

Finding Lane Lines

《自動駕駛實務》專案一

數據一 RE6121011 徐仁瓏

一、介紹

自動駕駛技術近年來取得了巨大的進步，其中車道辨識是實現自動駕駛的重要一環。隨著汽車產業對於自主駕駛技術的不斷追求和投入，車道辨識系統成為了實現車輛智能導航的核心技術之一。通過使用計算機視覺技術，車道辨識系統能夠準確地檢測和識別道路上的車道線，從而實現車輛的自主導航和行駛，大大提高了行駛安全性和行車舒適性。本專案旨在實現一個高效可靠的車道辨識系統，以應對不同道路和環境條件下的挑戰，從而為自動駕駛技術的發展做出貢獻。

二、研究方法

本專案選擇使用 Python 編程語言和 OpenCV 庫來實現車道辨識功能。主要的研究方法包括：

（一）顏色選擇

由於本次專案目的是希望能偵測出白色和黃色的道路線，因此我希望能萃取出圖片上只有白色和黃色的區塊，這是在實作車道辨識的第一步，通過分析車道線的顏色特徵來進行。在這一步中，我們將圖像轉換到不同的顏色空間，包括 RGB、HSL 和 HSV 等。我們想了解的是這些顏色空間是否有助於我們更好地理解圖像中的顏色信息，從而準確地選擇出代表車道線的白色和黃色部分。以下是對 RGB、HSL 和 HSV 的簡介。

- RGB (Red、Green、Blue) 是一種最常見的顏色表示方法，它將顏色表示為紅色、綠色和藍色的混合值。
- HSL (Hue、Saturation、Lightness) 將顏色表示為色相、飽和度和亮度三個屬性，這使得我們可以更直觀地理解顏色。
- HSV (Hue、Saturation、Value) 也是一種將顏色表示為色相、飽和度和亮度的方法，又稱 HSB，其中 B 即 Brightness。

(二) Canny 邊緣檢測

Canny 邊緣檢測 (Canny Edge Detection) 是一種經典的圖像處理技術，用於檢測圖像中的邊緣。它是由約翰·坎尼 (John F. Canny) 於 1986 年提出的。該技術的主要目的是找到圖像中像素強度變化劇烈的地方，通常這些地方就是圖像中的邊緣。Canny 邊緣檢測算法通常包含以下幾個步驟。

- 噪聲去除：通過對圖像進行高斯模糊處理來減少圖像中的噪聲。
- 計算梯度：利用一階或二階導數計算圖像中每個像素點的梯度和方向。
- 非極大值抑制：對梯度方向上的像素進行抑制，僅保留局部梯度最大的像素。
- 雙閾值檢測：將梯度值分為強邊緣和弱邊緣，通過設定兩個閾值來進行區分。
- 邊緣跟蹤：通過連接強邊緣像素點，形成完整的邊緣。

Canny 邊緣檢測通常能夠有效地檢測出圖像中的邊緣，並且對於噪聲有一定的魯棒性。它在許多計算機視覺應用中都得到了廣泛的應用，如目標檢測、圖像分割和特徵提取等。

(三) 區域選擇

區域選擇是為了排除圖像中無關的部分，從而提高系統的處理效率和準確性。在這一步中，我們選擇感興趣的區域，通常是道路的前方區域，並排除圖像中的其他部分。這樣可以減少後續處理的計算量，同時提高系統對車道線的準確識別能力。

（四） 霍夫變換線檢測

霍夫變換線檢測（Hough Transform Line Detection）是一種在圖像處理和計算機視覺領域常用的技術，用於檢測圖像中的直線。霍夫變換線檢測的基本思想是通過將圖像空間中的點映射到參數空間中，從而使得直線在參數空間中呈現為一個點。這樣，檢測圖像中的直線就可以轉換為在參數空間中尋找點的問題。霍夫變換線檢測的主要步驟如下。

