Journal of Science and Engineering Technology, Vol. 15, No. 2, pp. 1~11 (2019)

基於深度學習與視覺之車輛偵測

黃登淵
大葉大學資訊工程學系
51591 彰化縣大村鄉學府路 168 號
kevin@mail.dyu.edu.tw

摘　要

本文提出植基於 Yolov3 架構之深度旋積神經網路以達成端到端之車輛偵測與追蹤。由於利用數量龐大的 ImageNet 影像資料庫重新訓練整個深度網路能提升的準確度相當有限，因此採用微調方法是較為可行的做法。配合微調方法，本文自製車輛資料庫，其中包含巴士、汽車及卡車等三種不同型態的車輛。在微調階段預訓練過程中，本文所提方法可以達成 98%的分類水平。在車輛偵測方面，本文採用 4 種不同情境的影片，包含有高速公路、巷弄街道、夜間車道與市區道路等，達成的偵測率分別為 78.8%、69.1%、86.2%及 88.1%。另外，在輸入影像解析度為 1920×1080 像素以及基於 CPU 執行的情況下，最終偵測速率每幀可達 420-470ms 之水準。最後，總體的平均偵測率與誤判率分別為 82.1%與 9.8%，足以驗證本文所提方法之可行性與有效性。

關鍵詞：深度學習，旋積神經網路，車輛分類，車輛偵測

Deep Learning and Vision-Based Vehicle Detection

DENG-YUAN HUANG
Department of Computer Science and Information Engineering, Da-Yeh University
No. 168, University Rd., Dacun, Changhua 51591, Taiwan, R. O. C.
kevin@mail.dyu.edu.tw

ABSTRACT

This paper proposes a deep convolutional neural network（DCNN）based on the YOLOv3 architecture to design an end-to-end vehicle detection and tracking system. It is considered feasible to use the fine-tuning method because the increase in detection accuracy of vehicles is quite limited even through retraining the entire DCNN using numerous images in the ImageNet dataset. To facilitate use of the fine-tuning method, we designed a custom database that includes three different types of vehicles, namely buses, cars, and trucks. In the pre-training phase of the fine-tuning method, the proposed method achieved a classification rate of 98%. In vehicle detection, we used four test videos with different scenarios, namely highways, alleyways, night lanes, and urban roads, to achieve detection rates of 78.8%, 69.1%, 86.2%, and 88.1%, respectively. The CPU only-based detection speed can reach 420–470 ms per frame when the input image size is 1920 × 1080 pixels. The overall average detection rate and false alarm rate for the four test videos was 82.1% and 9.8%, respectively,

which indicates the feasibility and effectiveness of the proposed method.

***Key Words*:** deep learning, deep convolutional neural network（DCNN）, vehicle classification, Vehicle detection

一、文獻探討

根據維基百科對深度學習的定義：機器學習的分支，透過多層、非線性變換（Multi-layer Nonlinear Transformation）對複雜資料集進行特徵擷取（Feature Extraction）。由於深層神經網路（Deep Neural Network, DNN）是實現多層、非線性變換最常用的一種方法，所以在實際應用中，都將深度學習視為深層神經網路的代名詞。由維基百科給出的定義可知，深度學習具有兩個非常重要的特性：多層與非線性，正由於這兩個特性，讓深度學習在電腦視覺、影像分類、物體偵測、語音識別、自然語言處理及機器翻譯等領域發光發熱，獲得巨大的成功。常見的深度學習架構有 DNN、深度信念網路（Deep Belief Network, DBN）、深度旋積神經網路（Deep Convolution Neural Network, DCNN）及遞迴神經網路（Recurrent Neural Network, RNN）等。

深度學習真正受到大量關注是在 2012 年。這一年，Hinton 研究團隊 [13]提出 DCNN 模型 AlexNet，利用 ImageNet 影像庫所提供的巨量資料，一舉將 ImageNet 大規模視覺辨識競賽（ImageNet Large Scale Visual Recognition Challenge, ILSVRC）中的「影像分類」任務的 Top5 錯誤率降低到 15.3%，而當年採用傳統方法的錯誤率還高達 26.2%，其僅比 2011 年降低了 2 個百分點而已。AlexNet 驚人的結果讓全世界研究者看到了深度學習的強大威力，因此 2013 年 ILSVRC 競賽成績在前面的隊伍幾乎都採用了深度學習的方法。當年度「影像分類」任務的冠軍為來自紐約大學 LeCun 的研究團隊 [26]，他們將 Top 5 錯誤率降到 11.7%的水準，所採用的模型亦是進一步優化的 DCNN。在 2014 年的 ILSVRC 競賽中，Google 團隊提出一具有 22 層的深度旋積網路 GoogLeNet，又將「影像分類」任務的 Top 5 錯誤率自前一年的 11.7%降低到 6.6% [24]。到了 2015 年，微軟亞洲研究院的何凱明等人 [10]提出了一個深達 152 層的 ResNet 模型，將影像分類的錯誤率下降到 3.6%的水平，此一水平已比人眼 5%的分類錯誤率更低，顯示人工智慧已超越人類的水平。在序列預測方面，RNN 在深度架構中扮演著非常重要的角色，相較於其它類型的神經網路，RNN 具備有狀態記憶之能力，使得它的應用更為寬廣。目前，RNN 已成功地被應用在機器翻譯、行人追蹤、語音識別、圖像說明生成，以及物體偵測 [3, 15, 22, 23, 25]等領域上。其中 Milan [15]等人採用 RNN 與 LSTM （Long Short Term Memory, 長短期記憶）對多行人目標進行偵測與追蹤，其結果顯示深度學習在物體偵測與追蹤上具有相當優異的能力。

