國立雲林科技大學  
電子工程所

National Yunlin University of Science and Technology

Graduate School of Electronic Engineering

報告題目 :汽車駕駛者監控系統設計與實現

|  |  |
| --- | --- |
| 報告日期(Date)： | 2022/12/23 |
| 姓名(Name)： | 謝昕樺(碩一生) |
| 指導教授(Professor)： | 蘇慶龍副教授 |
| E-mail： | [M11113020 yuntech.edu.tw](mailto:M10813007@yuntech.edu.tw) |
| 學號 (Student ID)： | M11113020 |

**本週進度(Progress of this week)**

* 準備Paper報告

12/10 Sat 14:00~20:00 ( 6.0 個小時)

12/11 Sun 14:00~19:00 ( 5.0 個小時)

12/11 Sun 21:00~02:00 ( 4.0 個小時)

12/12 Mon 13:00~15:00 ( 2.0 個小時)

12/12 Mon 17:00~20:00 ( 3.0 個小時)

12/12 Mon 23:00~02:00 ( 3.0 個小時)

12/13 Tue 16:30~23:00 ( 6.5 個小時)

12/14 Wed 13:00~21:00 ( 8.0 個小時)

12/15 Thu 12:30~14:00 ( 2.5 個小時)

12/15 Thu 16:30~18:30 ( 2.0 個小時)

12/16 Fri 16:30~22:00 ( 5.5 個小時)

12/17 Sat 14:00~16:30 ( 1.5 個小時)

12/17 Sat 19:00~21:00 ( 2.0 個小時)

12/17 Sat 00:00~04:30 ( 4.0 個小時)

12/18 Sun 18:00~20:00 ( 2.0 個小時)

12/18 Sun 22:00~02:30 ( 4.5 個小時)

12/19 Mon 14:00~02:00 ( 12.0 個小時)

12/20 Tue 14:00~15:00 ( 1.0 個小時)

12/20 Tue 16:00~17:30 ( 1.5 個小時)

12/20 Tue 23:00~01:30 ( 2.5 個小時)

|  |  |
| --- | --- |
| **12/9進度** | **12/21進度** |
| 1. 看學長code架構 2. Yolov5/Yolov7架構 | 1. 準備Paper報告 2. 查看車道線論文 |
| **待辦事項** | |