- 參數空間表示：對於二維圖像中的直線，可以用斜率-截距（斜截式）或極坐標（極坐標式）表示。對於每一個像素點，計算其對應的參數空間中的曲線。
- 累加器空間：建立一個累加器陣列，用來記錄參數空間中各曲線的累積情況。
- 曲線累加：對於圖像中的每一個點，根據其在參數空間中的曲線，對累加器中相應的位置進行增量操作。
- 閾值處理：根據累加器中的值，檢測出可能的直線。
- 後處理：對檢測出的直線進行過濾和合併，以得到最終的直線結果。

霍夫變換線檢測具有較好的魯棒性，能夠檢測出圖像中各種方向和長度的直線。它在許多圖像處理應用中都得到了廣泛的應用，如道路線檢測、目標檢測、圖像分割等。

（五） 平均和外推線段

在車道辨識的過程中，霍夫變換等技術通常會檢測到多條線段來表示車道線的位置。然而，這些檢測到的線段可能存在一定的雜訊和不完整性，需要進一步處理以獲得更加準確和連貫的車道線。平均和外推線段是一種常見的處理方法，其主要目的是將多條線段合併為兩條完整的車道線，即左車道線和右車道線。平均和外推線段的過程如下：

- 分類線段：將檢測到的多條線段區分為左車道線和右車道線。通常，左車道線的斜率是負的，而右車道線的斜率是正的。

- 平均斜率和截距：對於每一類車道線（左車道線和右車道線），計算所有線段的平均斜率和截距。這樣可以獲得一條代表性的車道線。
- 外推線段：根據計算得到的平均斜率和截距，外推出兩條完整的車道線。這可以通過選取兩個固定的 y 值（如圖像的底部和中間位置），根據斜率和截距計算出對應的 x 值，從而得到兩個端點坐標，進而形成一條完整的車道線。

三、實驗結果

通過以上研究方法和步驟，成功實現了車道辨識功能。在測試圖像中，可以清晰地檢測到道路上的車道線，並準確地繪製出來。在測試影像中，雖然在不同光線條件和天氣環境下，使偵測道路的過程中出現許多困難，可能因為一些動態變化使偵測出的道路線較不穩定，但大致上還是成功檢測出來了，以下是我的實驗流程：

（一）顏色選擇結果

首先，載入所有圖片，包括檔案中原本六張圖片和我從指定 YouTube 影片中抓取的其中兩幀，以下的表格剛好對應的是八張圖片的名稱。

表 1 對應的圖片名稱

solidWhiteCurve.jpg	solidWhiteRight.jpg	solidYellowCurve.jpg	solidYellowCurve2.jpg
solidYellowLeft.jpg	whiteCarLaneSwitch.jpg	youtube_frame_0.jpg	youtube_frame_1.jpg



圖 1 原始 RGB 圖片

接著，將照片從 RGB 的格式轉換為 HSV 和 HSL，並尋找在哪種顏色空間中，可以使得我們要尋求的目標，即白色和黃色車道線，更為顯眼。



圖 2 轉換為 HSV 圖片

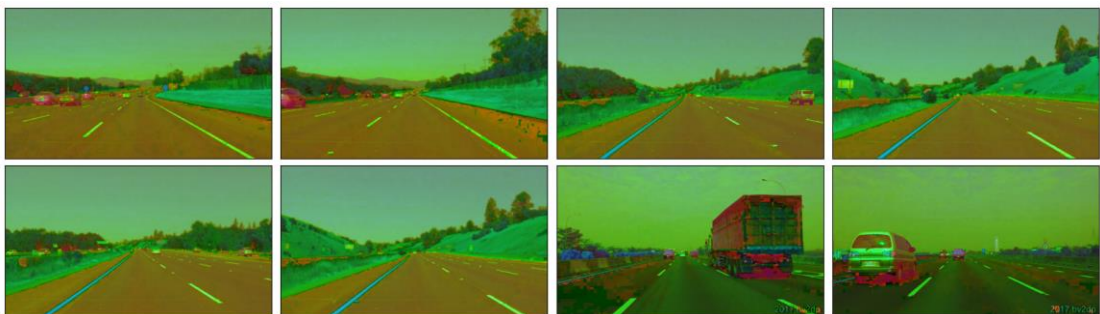


圖 3 轉換為 HSL 圖片

從上面三張圖中，我們可以發現白色車道線和黃色車道線在 HSV 的色彩空間中較不顯眼，但是在 HSL 中似乎卻更為顯眼，也有更強的對比度，因此我們選擇在 HSL 的色彩空間中萃取出白色和黃色的部分，最後再將萃取出的一部分和原本 RGB 圖像結合在一起，即形成下面這張圖。