傳統的目標偵測一般都是採用滑動窗口（Sliding Window）的方法，主要包含三個步驟：（1）利用不同尺寸的滑動窗口框住圖中的某一部分作為候選區域；（2）提取候選區域相關的視覺特徵，例如人臉偵測常用的 Harr 特徵 [1, 12, 21, 28]，以及行人偵測常用的 HOG（Histogram of Oriented Gradient） [2, 14, 20, 27]特徵等，最後（3）利用分類器進行識別，例如常用的 SVM （Support Vector Machine）模型 [11]。傳統的目標偵測中，多尺度變形部件模型（Deformable Part Model, DPM） [5]的效能在提出當時是相當出色的，曾經連續獲得 VOC（Visual Object Class）於 2007-2009 的目標偵測冠軍。2010 年時，DPM 作者 Felzenszwalb 更被 VOC 授予「終身成就獎」。DPM 的做法是將物體視為多部件（例如人的鼻子、嘴巴等），然後利用部件間的關係來描述物體，這個特性非常符合自然界很多物體的非剛體特徵。DPM 可以視為是 HOG+SVM 的擴展，很好的繼承了兩者的優點，在人臉偵測及行人偵測等任務上獲得相當不錯的結果，但 DPM 演算法相對複雜，偵測速度也較緩慢，為其缺點。

有鑑於 DPM 偵測緩慢的缺點，其後許多基於深度學習的目標偵測法陸續被提出來。在 2014 年時，Girshick 等人[7]提出植基於區域之旋積神經網路（Region-based Convolutional Neural Network, R-CNN），一舉將 PASCAL VOC 2007 測試集的 mAP（mean Average Precision） 自之前的 30%提升至 48%；2015 年再透過網路架構的修改，提出 Fast R-CNN，再將 mAP 自 48%提升至 66% [8]。其相關團隊在 2017 年又提出加速版 Faster R-CNN [19]，終於解決無法即時操作的困境。深度學習相關的目標偵測方法大致可以分成兩派，亦即：（1）基於區域提名（Region Proposal）的研究 [9]；（2）基於端到端（End-to-End），無需區域提名研

究等方法 [4]。就以目前來說，基於區域提名的方法較占上風，但端到端的方法則在速度上較占優勢，兩者的後續發展孰優孰劣仍有待時間的考驗。

目前基於區域提名的物體偵測方法中，準確度較高的還是以 R-CNN 及其後續改善版本（Fast R-CNN 與 Faster R-CNN）為代表 [7, 8, 19]，R-CNN 的做法是在影像中提取可能含有目標的候選框（Candidate Box），然後再將這些候選框輸入到 CNN 分類器，讓 CNN 分類器判斷候選框中是否含有目標物，以及目標物的類別是什麼。這種目標框標記的過程，基本上包含兩個部分，一是目標所在位置及大小，稱為定位（Localization）；其二則是目標識別（Recognition）。因此，在 R-CNN 整個算法中，目標位置與大小是包含在 Region Proposal 的過程中，而類別的判定則是依賴 CNN 分類器。至於 End-to-End 方法的典型代表就是 YOLO（You Look Only Once）及其後續改善版本 [16, 17, 18]，YOLO 基本上將物體偵測視為迴歸（Regression）問題，輸入影像，同時學習邊界框座標（Bounding Box Coordinates）及其對應類別機率。在 R-CNN 的方法中，CNN 的本質只是做為分類，基本上不涉及目標定位。而 YOLO 則是透過 CNN 網路直接實現目標定位與識別，亦即：自 CNN 網路輸入原始影像，直接輸出影像中所有的目標位置及目標類別。有鑑於 End-to-End 的效率性與方便性，本文採用基於 Yolov3 的網路架構並利用自製的車輛資料集（Custom Vehicle Dataset）來進行微調（Fine-tuning），進而訓練出自己的車輛偵測器。

二、研究方法

圖 1 為本文所提之系統架構，可分成：（1）影像前處理（Image Preprocessing），主要負責視訊影像框（Video Frame）的前處理，包含影像尺寸調整及像素灰階值的正規化等；（2）車輛偵測（Vehicle Detection），主要修改 Yolov3 的深度網路架構並利用自製的車輛資料庫來進行微調，以訓練出目標車輛偵測器。

（一）影像前處理（Image Preprocessing）

本文目標偵測器主要基於 Yolov3 的深度網路架構修改而提出的。由於 Yolov3 的影像輸入大小為 416×416 像素，而本文所用行車記錄器影像為 1920×1080 像素，示如圖 2（a），因此在應用 Yolov3 網路架構時，必須先進行影像大小調整。本文採用雙線性內插法（Bilinear Interpolation Method）[6]來進行影像縮小之處理，其結果示如圖 2（b），主要目的在於加速影像處理，以符合即時運算之目標。此外，為了降低光線變化對目標偵測結果之影響，在進行目標偵測之前，先進行影像強度正規化（Intensity Normalization），亦即對 RGB 各影像通道減去各通道之平均值，其結果示如圖 2（c）。

