.csv 的標籤文件也保存在文件夾中。總共有 20844 幀被檢視和標記，每一幀最多只能有一個網球被標記。

標籤文件中的每一行都是一個幀，由“檔名”（簡稱 FN）、“可見性”（簡稱 VC）、“x 坐標”（簡稱 X）、“y 坐標”（簡稱 Y），和“軌跡模式”（簡稱 TP）所組成。Table I 是標籤文件中的一部份。 FN 是圖檔檔名，以下將介紹其餘屬性的詳細資訊。

…
0008.jpg,2,727,447,0 0009.jpg,1,735,457,0 0010.jpg,1,722,433,1 0011.jpg,1,707,403,0
…
0029.jpg,1,555,220,0 0030.jpg,1,550,218,2 0031.jpg,1,547,206,0
…

Table I. 標籤文件的一小部分

VC 屬性指的是網球在圖像中的可見性。可能的值為 0、1、2，和 3。第一類（標籤為 0）表示球不在影像中。第二類（標籤為 1）暗示可以從圖像中輕易地識別球。第三類（標籤為 2）暗示球在前景中但無法從圖像中輕易地識別。第四類(標籤為 3)暗示球被前景中的某些東西遮擋。其餘屬性僅適用於 VC 屬性為 1、2，或 3 的幀。對於第 2、第 3，和第 4 類，網球的位置將被記錄於圖像中。在資料集中，VC0、1、2、3 的幀數分別為 659、18035、2143，和 7。

X 和 Y 屬性是圖像中的球位置。由於高移動速度，廣播影片中的網球圖像可能模糊不清，甚至有餘像痕跡。在此情況下，X 和 Y 是於餘像痕跡的最後一點。例如，在 Fig.5 中，球從 Player1 往 Player2 的方向飛行，的球圖像被延長了，此時球被定位於紅點處。Fig.6 是另一種情況，無法從圖像中輕易地識別網球。圖 6（a）、（b），和（c）是分別被命名為 0078.jpg、0079.jpg，和 0080.jpg 的三個連續圖像。在 0079.jpg 中，由於網球的顏色與球場上“台北”文字的顏色相似，因此難以觀察到球。 然而，藉由 0078.jpg 和 0080.jpg 的幫助之下，可以估計球在 0079.jpg 中的位置。圖 6（d）、（e），和（f）為標記結果。圖 6 是球被遮擋的情況。圖 7（a）、（b），和（c）是分別被命名為 0138.jpg，0139.jpg 和 0140.jpg 的三個連續圖像。在 0139.jpg 中，由於球被球員遮擋，無法看到球。圖 7（d），（e）和（f）為標記結果。

