國立台北大學資訊工程學系專題報告

智慧自駕系統
Intelligent Self-Driving System

專題組員:胡孝德、鍾淳良、陳彥儒、王品傑

專題編號:PRJ-NTPUCSIE-106-03
執行期間:106 年 09 月 至 107 年 05 月

一、 摘要

「智慧自駕系統」是現在尚未正式上路系統，希望透過此系統能夠比人類做出更及時、更精確的反應，來減少意外的發生，使得它的產生成為大家注目的焦點。在本專題中我們使用影像處理技術，判斷道路，透過深度學習，辨識路旁的交通號誌，在電腦端將資料彙整後，及時的傳遞訊息至自走車，使它能夠依照路況的不同，做出相對應的動作。隨著科技的進步，「智慧自駕系統」將會便利人類的生活，帶來更安全的交通環境，有效的降低社會成本。

二、 簡介

近年來，因為酒架與疲勞駕駛導致的事故頻傳，增加社會成本。在自動駕駛汽車裡，不須在意乘客是否達到必要的年齡、過老、無駕照、眼盲、精神不濟、酒醉等等。因為自動駕駛汽車不像人類駕駛一樣只有有限的環境感知能力，而可以使用主動與被動感測器持續做大範圍的感測，具有 360 度視野，因此可以對潛存危機做出安全的反應，且其反應較人類駕駛更為迅速。

為此，我們想打造一套自動駕駛系統，讓我們的汽車擁有跟人類駕駛一樣的行為，讓汽車自動保持在車道線內部，讓汽車自動遵守交通規則，像是紅燈停、綠燈行之類的，更要讓汽車不跟任何物體產生任何的碰撞。

三、 專題進行方式

(一) 系統設計

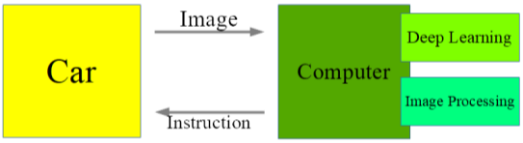


圖 0 系統架構

我們的系統架構如圖 0 所示，在我們的自走車將它的相機拍到的照片傳送到後端伺服器中做運算，運算一共有兩大類，第一類是使用影像處理找出車道線，另一類是使用深度學習偵測畫面中有無交通號誌，運算完之後將結果整合，決定車子該如何前進，是向前、還是向左又或者是向右，最後將結果回傳給車子，讓車子按照指令來做動作。

(二) 實作平台

自走車的部分主要使用兩個單晶片微控制器組合而成，如圖 1 所示，一個是樹莓派，搭配相機模組來拍照並透過網路將相片傳回後端；另一個是 Arduino，接收指令來控制伺服馬達前進，另外，它還連接超聲波模組來測量車子與物體之間的距離。

軟體的部分主要以 python 作

為主要程式語言開發，同時使用了兩大 library，一個是 OpenCV，跨平台的計算機視覺庫，另一個是 TensorFlow 用於各種感知和語言理解任務的機器學習。

在實驗環境的部分，我們使用 4 片瓦楞板加上膠布拼出車子行走的道路，如圖 2 所示；我們選定 4 種容易呈現出效果的交通號誌，有紅綠燈、快、慢與停，如圖 3 所示。快與慢用來告知車子現階段要用何種速度來行使，慢則是聽下來幾秒然後再繼續前進。