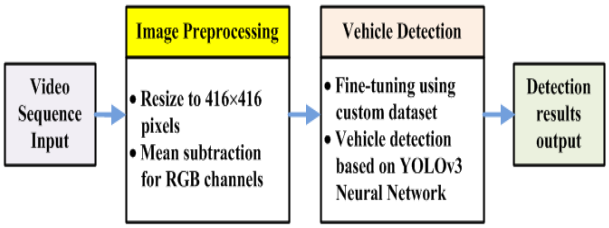

圖 1. 本文所提系統架構流程圖

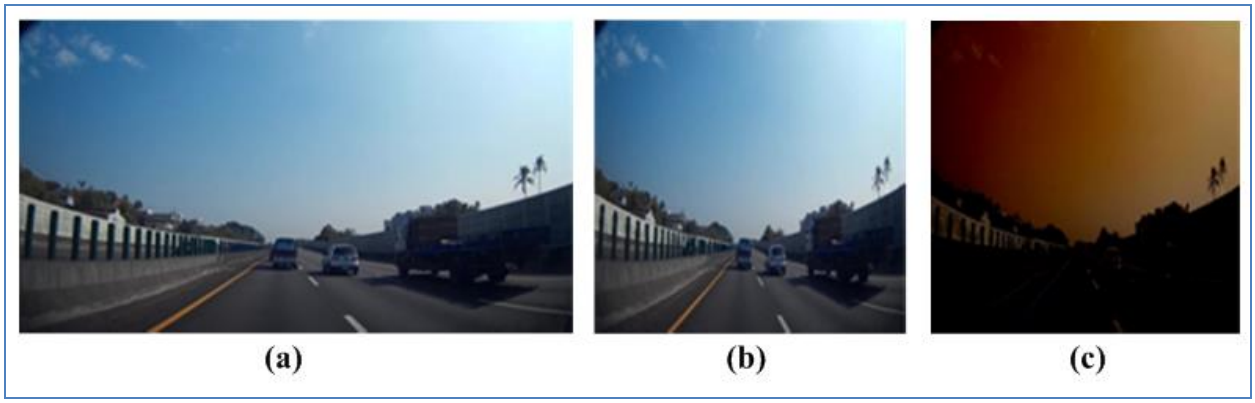

圖 2. 影像前處理之結果（a）1920×1080 之輸入影像，（b）416×416 之縮小影像，（c）針對影像（b）進行各通道去平均值處理

（二）車輛偵測（Vehicle Detection）

本文所提目標偵測器主要是基於 Yolov3 深度網路架構（見圖 3）修改而來，修改後的深度網路架構如圖 4 所示。由圖 3 可知，整個深度網路架構包含有 5 個 residual blocks，每個 residual block 均包含有 2 個旋積層以及 1 個 skip connection 架構，其中 skip connection 用來進行兩層間 element-wise 的相加，因此相加的兩層必須具有相同的尺寸與層數。另外，residual block×*n* 中的 *n* 表示該 residual block 架構連續重覆的次數，例如 residual block×8 表示該網路架構連續重覆 8 次。最後一個 skip connection 的輸出為（batch_size, 13, 13, 1024），再通過數個旋積運算後，最終輸出的張量（綠色方塊）為：（batch_size, 13, 13, 255）。至於另外兩個 skip connection 的輸出分別為：（batch_size, 26, 26, 512）與（batch_size, 52, 52, 256），這兩個輸出通道經過數個旋積運算後，再通過上採樣的過程（黃色方塊），最後上採樣結果再與原 skip connection 輸出進行 concatenation（亦即進行兩層間的堆疊，因此這兩層必須具有相同的尺寸，但層數可以不同），最終輸出的張量（綠色方塊）分別為：（batch_size, 26, 26, 255）與（batch_size, 52, 52, 255）；其中 255 可進一步表示為 3×（5+80），當中的 3 表示每個格點可預測 3 個 anchor boxes，5 表示 anchor boxes 的 5 個預測值（x_C, y_C, w, h, objectness），分別表示 anchor box 的中心點，寬，高以及是否包含有物體存在，80 則是表示預測的可能物體有 80 個類別。由於 Yolov3 的訓練是採用微軟的 COCO（Common Objects in Context）資料集，而該資料集的物體剛好就是 80 個類別。

