

Final project-turtlebot3 路標辨識

Motivation/Background:

近年來隨著科技的發展，技術與硬體上的進步，使 AI 也迅速發展，如今 AI 已廣泛應用在許多的領域上，語音辨識、自動繪圖、機器人與自駕車等等，其中近年來滿熱門的一個研究項目就是自駕車，雖然目前有些車中已搭配自動駕駛的使用，但經常會出現關於使用自動駕駛出意外的新聞，表示這項技術目前仍不夠完善，有許多改善的空間。

要達成自動駕駛有許多的任務要同時進行，例如：車距判斷、紅綠燈與路標辨識、行人辨識、道路線路辨識等等，要完成一項任務就需要一定程度上的訓練並且要能夠準確地做出判斷，畢竟行車關乎到不只是自己的安全還有其他行車與行人的安全，可以看出自動駕駛實現的難度也同時顯示出市場上的競爭激烈，誰能夠先開發出安全的自動駕駛技術便是許多車廠的目標。

我們這次希望能透過這次的期末 project 使用 turtlebot3 來試著達成其中一項自駕車的任務，我們選定的是紅綠燈與路標的辨識，原本期末 project 就有在思考要做紅綠燈的偵測來做行動，而老師上課也提到關於 turtlebot3 的自駕車比賽以及在台灣也有相關的比賽，表示自駕車這塊領域是很多人注重的部分，更確定了我要做這方面相關的題目，後來有維嶺與我一組決定做更多一點把路標辨識也一起完成。

Related work

- TurtleBot3 AutoRace(driving)
- https://emanual.robotis.com/docs/en/platform/turtlebot3/autonomous_driving/
- <http://humanoid.nkust.edu.tw/RoboSports2022/>
- http://humanoid.nkust.edu.tw/RoboSports2022/document/2022/AutoRace_20220411.pdf

Problem definition

Turtlebot 主要的任務是要可以自動前進，並根據他偵測與辨識出的路標做出相對應的動作，我們分成路標辨識與紅綠燈顏色判斷兩個任務去執行，首先 turtlebot 先能夠判斷出有沒有路標以及路標的種類，再來根據每個不同的路標我們給定不同的動作去完成，以下為每個路標的圖片以及動作：



1. stop: 停止



2. speed limit: 減慢速度



3. crosswalk: 減慢速度到停止，並等待 5 秒再前進



4. 紅綠燈

其中如果 turtlebot 偵測出圖片是紅綠燈，會再根據不同顏色做出不同判斷，紅色為停止(因為停止就不會動了，因此這部分讓 turtlebot 停止後過一段時間再前進)、黃色為減速、綠色為前進。

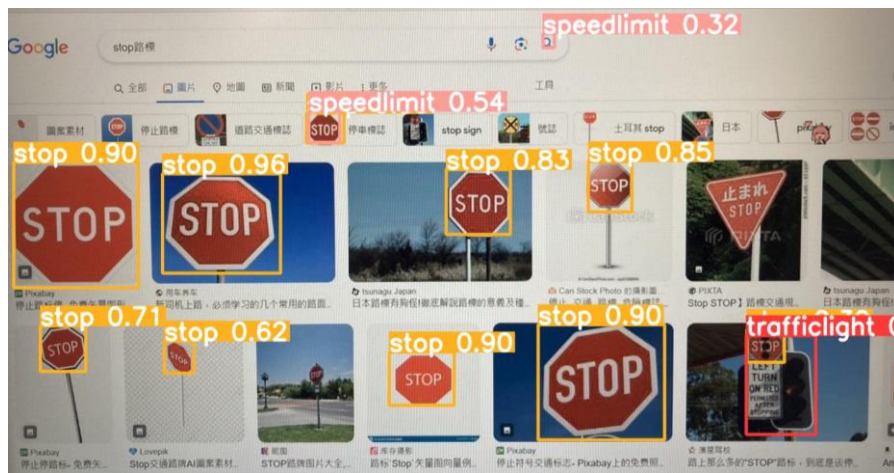
Technical details:

(1)Method

1. 路標偵測-yolov5:

要讓 turtlebot 可以來做偵測的任務，第一個想到的方法就是訓練一個模型去做判斷，而 yolo 就是我們的第一首選，之前其他課的期末報告就有使用過 yolov5 來做訓練，使用上的效果很好，因此這次也使用 yolov5 的模型來做訓練，在訓練上遇到的一個問題是 dataset 的格式，yolo 要求的標註檔案是一個 .txt 檔並且內容為一行例如 0/0.395917 0.479965 0.199642 0.123203 前面是這個圖片的分類，後面為標記的 xy 位置，之前訓練時都是直些使用 .txt 檔所以訓練起來沒問題，但這次的 dataset 給的是 .xml 的格式，因此還需要把這個格式轉換成 yolo 的要求格式，由於是第一碰到，所以在轉換的過程花了一點時間上來做處理。

處理好檔案後的步驟就比較簡單，因為 yolo 很多都幫忙寫好了，所以只需要將處理好的檔案路徑設置好，並執行命令就會開始訓練了，訓練完後先進行一些圖片的測試來確認訓練出的權重檔案可不可以使用，以下圖片是測試的結果圖，可以發現有滿不錯的結果就可以使用了。



2. 紅綠燈顏色判斷-resnet18

訓練完 yolo 後忘記要再做顏色的判斷，想到的時候覺得要再弄模型去訓練有點麻煩，所以想嘗試用不同的方法能不能做到一樣的效果，因為 opencv 本身可以做顏色的判斷，所以試著使用 opencv 的顏色偵測來做辨識，其中我試了以下幾種的方法：

- (1)經過 yolo 的偵測後會出現框，將框起來的部分當作一個照片，將照片的 RGB 轉換成 HSV，設置好 HSV 紅黃綠三種顏色的範圍，將圖片套進 CV2 去做辨識，這個方法是完全失敗且沒有成功，推測原因是因為光線或是被紅綠燈的黑色部分所影響所已造成在判斷上會讀取失敗。
- (2)沿用上面方法框起來的圖片，因為紅綠燈的顏色位置都是固定的，因此將圖片分成上、中、下三個部分，將三個部分都分開套進去 CV2 去做判斷，這個方法有成功，但可惜的部分是成功率沒有很高，原本以為在紅色和黃色的部分比較有可能會被搞混，但結果是在處理綠色和紅色辨識上會被搞混，黃色的部分成功率算是比較高的，由於成功率不高，因此還是放棄使用這個方法。
- (3)一樣使用被框起來的圖片，但這次不將圖片分割，而是將整個圖片拿去做辨識，先辨識出圖片最亮的像素，取用這個像素的顏色分量去做判定，因為大部分被框起來的圖片除了紅黃綠外基本上都是黑色，因此這個方法也是會成功，但前面也說到了是「大部分」，有些圖片還是會把一些背景一起框起來，這時紅黃綠就不一定會是最亮的顏色了，所以會影響判斷的結果，沒有辦法到有很高的成功率。

使用上面三種套用 CV2 都使用 10 張照片來做測試，而且是先以訓練 YOLO 的資料集中選擇 10 張，代表在框選上會有更好的表現，但使用這 10 張去做測試上面三種方法最高的也不超過 6 成，因此放棄使用上面三種方法，還是使用了 CNN 模型來做顏色辨識

(4)剛好有找到紅綠燈的資料集可以使用，因為這個資料集很小，再加上要做辨識的分類沒有很多，所以嘗試用一個比較小一點，訓練快一點的架構，最後選擇使用 resnet18 的架構來做訓練，過程也是如我所想訓練很快而且結果也滿不錯的，訓練出來的正確率大概 8、9 成，拿 10 張照片去做了測試大概有 8 成的成功率，成功率下降的主要原因我認為是因為 dataset 中的圖片幾乎都長的一樣，雖然看起來有多照片，但大部分的差異太小，所以在判斷上多少會有點影響，但我認為這已經算是還不錯的結果，所以就決定拿這個訓練出來的模型來負責顏色的判斷。