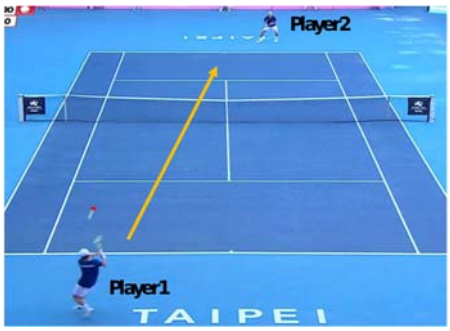

Fig. 5. 標記網球位置的時，網球呈現軌跡拖延的案例

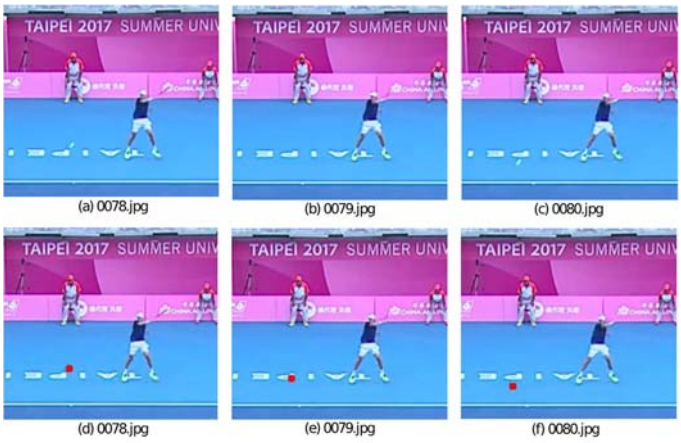

Fig. 6. 網球影像無法被識別

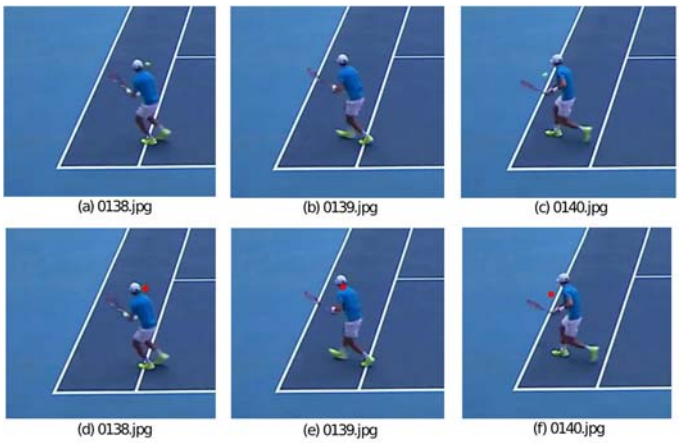

Fig. 7. 網球被遮擋

TP 屬性表示球的物理狀態並被分類為三類：飛行、命中，和彈跳，分別標記為 0、1，和 2，各有 19241、462，及 509 幀。在 Fig.8 中，三個連續圖像顯示了彈跳情況。0007.jpg 和 0008.jpg 是球與地面碰撞之前，所以被標記為飛行，而碰撞後或碰撞期間的 0009.jpg 則被標記為彈跳。在 Fig.9 中，三個連續圖像記錄了網球被命中。命中前的 0021.jpg 和 0022.jpg 被標記為飛行，而命中之後或命中期間的 0023.jpg 被標記為命中。

Fig. 8. 彈跳案例：（a）和（b）標記為飛行，（c）標記為彈跳。

Fig. 9. 命中案例：（a）和（b）被標記為飛行，（c）被標記為被命中。

4. 網球追蹤網絡

基於熱圖的 CNN 在解決問題方面已被證實是成功的[23] [24]。本研究提出了一種稱為 TrackNet 的卷積神經網絡（CNN）框架，它將幾個連續的幀一起作為輸入，以生成用於網球偵測的熱圖。連續幀的數量是網絡設計中的參數。如果網絡僅使用一幀來生成偵測熱圖，則此網絡可視為經典的圖形識別 CNN。另一方面，如果使用多於一個圖像來生成熱圖，則可以通過利用球

的外觀和軌跡來提高偵測的效能。本研究實作了兩個網絡，一個網絡只使用一幀來生成熱圖，另一個網絡則使用連續三幀。

訓練後的 CNN 可生成許多具有與輸入圖像相同大小的機率熱圖，而這些熱圖的答案（Ground Truth）是位於網球中心的 2D 高斯分佈圖形。球的中心位置可在資料集中取得，而高斯分佈的方差（Variance）則是網球圖像的大小。如果(x_0, y_0)是球心，則熱圖函數可由下式得出

$$G(x,y) = \left\lfloor \left(\frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}} \right) (2\pi \cdot \sigma^2 \cdot 255) \right\rfloor$$

其中第一部分是以(x_0, y_0)為中心的高斯分佈，具有方差 σ。而第二部分將值縮放到範圍$[0, 255]$。根據網球平均半徑約為 5 個像素（$G(x,y)$的半徑約 ≥ 128）的原因，我們使用$\sigma^2 = 10$。Fig.10 是網球的視覺化熱圖函數。

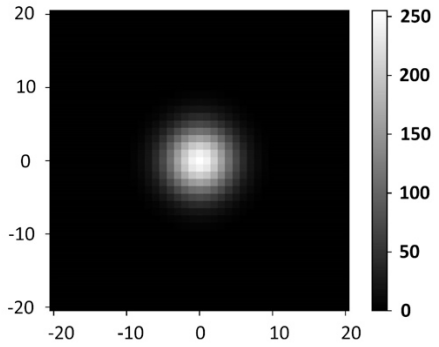

Fig. 10. 偵測熱圖的範例

本研究逖提出的 TrackNet 如 Fig.11 所示，逐層的組態設計呈現於 Table II 中。網絡的輸入可以是數個連續的影片幀。前 13 層如同 VGG16 [6]的前 13 層設計，用來對物體做分類，而 14-24 層則是根據 DeconvNet [25]。為了進行逐像素的預測，使用 Upsampling 來恢復由 Pooling 所引起的 Downsampling 效果。在所提出的網絡中可以看到同等數量的 Upsampling 層及 Pooling 層。