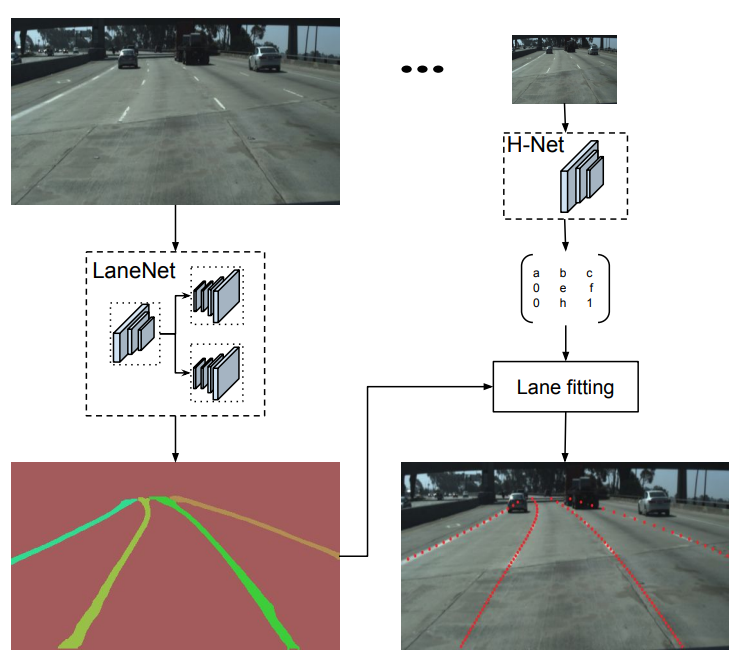
**閱讀文獻**

Towards End-to-End Lane Detection: an Instance Segmentation Approach

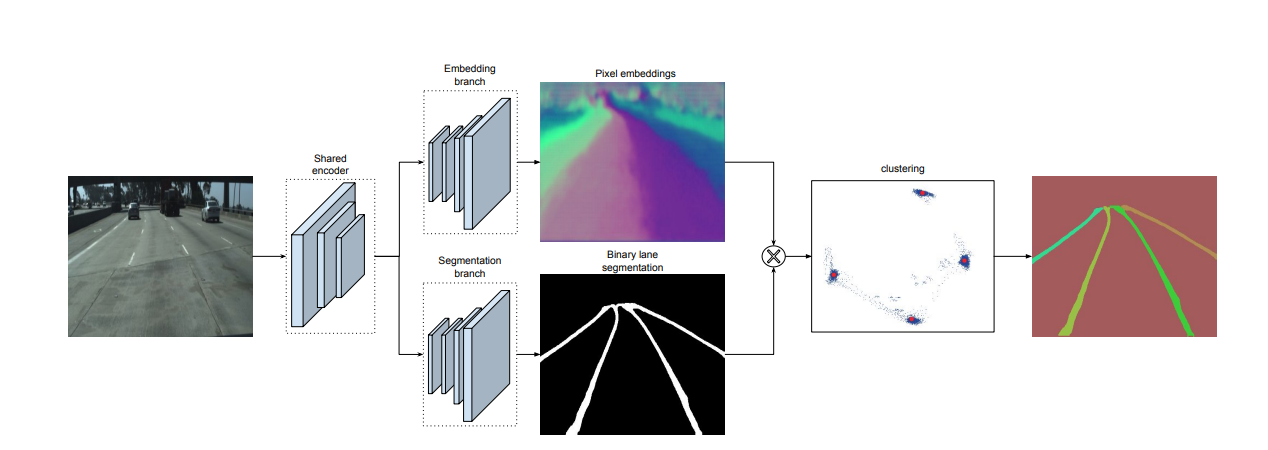
**摘要**

論文作者提出一種多分支網路架構，將車道線以實例分割， 讓每條車道線都有自己的實例去進行端到端訓練，接著在運用H-Net網路中的H矩陣對屬於同一點車道線的像素做回歸

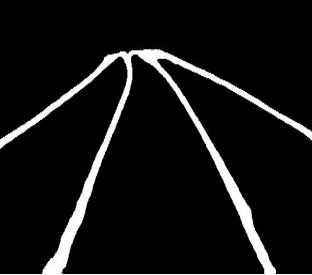
**重點整理:**

本論文使用了兩種網路架構分別為LaneNet和H-Net，LaneNet是輸入用Encoder的方式，輸出Decoder分為兩個分支，分別為語義分割和實例分割，最後將兩張圖合併，而H-NET是利用Conv和全連接層計算H矩陣內6個參數

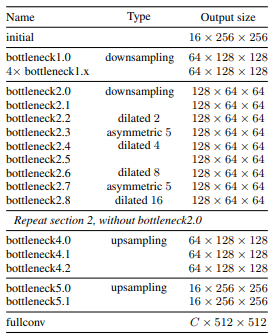
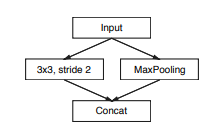
* LaneNet

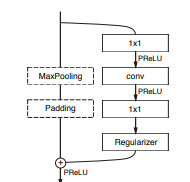
LaneNet是利用Encoder和Decoder的方式，其中Encoder共用，Decoder分成了兩個分支，分別為語義分割(semantic segmentation)和實例分割(instance segmentation)

* 語義分割(semantic segmentation)