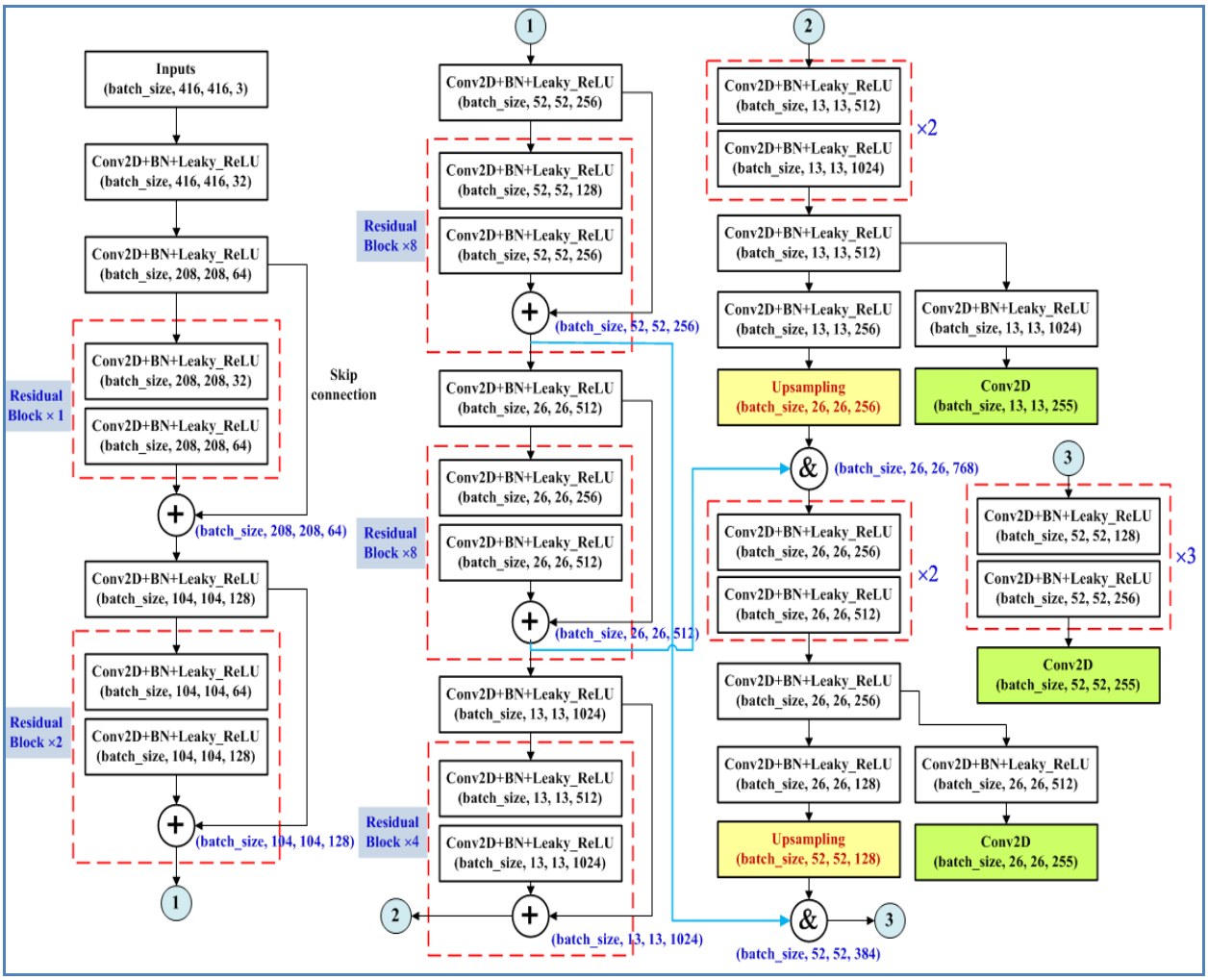

圖 3. 本文所提基於 Yolov3 深度網路架構修改之目標偵測器

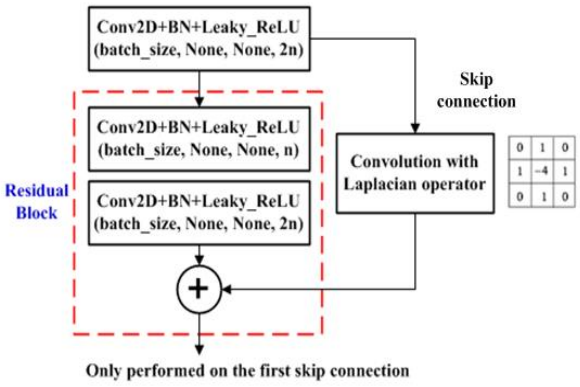

圖 4. 本文針對最後的 3 個 residual blocks（×8，×8，×4 等）進行修改，在每個 residual block 的第一條 skip connection 路徑中加入拉普拉斯旋積運算，以強化影像邊緣擷取的能力

Yolov3 為了可以偵測更小的物體，因此採用 FPN（Feature Pyramid Network）結構，其包含有 13×13 格點、26×26 格點與 52×52 格點。這種做法的目的在於克服影像金字塔效能不彰的問題。根據以上說明，整個網路架構預測的 anchor boxes 的數目為：13×13×3+26×26×3+52×52×3=10,647 個，基於以上概念，整個深度神經網路可以達到 end-to-end 物體偵測的目的。本文為強化影像邊緣擷取的能力，採用如圖 4 之修正架構，在最後 3 條 skip connection 的路徑中加入拉普拉斯運算子（Laplacian Operator）旋積運算，由於最後的 3 個 residual blocks 分別具有 8 條、8 條及 4 條 skip connection，但本文僅運用拉普拉斯旋積運算在每個 residual block 的第一個 skip connection 路徑上，其餘不採用。

1. 訓練誤差函數及網路微調（Fine-tuning）策略

物體偵測包含物體定位與辨識，因此在訓練過程中必須給予這兩個目標適當的懲罰項（Penalty），因此定義損失函數如式（1）所示。在（1）式中的最上面那兩列表示定位誤差（Localization Error），第三列表示邊界框（Bounding Box）是否含有物體（Objectness），最下面的那一列則是用來表示分類誤差（Classification Error）。在物體偵測過程中，目標影像通常被劃分成 $S\times S$ 個格點，每個格點負責偵測 B 個邊界框。由於大部分格點都沒有包含物體，因此可以知道不含物體邊界框的數量遠遠高於含物體的邊界框，基於平衡緣故，設權重 λ_{noobj} 為 0.5 及 λ_{coord} 為 5.0。在（1）式中，符號 $(x_i, y_i, w_i, h_i, C_i, p_i(c))$ 與 $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i, \hat{C}_i, \hat{p}_i(c))$ 分別表示真實與預測邊界框之中心點座標、寬、高、物體性（Objectness）及物體類別的機率等。其中 $I_{i,j}^{obj}$ 表示第 i 個格點的第 j 個邊界框含有物體，I_i^{obj} 表示第 i 個格點含有物體，$I_{i,j}^{noobj}$ 則是表示第 i 個格點的第 j 個邊界框不含有物體。