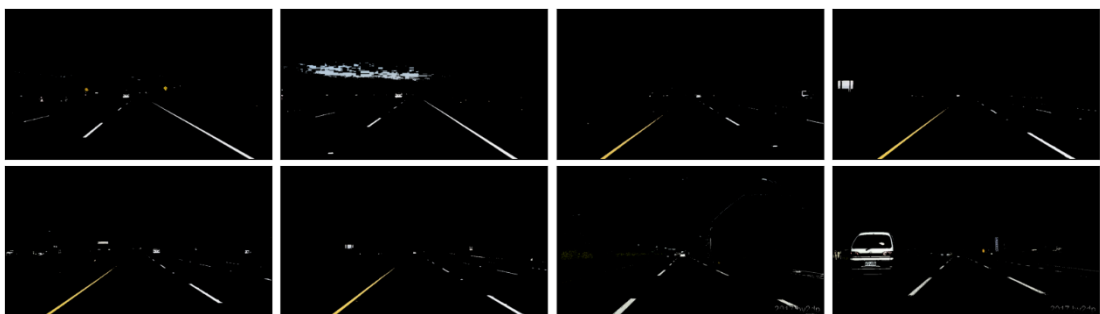


圖 4 萃取出白色和黃色部分的 RGB 圖片

當我們成功萃取出白色和黃色車道線的部分後，即可開始進行下一步，對圖片作 Canny 邊緣檢測，可以從中找出圖像的主要特徵線條，讓我們的车道偵測更為準確。

(二) Canny 邊緣檢測結果

首先，我們在做 Canny 邊緣檢測前，需要去除圖片的噪聲，因此我們先將圖像灰階化 (Gray Scaling)，再將圖像進行高斯模糊處理 (Gaussian Blur)，以下圖像分別為處理完後的結果。