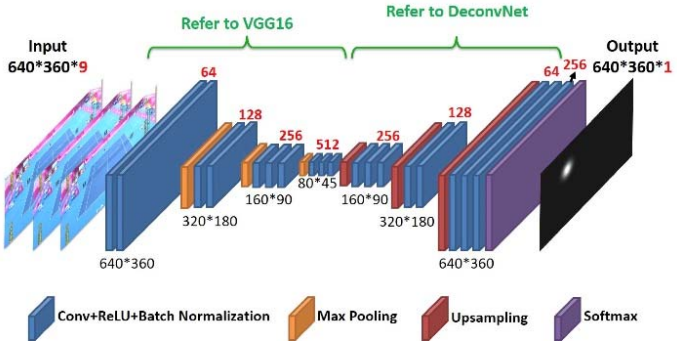

Fig. 11. 所提出的 TrackNet 架構

Layer	Filter Size	Depth	Padding	Stride	Activation
Conv1	3×3	64	2	1	ReLU+BN
Conv2	3×3	64	2	1	ReLU+BN

Pool1	2 × 2 max pooling and Stride = 2				
Conv3	3×3	128	2	1	ReLU+BN
Conv4	3×3	128	2	1	ReLU+BN
Pool2	2 × 2 max pooling and Stride = 2				
Conv5	3×3	256	2	1	ReLU+BN
Conv6	3×3	256	2	1	ReLU+BN
Conv7	3×3	256	2	1	ReLU+BN
Pool3	2 × 2 max pooling and Stride = 2				
Conv8	3×3	512	2	1	ReLU+BN
Conv9	3×3	512	2	1	ReLU+BN
Conv10	3×3	512	2	1	ReLU+BN
UpS1	2 × 2 upsampling				
Conv11	3×3	512	2	1	ReLU+BN
Conv12	3×3	512	2	1	ReLU+BN
Conv13	3×3	512	2	1	ReLU+BN
UpS2	2 × 2 upsampling				
Conv14	3×3	128	2	1	ReLU+BN
Conv15	3×3	128	2	1	ReLU+BN
UpS3	2 × 2 upsampling				
Conv16	3×3	64	2	1	ReLU+BN
Conv17	3×3	64	2	1	ReLU+BN
Conv18	3×3	256	2	1	ReLU+BN
Softmax					

Table II. TrackNet 網絡之參數

熱圖不會直接用於損失函數的計算，因此不被視為深度卷積網絡的一部分。在 Softmax 的前一層（倒數第二層）大小和深度分別為 640×360 和 256。此大小與輸入圖像的大小相同，而深度對應於灰階 0-255。設$L(i，j，k)$表示數據陣列，其中像素值為(0,0) ≤ (i，j) ≤ (639,359)，深度指數為$0 \leq k \leq 255$。通過 Softmax 函數逐像素地生成也具有相同大小和深度的最後一層，以將每個像素正規化至 256 個數值，使其成為灰階上的概率分佈。令P(i，j，k)表示灰階 k 的於位置(i，j)處的機率值。Softmax 函數由下式得出

$$P(i,j,k) = \frac{e^{L(i,j,k)}}{\sum_{l=0}^{255} e^{L(i,j,l)}}$$

熱圖由$P(i，j，k)$得出。而位置$(i，j)$，令

$$h(i,j) = \arg\max_{k} P(i,j,k)$$

為熱圖在位置$(i，j)$的值。換句話說，熱圖上的值是為具有最高機率的灰階等級。

在訓練階段，Cross-Entropy 函數被用來根據P(i，j，k)計算損失函數。相應的 Ground Truth 函數由Q(i，j，k)表示，並由下式表示

$$Q(i，j，k) = \begin{cases} 1, & if\ G(i,j) = k; \\ 0, & otherwise \end{cases}$$

令$H_Q(P)$代表損失函數，則

$$H_Q(P) = -\sum_{i,j,k} Q(i,j,k) \log P(i,j,k)$$

若提供偵測熱圖，則可以如以下確定網球的位置。首先，熱圖通過閾值（Threshold）轉換為黑白二位元圖。如果熱圖上的值大於閾值，則輸出設置為 255。否則，輸出設置為 0。本研究中，閾值設定為 127。然後，名為 Hough Gradient Method [26]的圓形查找演算法則被用於查找每個點的中心。若只有一個圓被查找到，則輸出圓的中心。在其他情況下，圖像被認為沒有網球。

5. 效能評估

在我們的實驗中，訓練集由數據集中的 70%圖像組成，剩下的則用於測試集。????