$$
\begin{aligned}
Loss = {} & \lambda_{coord}\sum_{i=0}^{S^2-1}\sum_{j=0}^{B-1} \mathrm{I}_{i,j}^{obj}\left[\left(x_i-\hat{x}_i\right)^2+\left(y_i-\hat{y}_i\right)^2\right] \\
& +\lambda_{coord}\sum_{i=0}^{S^2-1}\sum_{j=0}^{B-1} \mathrm{I}_{i,j}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right] \\
& +\sum_{i=0}^{S^2-1}\sum_{j=0}^{B-1} \mathrm{I}_{i,j}^{obj}\left(C_i-\hat{C}_i\right)^2+\lambda_{noobj}\sum_{i=0}^{S^2-1}\sum_{j=0}^{B-1} \mathrm{I}_{i,j}^{noobj}\left(C_i-\hat{C}_i\right)^2 \\
& +\sum_{i=0}^{S^2-1} \mathrm{I}_i^{obj}\sum_{c\in classes}\left(p_i(c)-\hat{p}_i(c)\right)^2
\end{aligned}
\qquad (1)
$$

由於利用 130 萬張的 ImageNet 影像資料庫來重新訓練整個深度神經網路將耗費數天時間，然後提升的準確度相當有限，因此可行的做法是利用自製的影像資料庫進行微調訓練。由於本文目的在於進行車輛偵測，因此自製的資料庫將以車輛為主，其主要包含巴士（Bus）（見圖 5（a）），卡車（Truck）（見圖 5（b））以及汽車（Car）（見圖 5（c）與（d））等，其中小轎車與休旅車均歸類於汽車中。表 1 為自製資料庫中巴士、汽車及卡車分別在訓練集與測試集中的數量分佈，其數量比約為 3:1。在微調訓練中，首先修改圖 3 中的 3 個輸出層（綠色方塊）分別為：（batch_size, 13, 13, 24）、（batch_size, 26, 26, 24）及（batch_size, 52, 52, 24），其中 24=3×（5+3），第一個 3 表示每個格點負責偵測 3 個邊界框，而括號內的 3 則表示有 3 個物體類別要偵測。整個預訓練過程中（訓練 20 萬次）的邊界框分類誤差、定位誤差及總體誤差變化示於圖 6 中。

表 1. 自製車輛資料庫中，巴士、汽車及卡車在訓練集與測試集中的數量分佈

	Bus	Car	Truck
訓練集	547	559	518
測試集	137	140	130
Total	684	699	648

圖 5. 本文訓練的車輛資料庫，包含（a）巴士（Bus），（b）卡車（Truck），（c）小轎車（Sedan）及（d）休旅車（SUV），其中（c）與（d）均歸類為汽車（Car）

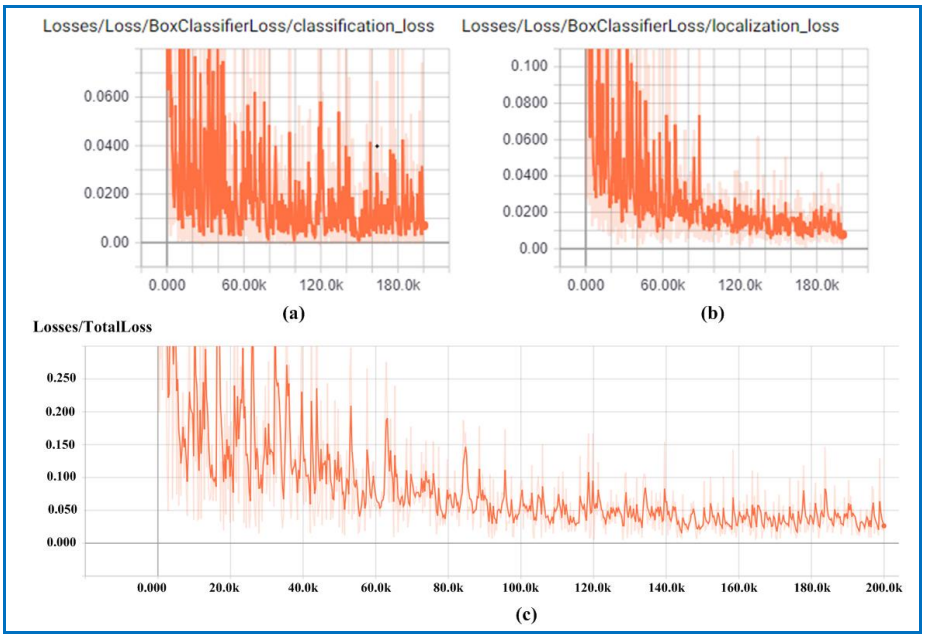

圖 6. 20 萬次的預訓練中，（a）bounding boxes 分類誤差，（b）bounding boxes 定位誤差，以及（c）總體訓練誤差

2. 邊界框的尺寸選擇

邊界框在 Yolov3 可視為 anchor boxes。本文採用 Yolov3 的 region proposal 方法，它不像 R-CNN [16]、Fast R-CNN [17] 及 Faster R-CNN [18]等文獻中所採用的方法，其假設 anchor boxes 的寬高比固定為 1:1，1:2 及 2:1，設定的邊長為 128 像素、256 像素與 512 像素等，因此總共有 9 種不同的組合。在本文中，每個輸出層的每個格點須負責 3 個邊界框的預測，由於有 3 個輸出層，因此總共有 9 個邊界框需要被預測。在本文中，我們採用 K-means 來尋找可能的 anchor boxes 的大小，其做法如下：首先蒐集 VOC2007 訓練資料庫