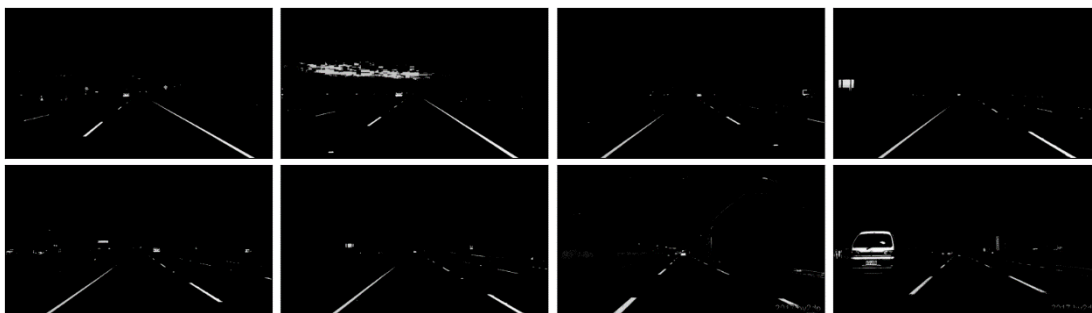


圖 5 灰階化後的圖片

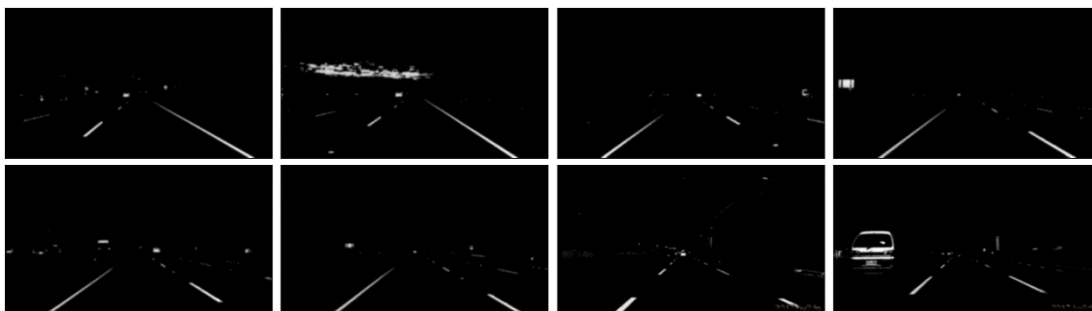


圖 6 高斯模糊處理後的圖片

接著，在做完噪聲去除後，即可開始將高斯模糊處理後的圖片進行 Canny 邊緣檢測，計算出像素中發生劇烈變化的地方，即為我們想要偵測的車道線，以下為做完 Canny 邊緣檢測的結果。

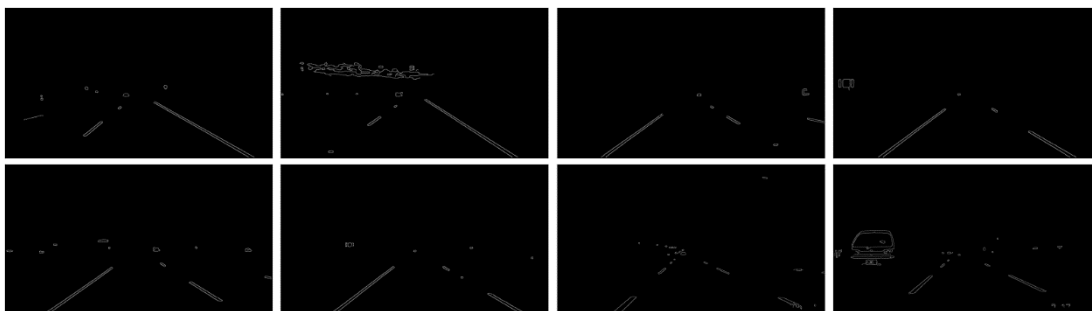


圖 7 Canny 邊緣檢測後的圖片

（三） 區域選擇結果

做完 Canny 邊緣檢測後，我們可以發現偵測出的邊緣不只有車道線，連同其他我們不感興趣的資訊也偵測到了，例如上方的天空、旁邊的山丘，因此我們將這些地方塗黑，只選擇前方需要偵測車道線的四邊形區域，以下為去除不重要地方的結果。

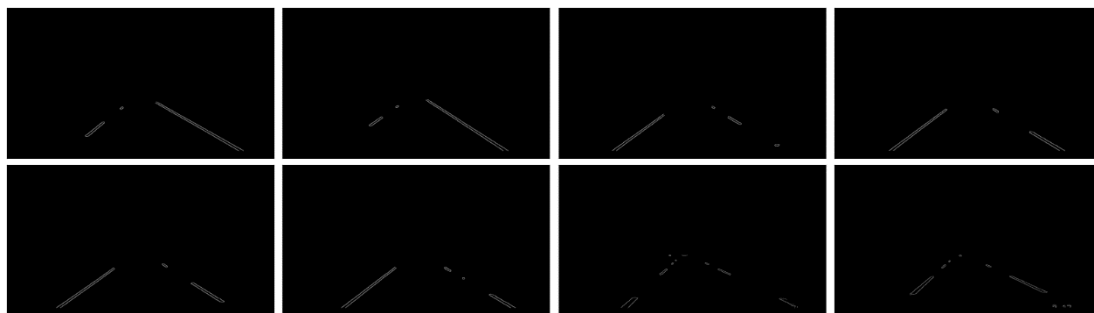


圖 8 區域選擇後的圖片

（四） 霍夫變換線檢測結果

霍夫變換線檢測是為了進一步識別圖像中的直線結構，包括道路上的車道線。我們對經過區域選擇後的圖像進行了霍夫變換線檢測，以檢測圖像中的直線。通過調整霍夫變換的參數，我們得到了檢測到的直線，進一步確定了道路上的車道線位置和方向，以下為霍夫變換線的檢測結果。