本實驗中所使用的深度學習伺服器配備了一個 Titan X GPU，一個 Intel Core i5-7500 CPU 和 32G RAM。作業系統是 ubuntu 16.04 LTS，深度學習使用的 API 是 Keras [27]。Adadelta 優化器 [28]被用於更新權重。資料集中的原始圖像解析度為 1280×720，但被重新調整為 640×360，以減少對系統資源的需求並加快處理速度。其他在訓練階段的關鍵參數表列於 Table III。

Parameters	Setting
Learning rate	1.0
Batch size	2
Steps per epochs	200
epochs	500
Initial weights	random uniform
Range of initial weights	[−0.05, 0.05]

Table III. 用於訓練網絡的關鍵參數

將資料集隨機分為兩組，70%的影像用於訓練網絡，剩餘的 30%影像用於測試網絡性能。我們提出了兩種版本的 TrackNet。模型 I 僅使用單一影像作為輸入，而模型 II 將三個連續影像作為輸入。在模型 II 中，三個連續影像被用來偵測最後一幀中球的位置。在模型 II 網絡的訓練中，只有當最後一個影像是來自訓練資料集時，才將三個連續影像視為訓練實例。另一方面，若最後一個影像來自測試資料集，則無論前兩個幀來自訓練集還是測試集，三個圖像都被視為測試實例。在訓練階段，訓練模型 I 需要大約 50 小時 11 分鐘，而訓練模型 II 大約需要 50 小時 58 分鐘。在 Inference Stage 時，模型 I 需要 0.7824 秒來處理一幀，而模型 II 需要 0.7936 秒來處理一幀。由於兩個模型的差別只在於第一層，模型 II 並不需要付出過多的成本來獲得改進。本研究所提出的架構俱有一次處理更多影像的潛力。

首先，定位誤差（Positioning Error），在下面的討論中將其表示為 PE。由歐幾里德距離所測量的 PE 的分佈如 Fig.12 所示。x 軸為像素單位，y 軸表示機率。綠色實線和紅色虛線分別表

示模型 I 和模型 II 的分佈。在 x＝0 處是精確偵測的百分比。在 x＝1 處表示的機率是0 ＜PE ≤ 1 的累積機率，在 x＝2 處表示的機率是1 ＜PE ≤ 2的累積機率，以此類推。平均來說，模型 I 和模型 II 的定位誤差分別是 2.01 像素和 1.32 像素。請注意到模型 I 和模型 II 在 PE＞5 時的累積機率分別為 4.74%和 0.18%。換句話說，模型 I 和模型 II 分別有 95.26%和 99.82%的偵測是 PE 不超過 5 個像素的。影片中大多數網球影像的直徑在 2 到 12 個像素之間，而 5 個像素大致是網球的平均半徑。因此，選擇 5 個像素作為判斷的基準，以決定球是否被正確識別和定位。更嚴格地說，在以下分析中，如果 PE 大於 5 個像素，則將被視為錯誤的偵測。

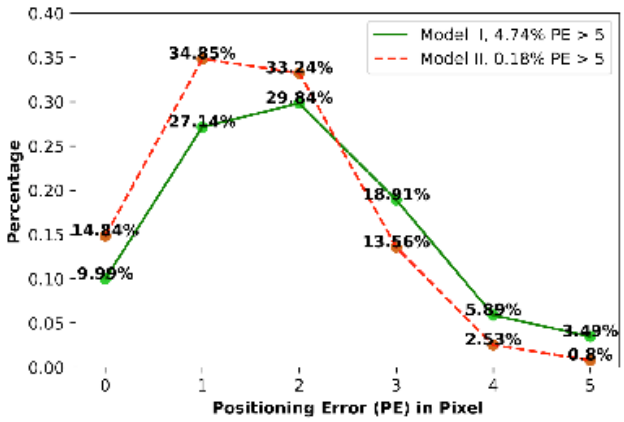

Fig. 12. 定位誤差分佈：綠線是模型 I，紅線是模型 II

按照球的可見性（VC）來分組的預測結果如 Table IV，V 和 VI 所示。請注意到如果在圖像中偵測到多個球，則將其視為失敗並歸類為 Negative。為方便起見，VC 是以下討論中“可見性分類”的簡寫。 VC1，VC2 和 VC3 的“False Positive”表示 PE 大於 5 個像素。 VC1，VC2，VC3 的“False Negative”表示沒有偵測到球或偵測到一個以上的球。表中的“-”表示該組合永遠不會發生。由於訓練資料和測試資料是針對模型 I 和模型 II 獨立選擇的，因此在測試時所使用的影像數量並不完全相同。為了與傳統的影像處理技術進行比較，我們也實作了[1]中由 Archana 和 Geetha 所提出的方法。Table IV，V，VI 分別是 Archana，Model I 和 Model II 的結果。