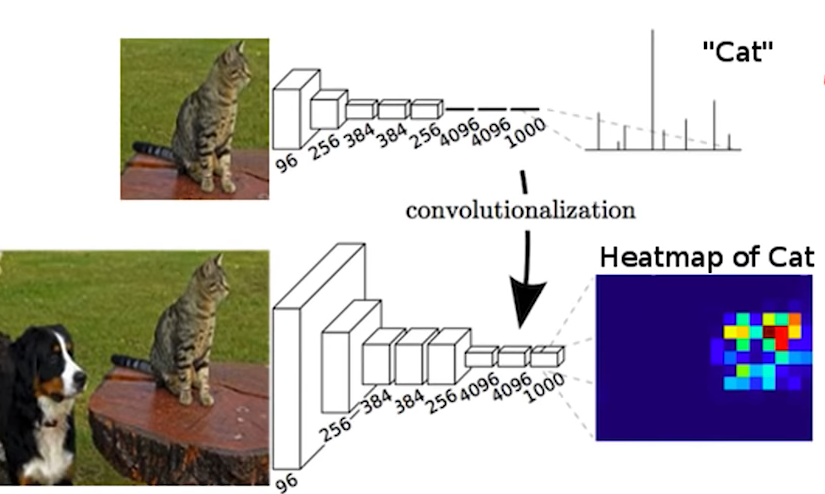
將相同類別的像素運用相同顏色做分類，而這篇論文是將每個像素點給一個Label去判定是車道線還是背影，目的是將車道線和背景做分割

本篇論文是參考E-Net論文中使用的VGG16架構，並將VGG16架構中Fully Connect改成Fully Convolution，Stage 1~3為Encoder，其中Stage 1和Stage2都使用下採樣，Stage 3沒有下採樣，Satge 4~5為Decoder，將照片還原為原本大小並對其像素分類

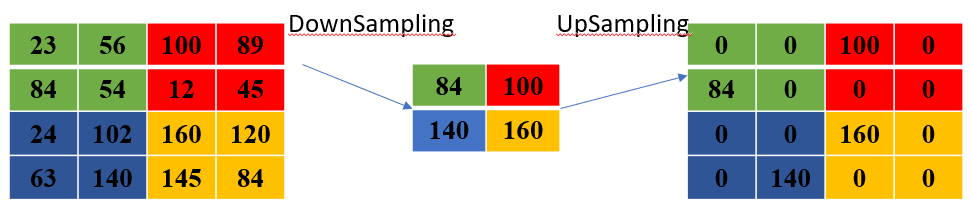




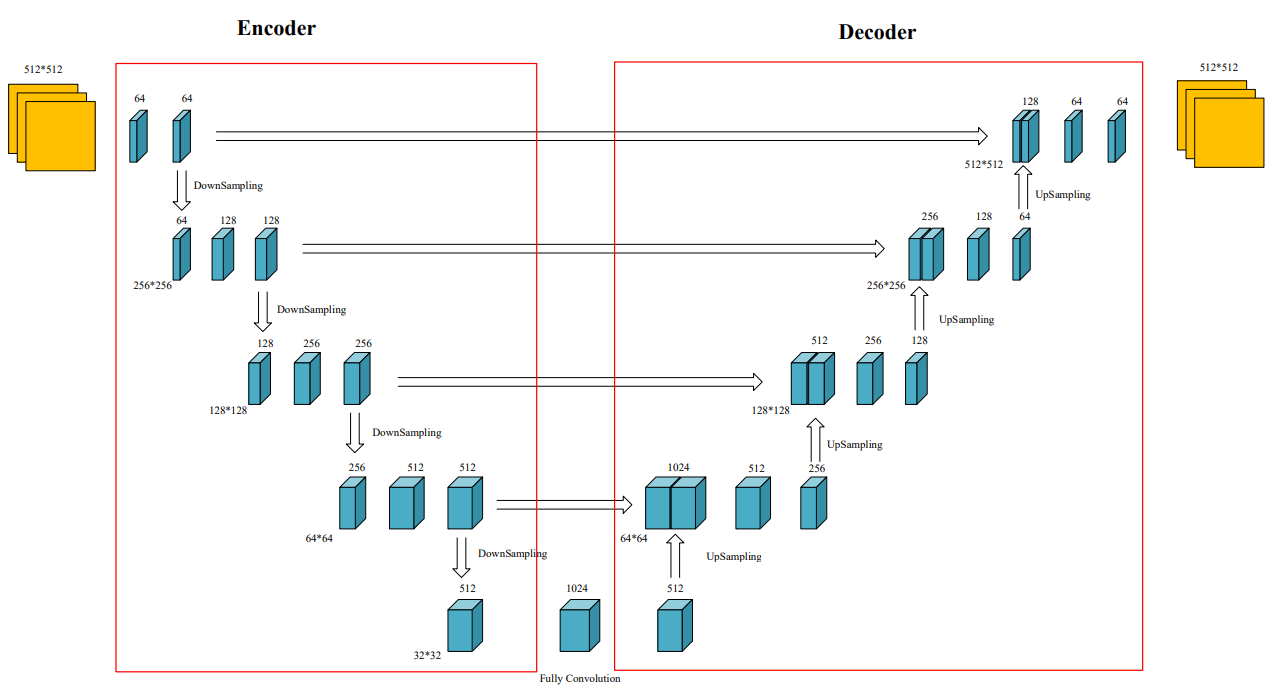
首先是Fully Convolution部分，傳統Conv是將最後一層特徵壓縮至一個1\*1\*n維的Vector後進全連接利用Softmax進行分類，這樣做會讓Image丟失很多空間訊息，沒有辦法利用Decoder還原每個像素點的類別，若是將Fully Connect改成Fully Conv方式輸出到最後可以有N維的Feature Map，這樣就可以對像素進行分類