圖 9 霍夫變換線檢測後的圖片

（五） 平均和外推線段結果

最後，我們對檢測到的多條車道線進行了平均和外推處理，以獲得兩條完整

的車道線。通過計算檢測到的車道線的平均斜率和截距，我們得到了代表性的車道線。然後，我們根據這些參數，外推出兩條完整的車道線，從而實現對道路的準確定位和行車指引，以下為平均和外推線段後的結果。



圖 10 平均和外推線段後的圖片

四、影像展示

根據在圖片上做偵測道路線的步驟，我們將此一系列的步驟套用在影片上，即可做出一部動態的檢測道路線影像，以下是四部影片的連結及對其檢測結果的評價。

（一）影片一

- 名稱：solidWhiteRight.mp4
- 連結：[solidWhiteRight \(youtube.com\)](https://www.youtube.com/watch?v=solidWhiteRight)
- 檢測結果：在這部影片中，車道辨識系統表現出色，能夠準確識別道路兩旁的白色車道線，並且穩定地跟蹤它們。車道線的外推也相當準確，能夠清晰地顯示出車道的邊界，整體效果優秀。

（二）影片二

- 名稱：solidYellowLeft.mp4
- 連結：[solidYellowLeft \(youtube.com\)](https://www.youtube.com/watch?v=solidYellowLeft)
- 檢測結果：這部影片中，車道辨識系統整體表現優秀，能夠準確識別和跟蹤車道線。然而，在某些情況下，例如車輛晃動時，系統可能會跟著晃動，導致車道線的檢測也跟著不穩定，但整體效果還是相當優秀。

（三） 影片三

- 名稱：challenge.mp4
- 連結：[challenge \(youtube.com\)](https://www.youtube.com/watch?v=challenge)
- 檢測結果：這部影片中，車道辨識系統的表現整體上還算穩定，但偶爾會出現對車道線的誤判或跟蹤不穩定的情況。特別是在車道線顏色與路面背景顏色相近或者路面狀況較差的情況下，系統的表現稍顯不足。此外，霍夫變換線檢測算法適用於直線，對於彎道和曲線較難處理，也可能影響了車道線的檢測準確性。

（四） 影片四

- 名稱：youtube_30s.mp4
- 連結：[youtube 30s](https://www.youtube.com/watch?v=youtube_30s)
- 檢測結果：這部影片中，車道辨識系統在大部分時間內能夠準確地檢測和跟蹤車道線，但在某些情況下，例如有車輛變換或道路狀況複雜時，系統的表現稍顯不足，需要進一步的優化。此外，可能由於僅偵測到較近的白色線段而未能偵測到較遠的白色線段，導致在平均和外推時，較遠的白色道路線較不準確。這可能是導致車道線檢測不夠準確的主要原因之一。

五、結論與展望

透過這份專案，我深入了解了道路線偵測的基本步驟和技術手法。通過不斷嘗試和實驗，我成功實現了車道辨識功能，並且在實踐中遇到了各種挑戰和問題。從一開始不知道怎麼著手進行，到初版將上課教到的內容和步驟直接套用上去，再到上網學習參考其他人在實作道路偵測時的技巧手法，每個階段都讓我學習到很多，也讓我成功完成了本次的專案。其中調整參數方面也是一大挑戰，我投入了大量的時間和精力，尤其是在白色和黃色部分的顏色選擇上。此外，將 RGB 圖片轉換為 HSV 和 HSL 色彩空間等技巧也為我解決問題提供了新的思路。

在未來的工作中，可以將針對車道辨識系統的參數進行進一步的調整和優化。

特別是，我將專注於改進顏色選擇部分，以找到更適合不同光線條件和路面顏色的參數設置。透過對顏色空間和閾值的更深入理解，調整相應的參數以提高系統對道路標誌的準確識別能力。其次，或許可以研究如何利用機器學習和深度學習技術，從大量的數據中學習和提取有用的特徵，以實施更高級的影像處理技術，應對各種複雜路況和環境條件。例如，針對曲線和彎道路段，進一步提升系統的性能。