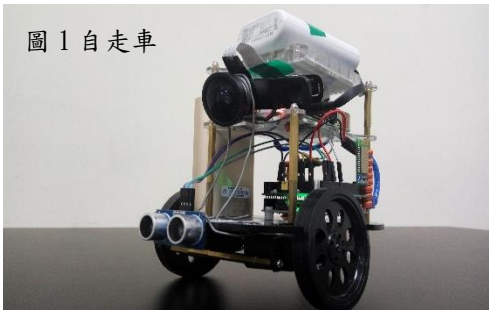


圖 1 自走車


圖 2 實驗道路

圖 3 交通號誌

(三) 實作技術

1、 影像處理

我們使用影像處理來找出車道線的位子，藉此得知是否該轉彎了還是繼續前進。整個流程，如圖 4 所示，可以分成 4 個步驟。

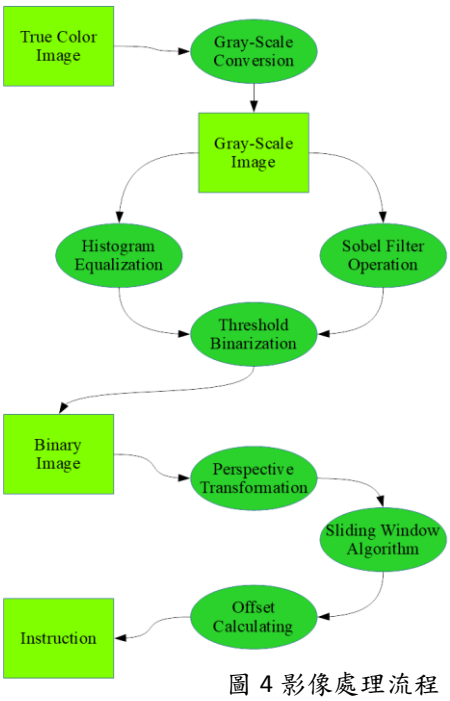


圖 4 影像處理流程


圖 5 黑白影像

由於道路是黑白的，首先我們將建立黑白影像，先將影像作灰階化的處理，然後將影像複製一份。針對第一份影像，使用 histogram equalization 來增加影像整體的對比度，尤其是當圖像的像素分布非常密集的時候，使用這種方法，將亮度密集的區間做拉伸的動作，我們之後才能更好地使用 threshold 值切出我們所需要的部分。針對第二份影像，使用 sobel 運算子來做邊緣檢測，它能用來畫出黑白之間的分界線，正好就是我們要的車道線邊界。最後將兩份影像作合併，就完成了我們第 1 步驟的黑白影像，如

圖 5 所示。

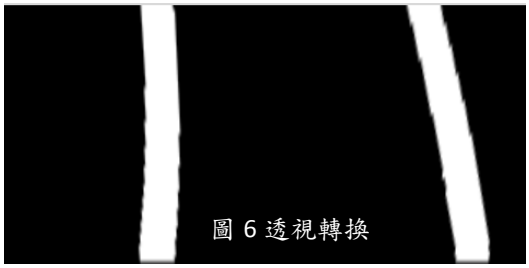


圖 6 透視轉換

單單只是轉成黑白影像還不夠，我們需要的只有影像下半部的兩條白色車道線，所以我們針對那部分用梯形圈出來，之後使用 perspective transform 將梯形的範圍垂直的翻上來，這樣一來，我們就將車道線以外的都過濾掉了，如圖 6 所示。

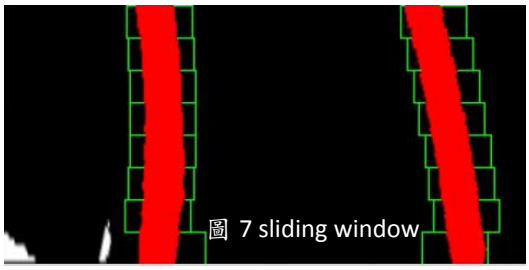


圖 7 sliding window

接下來，我們要針對此影像找出所有車道線的座標，使用 sliding window 的方式。我們統計每一個 column 的白色數量，兩邊各以最多的點為基準點，畫出一個小矩形，矩形範圍的白點就視為是車道線的一部分，將所有白色的座標存起來，然後再以那些白色點的 x 座標取平均，為上方新的矩形計算新的基準點，這樣一路往上找出所有的白色點，如圖七所示。

最後，我們要計算車道線中點，也就是兩車道線最下方的矩形基準點的中心點，對於畫面正中央的偏查值，藉由判斷偏差值所在的區間，我們可以得知車子的下一步該如何走。

2、 深度學習

目前的深度學習總體趨勢為通過更深與更複雜的神經網路來得到更高的精確度。考量到影像是即時性的，這種神經網路往往在模型的大小與運行速度上是不符合需求的。為此，我們使用的 model 是比較輕量級的 MobileNet，專們針對延遲的部分作優化，同時確保精確度是可接受的。此 model 適用於手機、嵌入式裝置，應用也較其他神經網路來的多。其整體架構如圖 8 所示。

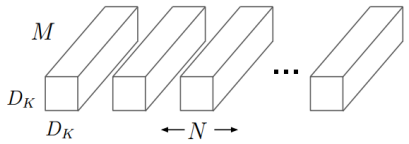
Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1 Conv / s1	$3 \times 3 \times 512$ dw $1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$ $14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

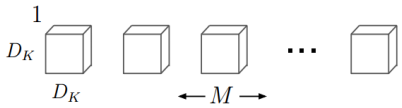
圖 8 MobileNet 架構

MobileNet 模型基於深度可分解卷積(depthwise separable convolutions)，它由分解後的卷積組成，也就是將標準卷積分解成一個深度卷積(depthwise convolution)和一個 1x1 的點卷積(pointwise convolution)。深度卷積將每個卷積核應用於每個 channel，然後深度卷積的輸出將作為點卷積的輸入，點卷積用