（http://host.robots.ox.ac.uk/pascal/VOC/voc2007/）中所有標注框的寬與高，並將之對原影像的寬與高進行正規化到介於 0 與 1 之間，其線性與對數分佈示於圖 7（a）與（b）中。緊接著，對所有正規化的寬高對（Normalized Width, Normalized Height）並任意選取九個點，再採用 K-means 進行反覆迭代運算，最終選取的質心示如圖 7（a）與（b）中的藍色點所示。在運用時，只要將這些正規化寬高點乘回原影像的寬高，即可得到先驗 anchor boxes 的大小。

3. 邊界框位置預測及非極大值抑制

Yolov3 預測邊界框中心點相對於所在格點（或稱 cell）左上角位置的相對偏移值，為了將邊界框中心點約束在當前的格點中，採用 sigmoid 函數處理偏移值，這樣預測的偏移值在（0,1）範圍內（因為每個格點的尺度可視為 1）。根據邊界框預測的 4 個 offsets（t_x, t_y, t_w, t_h），再按照圖 8 中所示的公式可計算出邊界框的實際位置與大小，其中（c_x, c_y）為所在格點的左上角坐標，p_w 和 p_h 為先驗邊界框的寬度與長度，其值由上一節的 K-means 方法估計得到。

若特徵圖的大小為（W, H），則邊界框的中心點與寬、高可由（2）式計算得到。另外，非極大值抑制（Non-Maximum Suppression; NMS）則是用來移除多餘的邊界框，當兩個邊界框重疊比例高於某一檻值時 （一般設為 0.5），表示兩個邊界框具有相同的物體，因此可移除其中的一個，用這種方法，可以控制每個邊界框僅有一個物體。

$$b_x = (\sigma(t_x) + c_x)/W;\ b_y = (\sigma(t_y) + c_y)/H$$
$$b_w = p_w e^{t_w}/W;\ b_h = p_h e^{t_h}/H \qquad (2)$$

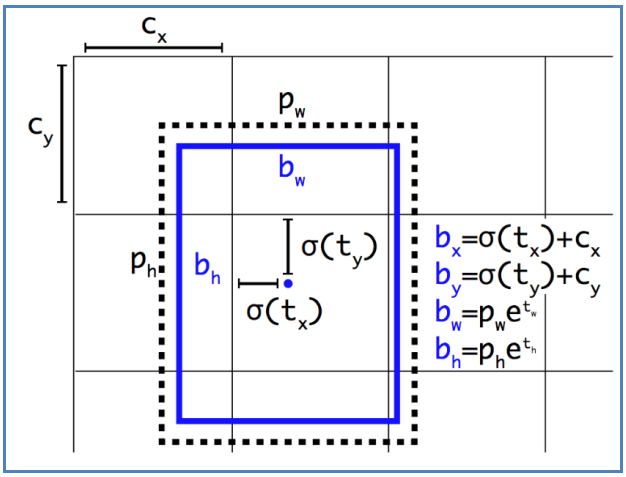

圖 8. Yolov3 預測邊界框示意圖（取自文獻[18]）

三、結果與討論

執行本實驗所用的作業系統（Operation System; OS）為 Linux Ubuntu 16.04。系統整合開發環境（Integrated Development Environment; IDE）是 Python 3.5 64 位元的版本，包含 Keras 2.1.6，Tensorflow 1.8.0 和 Opencv-python 3.4.1.15。本系統運行在 Intel i7-7770 處理器上，其具備有 16 GB 的記憶體，並配有 11GB 記憶體的 GTX 1080Ti 型號的 NVIDIA GPU 卡。本實驗所用圖片均蒐集自 Google Images。

本實驗訓練與測試所用資料庫的車輛圖片將分成巴士、汽車及卡車等三大類，其如圖 5 所示。所蒐集的巴士、汽車及卡車的圖片數量分別為 684、699 及 648 張。整個資料庫的 3/4 影像用來做為訓練，其餘的 1/4 則用做測試，其如表 1 所示。

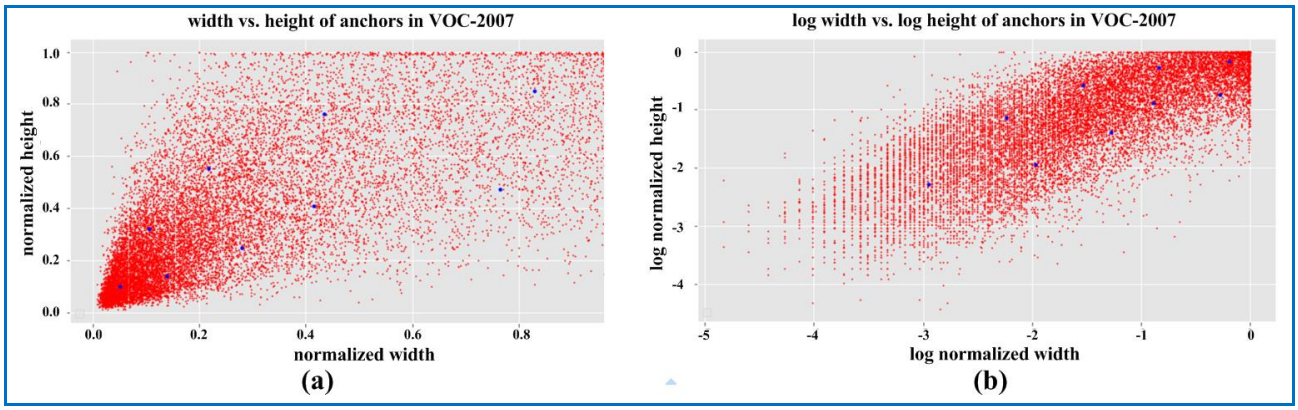

圖 7. 對於 VOC2007 訓練資料庫中所有的標注目標框，利用 K-means 聚類法預測 9 個 anchor boxes 最大可能的寬、高分佈，其如圖中藍色點所示：（a）線性表示所有標注目標框的寬、高分佈，（b）對數表示所有標注目標框的寬、高分佈