3. environment:

使用模擬器中的 turtlebot，在 gazebo 中建立幾個方塊，將要辨識的圖片貼在方塊上，使機器人的相機可以拍攝到這些圖片來做辨識。

system design

模型訓練好了之後首先寫個程式把兩個模型套用在一起來做測試，先讀取一個資料夾，因為 turtlebot 讀取的照片會一直做更新，所以資料夾內只放一張照片，程式會先去讀取資料夾的照片，放入 yolo 模型去做判斷，而如果輸出是紅綠燈，再放入 resnet 去做判斷。

測試成功後，在模擬器使用上發現 turtlebot 在讀取與存取圖片上會有時間差，造成判斷上的影響，因此改成使用 cv2 直接讀取圖片，不做存取圖片的動作，並加入使用在 ros 系統上的一些程式碼，創立一個可以讓 roscore 可以接收的 topic 並讓這個程式碼可以不斷執行，一直將讀取的圖片結果輸出給 roscore，再來寫一個程式負責去使用這個 topic，並利用接收到的 topic 內容做出相對應的動作，就可以實現 turtlebot 自主行動並根據偵測到的路標進行相對應的動作。

Dataset

使用 kaggle 上的兩組 dataset

路標:<https://www.kaggle.com/datasets/andrewmvd/road-sign-detection>

紅綠燈: <https://www.kaggle.com/datasets/sachsene/carla-traffic-lights-images>

Results and evaluation

最終 demo 影片中可以看出我們的模擬器中的 turtlebot 確實可以根據偵測到的圖片做出相對應且正確的行動，表示我們這次期末 project 想要實現的目的算是成功的，但其中也有很多可惜的部分和可以改善的問題存在：

1. 無法使用現實中的 turtlebot，我們要測試的時候發現 turtlebot 的相機因不明原因無法啟用，最後只能使用模擬器來完成這次的 project，等 turtlebot 修好再找機會看能不能把我們的程式碼成功使用在實體的 turtlebot 上。
2. 模擬器相機捕捉問題，從影片中可以發現 turtlebot 在還沒有穿過上一個圖片就在執行下一個圖片的行動，猜測是因為模擬器上相機的設定就是在 turtlebot 本身再前面一點做拍攝，因此才會出現這樣的問題。
3. 從影片中可以發現我們的 turtlebot 是穿過一張圖片去做下一張圖片的辨識，原本的想法是像平常開車一樣將這些圖片放在 turtlebot 的右側去做偵測，但主要有個問題是可能會把後面的圖片也一起偵測進去造成判斷上受到影響，因此最後只能選擇讓 turtlebot 一次對一張圖片做辨識，這部份我們還有很大的改善空間。

可能的方法是雖然一張圖片出現了兩個辨識類別，那就讓這兩個框起來的部份去比較大小，因為通常離的較近的圖片會比較大，較遠的圖片難判斷框會比較小，因此讓 turtlebot 只執行較大框的類別動作可能會是一個不錯的方法。

4. 當初在做辨識顏色的部分使用了多種的方法但沒有成功或是成功的機率不高，認為這些方法應該還有可以改進的地方，尤其是第一個方法，當初是看網路上很多使用 cv2 來做顏色的辨識，都是成功的所以打算來使用，也許是我的程式碼有寫錯導致影響了判斷，需要再多研究。

5. 模擬器中的紅綠燈是放三個紅、黃、綠燈的圖片，可以的話想試著讓這個圖片可以隨時間做更換來達到模擬真實紅綠燈的效果。

Reference

無

Answer to questions during the final presentation

上面有詳細描述關於 turtlebot 如何讀檔和模型如何使用

Work

品程:訓練兩組模型與測試、編寫模型使用程式碼(test.py)以及接收指令的離型架構(drive.py)

維嶺:負責建構模擬器環境使用、增加 ros 系統相關指令(test.py)與修正相關程式碼使其能在模擬上使用、負責修改動作指令使其能順利在模擬器上執行(drive.py)