接下來是Upsampling部分，當今天要Decoder回原本Feature Map Size的時候，是會先去尋找Encoder 相同層的Feature Map Size，將Maxpooling後的位置記錄下來，將其位置複製給Decoder的Feature Map Size，其餘空洞地方補0

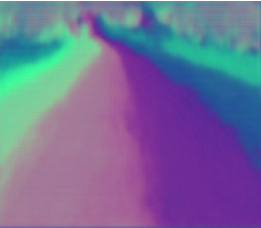


接下來將Encoder經過幾層Conv，Decoder就做幾層Conv，並且和Encoder的Feature Map做Concat

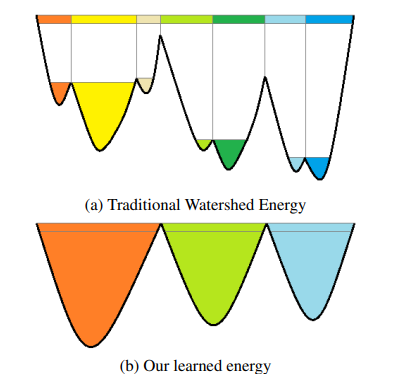


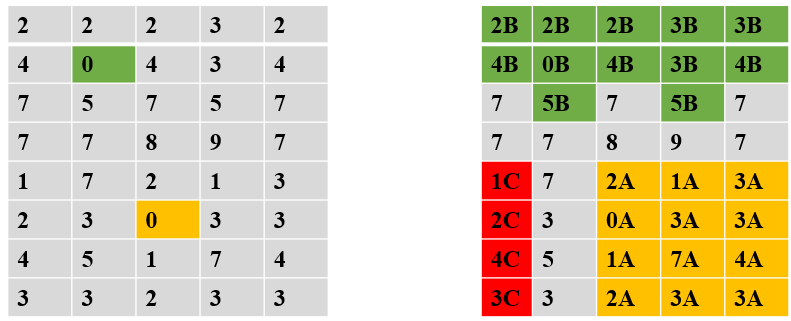
* 實例分割(instance segmentation)

除了分類相同類別的像素外，還要知道有幾組相同類別的像素，每組都以不同顏色做標記，而這篇論文是將標記出來的車道線再進行分組依據Input輸入有幾組車道線就分幾組

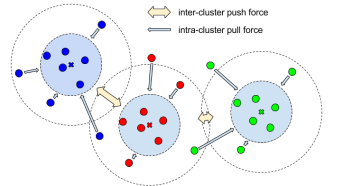


車道線分組是參考Deep Watershed Transform for Instance Segmentation這篇論文，傳統的Watershed Transform會因為噪聲太多而產生許多分水嶺，因此利用卷積網路去學習特徵取代傳統梯度值