（一）旋積網路微調階段預訓練結果

在預訓練階段，本文用來評估所提方法影像分類性能，採用精準度（Precision），召回率（Recall）以及 f1-score 等度量值，各評估度量值定義如下：

$$precision = \frac{TP}{TP + FP} \tag{3}$$

$$recall = \frac{TP}{TP + FN} \tag{4}$$

$$f_1 - score = \frac{2}{1/\ precision + 1/\ recall} \tag{5}$$

其中 *TP*（True Positive）和 *FP*（False Positive）分別表示正確和錯誤地分類為所要分類的類別，而 *FN*（False Negative）表示識別為非所要分類的類別，但它卻是欲分類之類別。精準度也稱為正預測值（Positive Predicted Value; PPV），而召回率也稱做敏感度（Sensitivity）。至於 f1-score，它是精準度和召回率的倒數和的平均值。

在微調的預訓練階段，影像分類和它所對應的混淆矩陣（Confusion Matrix），其結果分別顯示在表 2 與表 3 中。從表 2 與表 3 的觀察中，本文所提的分類器可以準確地分類出巴士、汽車以及卡車等三種不同的車輛型態，其召回率整體可達 98%的水準。

表 2. 微調階段預訓練過程之影像分類結果

	Precision	Recall	f1-score
巴士	0.98	0.98	0.98
汽車	0.99	0.99	0.99
卡車	0.98	0.98	0.98
平均	0.98	0.98	0.98

表 3. 微調階段預訓練過程之影像分類混淆矩陣

	巴士	汽車	卡車
巴士	0.98	0.00	0.02
汽車	0.01	0.99	0.00
卡車	0.02	0.00	0.98

圖 9. 車輛偵測所用的測試影片，其解析度均為 1920×1080 像素，記錄速度為 30 幀／秒，這些影片分別為（a）高速公路，（b）巷弄街道，（c）夜間道路，以及（d）市區道路

（二）車輛偵測結果

本文主要是針對行車紀錄器所拍攝的畫面，對前方車輛進行偵測與追蹤。在本實驗中，行車紀錄器都是安裝在車輛擋風玻璃後靠近中央的位置。本次測試的影片拍攝情景包含有高速公路（TEST-01）、巷弄街道（TEST-02）、夜間道路（TEST-03）以及市區道路（TEST-04）等，其如圖 9 所示。在本次實驗中，每個測試影片都是 120 秒，影片的解析度均為 1920×1080 像素。

在車輛偵測評估方面，首先是視覺的主觀評量，其如圖 10 所示，在圖中採用綠色框表示汽車，粉紅色框表示卡車以及黃色框表示巴士等。由圖中可知，不同型態的車輛及不同的道路情境均可順利偵測到各種不同的車輛。此外，為了讓所提車輛偵測模型的評估更為客觀，本實驗採用偵測率（Detection Rate）與誤判率（False Alarm Rate）等兩個度量值（Metrics）來進行衡量，其定義分別示於式（6）與式（7）。

$$DR = \frac{TP}{GT} \times 100\% \qquad (6)$$

$$FR = \frac{FP}{TP + FP} \times 100\% \qquad (7)$$

其中 TP 與 FP 的定義如上一節所述，GT 為 Ground Truth，表示影像中所存在欲分類車輛類別的總數。因此偵測率的定義與召回率是相同的，而誤判率則等於 1-精準度。為了儘量排除人為計算車輛偵測率與誤判率的偏差，本實驗訂定以下幾個規則：（1）偵測的車輛太小，若其面積對整個影像的占比小於 5%，予以排除；（2）當車輛被其它物體遮蔽超過 50%時，予以剔除；（3）為了降低評估影片的負荷量，平均每 350ms 計算一次，因此每個測試影片約計算 340 張左右。透過以上方法，車輛偵測率與誤判率示於表 4，由表中可知高速公路上具有較高的誤判率，這是由於對向車道的車輛被車道中間隔欄部分遮蔽所致。此外，巷弄街道的偵測率較低，這是由於它的背景比其它情境更為複雜所致。最後，透過公式（6）與（7）亦可計算出 4 個測試影片的平均偵測率與誤判率，其值分別為 82.1%及 9.8%。

圖 10. 車輛偵測的結果，分別為（a）高速公路，（b）巷弄街道，（c）夜間道路，以及（d）市區道路

表 4. 車輛偵測各測試影片的偵測率（DR）與誤判率（FR）

	Bus			Car			Truck			DR	FR
	TP	FP	GT	TP	FP	GT	TP	FP	GT		
TEST-01 高速公路	0	7	0	446	99	567	164	116	207	78.8%	26.7%
TEST-02 巷弄街道	0	3	0	943	124	1263	101	37	247	69.1%	13.6%
TEST-03 夜間道路	0	0	0	2116	24	2419	0	2	37	86.2%	1.2%
TEST-04 市區道路	81	3	172	1653	140	1746	72	54	132	88.1%	9.8%
總計	81	13	172	5158	387	5995	337	209	623	82.1%	9.8%