	VC0	VC1	VC2	VC3
True Positive	-	4046	418	0
False Positive	201	334	29	1
True Negative	9	-	-	-
False Negative	-	947	214	6
Total	210	5327	661	7

Table IV. 使用 Archana 方法之效能

	VC0	VC1	VC2	VC3
True Positive	-	4909	496	0

False Positive	0	247	22	0
True Negative	196	-	-	-
False Negative	-	239	138	7
Total	196	5395	656	7

Table V. 模型 I 之效能

	VC0	VC1	VC2	VC3
True Positive	-	5204	559	2
False Positive	0	6	5	0
True Negative	210	-	-	-
False Negative	-	117	97	5
Total	210	5327	661	7

Table VI. 模型 II 之效能

我們可以輕易地看出 TrackNet 無論是模型 I 還是模型 II 在所有方面都比傳統作法更好。模型 I 和模型 II 中的“False Positive”和“False Negative”案例顯著地被減少，而“True Positive”和“True Negative”案例平均增加 20.9%。更有趣的是，在模型 II 中，7 個球被遮擋的案例中有 2 個甚至可以被識別。此結果顯示，利用連續圖像可以提高性能，網絡不僅可以從模糊圖像中偵測球，還可以找出被遮擋的球。通過 Table VII 中給出的 Precision、Recall，和 F1-measure 來評估整體效能。這三個指標由以下公式計算

$$Precision = \frac{\#\ of\ \ true\ \ positive}{\#\ of\ \ true/false\ \ positive}$$

$$Recall = \frac{\#\ of\ \ true\ \ positive}{\#\ of\ \ VC1 + VC2 + VC3}$$

$$F1 = \frac{2(Precision \cdot Recall)}{Precision + Recall}$$

如 Table VII 所示，TrackNet II 的 Precision 及 Recall 可分別高達 99.8%和 96.1%，而 TrackNet I 達到 95.2%的 Precision 及 89.2%的 Recall。與傳統的作法相比，所提出的 TrackNet 具有顯著的進步。

Mothods	Precision	Recall	F1-measure
Archana’s [1]	88.7%	74.4%	80.9%
TrackNet Model I	95.2%	89.2%	92.1%
TrackNet Model II	99.8%	96.1%	97.9%
TrackNet Model II’	99.5%	96.9%	98.1%

Table VII. 三個偵測演算法的準確度指標

過度擬合（Overfitting）的問題存在於大多數的深度學習網絡，實際上，如果模型僅被來自單一來源的資料集訓練，就可能會發生過度擬合的情況。由於 TrackNet 的表現如此之好，過度擬合的問題令人擔心。因此，我們標記了來自另外 8 個影片的 74 個剪輯，將總共 16118 幀的資料添加到訓練資料集中。由豐富資料集所訓練的模型，稱之為模型 II'。Table VIII 中給出了模型 II'的結果，並且 Precision、Recall，和 F1-measure 仍分別高達 99.5%，96.9%和 98.1%。此結果如 Table VII 所示。

	VC0	VC1	VC2	VC3
True Positive	-	5220	590	1
False Positive	4	8	10	2
True Negative	206	-	-	-
False Negative	-	99	61	4
Total	210	5327	661	7

Table VIII. 模型 II'之效能

6. 結論

低畫質影片的網球追蹤任務在球尺寸小且球速高的情況下極具有挑戰性，廣播影片中可能模糊且不清楚，通常會需要高幀率和高解析度相機。在本文中，我們認識到深度學習的力量，並呈現了一個有效地從廣播影片中偵測網球的方法。以熱圖基礎的深度學習網絡被設計用來從模糊圖像中追蹤網球。此外，此方法利用多個連續幀來增強識別和偵測性能。本研究實作了兩個卷積神經網絡及一個傳統方法。與俱有 88.7% Precision 和 74.4% Recall 的傳統方法相比，使用連續三幀作為輸入的 TrackNet 達到了 99.8%的 Precision 和 96.1%的 Recall。此外，我們使用的資料集可由網路取得。據我們所知，這是第一個用於網球偵測的開放資料。