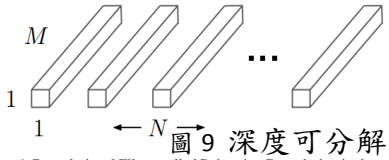
1x1 卷積來組合這些輸入，如圖 9 所示。藉由使用上述的分解，大幅地減少了計算量(8 到 9 倍)，並且準確率也不會下降太多，相較於其他的模型，極具有優勢。



(a) Standard Convolution Filters

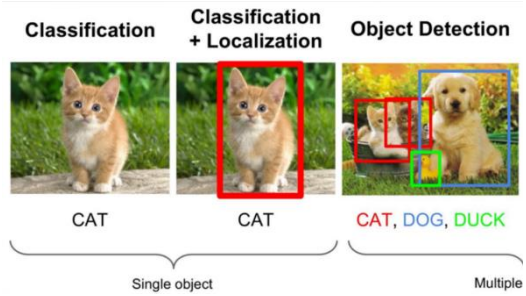


(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the con-

圖 9 深度可分解卷積



以往的物件偵測，大多都是使用分類器的方式來達成。針對某一類的照片，使用大量是這類與不是這類的照片來學習。但這種學習方式是不具有效率的，因為假如我今天要訓練一個動物識別偵測器，所以我要為每種動物建立分類器，然後每次再判斷的時候，要經過很多層的判斷才能給出答案。為此，我們基於 MobileNet，採用一種 SSMD 的物體檢測模型，如圖 10 所示，最大的優勢在於將檢測過程整合，也就是我只需一次性的判斷就能將所有圖片中的符合的都圈出來，便於訓練與優化，同時提高檢測速度。

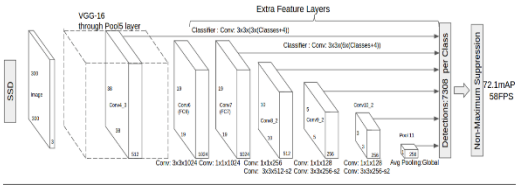
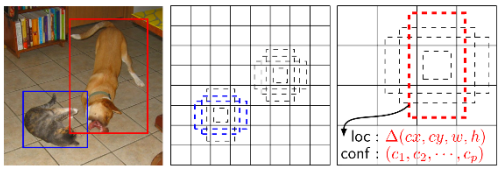


圖 10 SSMD

SSMD 的原理為針對每張的圖片，我們會對它們切割成不同尺度，如圖 11 中 8X8 與 4X4，以每個方框的中心點，圈出多個默認可能是物體的矩形範圍，如圖上面的虛線，再將每個矩形中提取出的特徵值與與每一個類別做比對，結果會得到許多介於 0-1 的值表示這矩形包含此類別的可能性，我們能自訂我們想要的界線，做出比傳統的分類器更加精準的識別。

圖 11 SSMD 原理



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

四、主要成果與評估

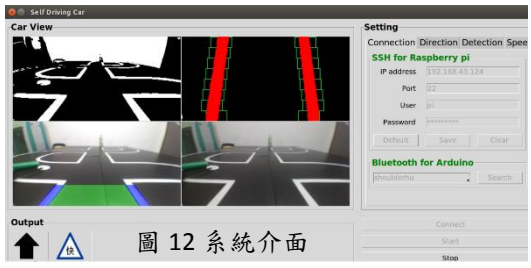


圖 12 系統介面

(一) 系統畫面

如圖 12 所示，整合了 Raspberry pi 與 Arduino 的連線、深度學習與影像處理的成

果、簡單圖示表達現在的路況、能動態地調整各種參數，操作起來容易。

(二) 交通號誌識別

圖 13 紅綠燈識別

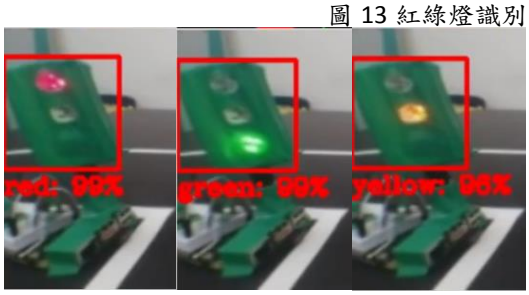
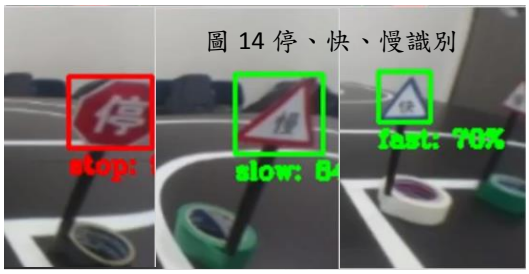