四、結論

本文提出植基於 Yolov3 架構之深度旋積神經網路（DCNN）以達成端到端之車輛偵測與追蹤系統。由於利用數量龐大的 ImageNet 影像資料庫重新訓練整個 DCNN 所能提升的準確度相當有限，因此採用微調方法是較為可行的做法。配合微調方法，本文自製車輛資料庫，其中包含巴士、汽車及卡車等三種不同型態的車輛。在微調階段預訓練過程中，本文所提方法可以達成 98%的分類水平。在車輛偵測方面，本文採用 4 種不同情境的影片，包含有高速公路、巷弄街道、夜間車道與市區道路等，達成的偵測率分別為 78.8%、69.1%、86.2%及 88.1%。另外，在輸入影像解析度為 1920×1080 像素以及基於 CPU 執行的情況下，最終偵測速率每幀可達 420-470ms 之水準。最後，達成的總體平均偵測率與誤判率分別為 82.1%與 9.8%，足以驗證本文所提方法之可行性與有效性。

五、誌謝

本文承科技部計畫「應用深度學習於行車記錄器中車輛偵測與追蹤系統之研究，MOST 106-2221-E-212-013」補助完成，在此表達十分感謝之意。

參考文獻

1. Chang, W. C. and C. W. Cho (2008) Real-time side vehicle tracking using parts-based boosting. IEEE International Conference on Systems, Man and Cybernetics, Singapore.
2. Cheon M., M. Lee, C. Yoon and M. Park (2012) Vision-based vehicle detection system with consideration of the detecting location. *IEEE Transactions on Intelligent Transportation Systems*, 13(3), 1243-1252.
3. Chorowski, J., D. Bahdanau, K. Cho and Y. Bengio (2014) End-to-end continuous speech recognition using attention-based recurrent NN: First results. arXiv:1412.1602.
4. Dai, J., Y. Li, K. He and J. Sun (2016) R-FCN: object detection via region-based fully convolutional networks. arXiv:1605.06409v2.
5. Felzenszwalb, P. F., R. B. Girshick, D. McAllester, and D. Ramanan. (2010) Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 1627-1645.
6. Gonzalez, R. C. and R. E. Woods (2002) *Digital image processing*, 2nd Ed., 273-274. Prentice-Hall, Upper Saddle River, New Jersey, U.S.A.
7. Girshick, R., J. Donahue, T. Darrell and J. Malik (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, Ohio.
8. Girshick, R. (2015) Fast R-CNN. IEEE International Conference on Computer Vision (ICCV), Santiago, Chile.
9. He, K., X. Zhang, S. Ren and J. Sun (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. arXiv:1406.4729v4.
10. He, K., X. Zhang, S. Ren and J. Sun (2016) Deep residual learning for image recognition. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, USA.
11. Huang, D. Y., T. W. Lin, W. C. Hu and C. H. Cheng (2013) Gait Recognition Based on Gabor Wavelets and Modified Gait Energy Image for Human Identification. *Journal of Electronic Imaging*, 22(4), 043039, 1-11.
12. Kim, G. and J. S. Cho (2012) Vision-based vehicle detection and inter-vehicle distance estimation. IEEE International Conference on Control, Automation and Systems, Jeju Island, Korea.
13. Krizhevsky, Sutskever, I. and G. E. Hinton (2012) ImageNet classification with deep CNNs, Advances in neural information processing systems (NIPS), pp. 1097-1105.

14. Li, X. and X. Guo (2013) A HOG feature and SVM based method for forward vehicle detection with Single Camera. IEEE International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China.
15. Milan, Rezatofighi, S. H., A. Dick, K. Schindler and I. Reid (2016) Online multi-target tracking using recurrent neural networks. arXiv:1604.03635v1.
16. Redmon, J., S. Divvala, R. Girshick and A. Farhadi (2016) You only look once: unified, real-time object detection. arXiv:1506.02640v5.
17. Redmon, J. and A. Farhadi (2016) YOLO9000: Better, faster, stronger. arXiv:1612.08242v1.
18. Redmon, J. and A. Farhadi (2018) Yolov3: An incremental improvement. arXiv:1804.02767.
19. Ren, S., K. He, R. Girshick and J. Sun (2017) Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137 - 1149.
20. Satzoda, R. K. and M. M.Trivedi (2014) Efficient lane and vehicle detection with integrated synergies (ELVIS). IEEE International Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH.
21. Sivaraman, S. and M. M.Trivedi (2012) Real-time vehicle detection using parts at intersections. IEEE International Conference on Intelligent Transportation Systems, Anchorage, Alaska, USA.
22. Stewart, R. and M. Andriluka (2015) End-to-end people detection in crowded scenes. arXiv:1506.04878.
23. Sutskever, Vinyals, O. and Q. V. Le (2014) Sequence to sequence learning with neural networks. Advances in neural information processing systems (NIPS), pp. 3104-3112.
24. Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich (2015) Going deeper with convolutions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, Massachusetts.
25. Vinyals, Toshev, A., S. Bengio and D. Erhan (2015) Show and tell: A neural image caption generator. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, Massachusetts.
26. Wan, L., M. Zeiler , S. Zhang , Y. LeCun and R. Fergus (2013) Regularization of neural network using DropConnect. International Conference on Machine Learning, Atlanta, USA.
27. Wijnhoven, G. J. and H. N. deWith (2011) Unsupervised sub-categorization for object detection: Finding cars from a driving vehicle. IEEE International Conference on Computer Vision, Barcelona, Spain.
28. Yang, F. and H. Wang (2013) A driver assistance system based on mobile device. IEEE Global Congress Intelligent Systems, Hong Kong.

收件：**108.01.08**　修正：**108.03.22**　接受：**108.04.24**