在所提出的方法中，我們也嘗試過使用二元影像當作 Ground Truth 來替換高斯熱圖影像，但實驗結果顯示並不比高斯熱圖好。此外，網球的軌跡也被嘗試分類為擊球，彈跳和移動，但實驗結果表現不佳。深度學習網絡比傳統方法有著更好的效能，但計算資源也相對昂貴。此外，由於資料集特性並不夠多元，造成了過度擬合的問題。在未來，我們希望透過更多網球比賽和其他場景來擴展資料集。此外，我們將嘗試整合軌跡資訊，以便於實現高水準的網球比賽分析。

致謝

參考文獻

[1] M. Archana and M. K. Geetha, “Object detection and tracking based on trajectory in broadcast tennis video,” Procedia Computer Science, vol. 58, pp. 225–232, 2015.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[3] R. Girshick, “Fast r-cnn,” arXiv preprint arXiv:1504.08083, 2015.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91–99.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[6] K.Simonyan and A.Zisserman,"Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[7] "The tennis men's singles final at the 2017 summer universiade," https://www.youtube.com/watch?v=fc50hOmbOuI, 29 August 2017, Taipei, Taiwan.

[8] H.-T. Chen, W.-J. Tsai, S.-Y. Lee, and J.-Y. Yu, "Ball tracking and 3D trajectory approximation with applications to tactics analysis from single-camera volleyball sequences," Multimedia Tools and Applications, vol. 60, no. 3, pp. 641–667, October 2012.

[9] X. Wang, V. Ablavsky, H. B. Shitrit, and P. Fua, "Take your eyes off the ball: Improving ball-tracking by focusing on team play," Computer Vision and Image Understanding, vol. 119, pp. 102–115, February 2014.

[10] T.-S. Fu, H.-T. Chen, C.-L. Chou, W.-J. Tsai, and S.-Y. Lee, "Screen-strategy analysis in broadcast basketball video using player tracking," in Processing of the 2011 IEEE Visual Communications and Image (VCIP), 6-9 November 2011.

[11] H. Myint, P. Wong, L. Dooley, and A. Hopgood, "Tracking a table tennis ball for umpiring purposes," in Proceedings of the 14th IAPR International Conference on Machine Vision Applications (MVA 2015), 18-22 May 2015, pp. 170–173.

[12] "Hawk-eye," https://en.wikipedia.org/wiki/Hawk-Eye.

[13] X. Yu, C.-H. Sim, J. R. Wang, and L. F. Cheong, "A trajectory-based ball detection and tracking algorithm in broadcast tennis video," in Image Processing, 2004. ICIP'04. 2004 International Conference on, vol. 2. IEEE, 2004, pp. 1049–1052.

[14] V. Reno`, N. Mosca, M. Nitti, C. Guaragnella, T. D'Orazio, and E. Stella, "Real-time tracking of a tennis ball by combining 3d data and domain knowledge," in *Technology and Innovation in Sports, Health and Wellbeing (TISHW), International Conference on*. IEEE, 2016, pp. 1–7.

[15] F. Yan, W. Christmas, and J. Kittler, "A tennis ball tracking algorithm for automatic annotation of tennis match," in British Machine Vision Conference (BMVC), vol. 2, 2005, pp. 619–628.

[16] X. Zhou, L. Xie, Q. Huang, S. J. Cox, and Y. Zhang, "Tennis ball tracking using a two-layered data association approach," IEEE Transactions on Multimedia, vol. 17, no. 2, pp. 145–156, 2015.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Image net classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.

[19] W. Jiang and Z. Yin, “Human activity recognition using wearable sensors by deep convolutional neural networks,” in Proceedings of the 23rd ACM international conference on Multimedia. ACM, 2015, pp. 1307–1310.

[20] Y. Chen and Y. Xue, “A deep learning approach to humanactivity recognition based on single accelerometer,” in 2015 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2015, pp. 1488–1492.

[21] V. Badrinarayanan, A. Handa, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling,” arXiv preprint arXiv:1505.07293, 2015.

[22] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

[23] V. Belagiannis and A. Zisserman, “Recurrent human pose estimation,” in 2017 12th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2017). IEEE, 2017, pp. 468–475.

[24] T. Pfister, J. Charles, and A. Zisserman, “Flowing convnets for human pose estimation in videos,” in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1913–1921.

[25] H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1520–1528.

[26] “Hough gradient method,” https://goo.gl/gZTQRm.

[27] “Keras: The python deep learning library,” https://keras.io.

[28] M. D. Zeiler, “Adadelta: an adaptive learning rate method,” arXiv preprint arXiv:1212.5701, 2012