圖 14 停、快、慢識別



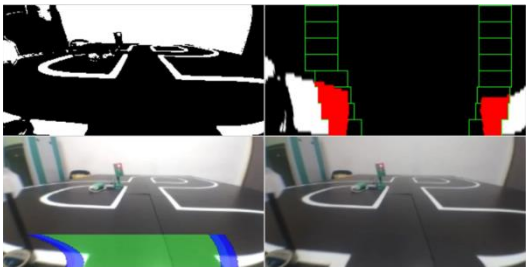
紅綠燈的識別，如圖 13 所示。我們不將三種燈號作為三類來處理，而是通通作為 1 類來識別，有鑑於他們之間的差異不大，要成功地分辨出來恐怕需要深度較深的神經網路。因此，在判別為紅綠燈之後，我們將矩形的範圍取出，上下分為三等份，分別過濾出紅色、綠色、黃色的像素，如果其中某個像素超過一定數量，我們就視為是某個燈號，否則就當作無燈號。

停的判別如圖 14 所示，看似很容易，只需要停下來就好了，但是這樣當看到停就會進入無窮的停止，我只是要停個幾秒，之後就繼續前進。所以，我們需要讓車子在準備前進時的一小段時間內，即使看到停也要不遵守，我們稱之為「無敵時間」。

快、慢的識別，如圖 14 所示。我們的車子具有兩種速度，而快與慢就是告知我們的車子該切換速度了。兩者的像似度較高，在區別時別具有挑戰性。

(三) 跟隨車道線系統

圖 14 十字路口

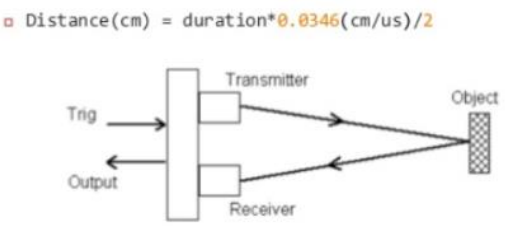


透過上述影像處理的四個步驟，即可快速地找出車道線，順便計算偏差量來通知車子如何前進。可是，當碰到十字路口的時候，便是一大問題了，如圖 14 所示。白色道路的數量急遽減少，尤其是在底部的白色都看不見時，恐怕會走歪，導致後續的判斷跟著錯誤。

我們提出的解決方法為紀錄 Sliding Window 的狀態，一共分成四種，白色點全部皆有、上面少了一些 window、下面少了一些 window、全部皆沒有。因為總共有兩條線，組合起來總共有 16 種可能。透過將狀態正確地做分類，就能夠安心地讓車子通過任何道路，不僅僅是十字路口而已。

(四) 防碰撞系統

圖 15 碰撞偵測



我們藉由超音波模組發出40KHz的超音波脈衝，量測脈衝遇障礙物反射回來的時間，來計算兩者之間的距離，如圖15所示。

之所以不採用物件偵測的做法是因為障礙物太多種了，如果沒辦法精確定義出每種障礙物，就沒辦法達到防碰撞的目的，而且訓練起來也是相當耗時的。而超音波模組使用超音波偵測速度較快，準確度也較高，實在是沒理由不選擇它。

五、 結語與展望

在本次的專題當中，我們做出了一個自動駕駛系統應具備的基本能力。首先是讓車子具有跟隨車道線行駛的能力，再來是讓車子具有防止碰撞的能力，最後是讓車子具有跟人類一樣的視覺能力。一但有了自駕系統後，就不會再有酒駕、疲勞駕駛或分心等原因所引發的交通事故了。

未來，如果還有機會的話，我們希望能夠在我們現有的交通號誌識別模型中加入更多的號誌，讓我們的自駕系統更強大、更完善。還有，我們希望能採用其他運算能力較強的硬體，將我們後端運算的部分整合到車子當中，不但能增加系統的反應速度，同時還增加系統的安全性。

六、 銘謝

感謝教授在這一年來的指導，同時也感謝組員們的付出。從一開始的題目決定、整體架構設定，一步一腳印的實作出我們的成果。在這過程當中，不免遇到許多的挫折，但是在圖書館的討論室中待上一個下午後，任何問題在我們面前總是有辦法可以解決的。

七、 參考文獻

[1] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolution neural networks for mobile vision applications. ArXiv preprint arXiv:1704.04861, 2017.

[2] W. Liu, D. Anguelov, D. Erhan, S. Christian, S. Reed, C.-Y. Fu and A.C. Berg. SSD: single shot multibox detector. In ECCV, 2016.

[3] Udacity Self-Driving Car Engineer Nanodegree. https://www.udacity.com/course/self-driving-car-engineer-nanodegree--nd013