Watershed Transform作法會找出局部最小值，將他們分為N類，接下來由這N類向外延伸找尋，以下圖為例目前是分成兩類，將0放入Priority Queue內，此時局部最小值為1，並搜尋0周圍8個像素點看有沒有像素點1，若有與最近的0歸為同一類，若像素點1是在2組8個像素點之外，則自己歸類為一類以此類推

而相同Label的Pixel應該要更靠近彼此，不同Label的需要遠離彼此，因此Loss為式1-1和式1-2，Loss Total為式1-3



式1-1

式1-2

式(1-3)

|  |  |
| --- | --- |
| 參數 | 作用 |
| C | 總共有幾條車道線 |
| NC | 每條車道線中實例像素數量 |
|  | 每條車道線實例像素平均 |
|  | 當前車道線像素 |
|  | 自訂義參數，用於調整和距離 |
|  | 自訂義參數，用於調整和距離 |
|  | 同一車道線像素向量和車道線像素均值相減距離若大於，模型更新讓更靠近 |
|  | 不同車道線像素均值和若小於，模型更新讓2者遠離 |

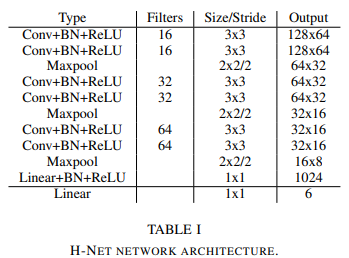
上圖中像素點Loss是以Mean Shift為基礎，以半徑大小和的圓做出發去進行分類，若在半徑大小內相同組別像素者，計算其像素點平均值，更新圓的中心點，再進行比較直到中心點不再移動為止

* H-Net

LaneNet的輸出是每條車道線的像素點集合，需要利用多項式回歸將車道線擬合出來，因此論文利用了一個可以轉置矩陣H(式1-4)的H-Net神經網路訓練

式1-4

轉置矩陣H內部有6個參數，因此H-Net輸出是一個6維Vector，神經網路訓練如下



接下來要計算其損失函數，計算方法是以y座標去預測x座標，首先假設地面有N個車道線真實像素點，接下來利用H-Net計算出來的輸出H進行座標轉換如式1-5

式1-5

接下來用最小平方法進行多項式參數預測如式1-6

利用真實座標點減去預測座標點

式(1-6)

接下來利用預測出結果的多項式參數預測出如式(1-7)

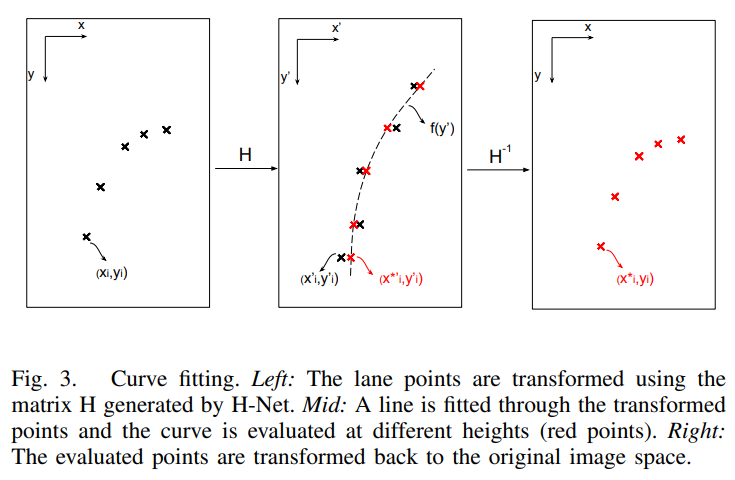
式(1-7)

將投射回真實座標如式(1-8)

式(1-8)

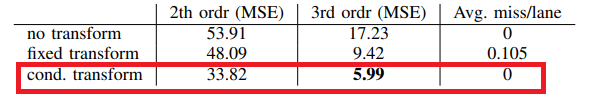
最後將利用MSE預測和的Loss Function如式(1-9)

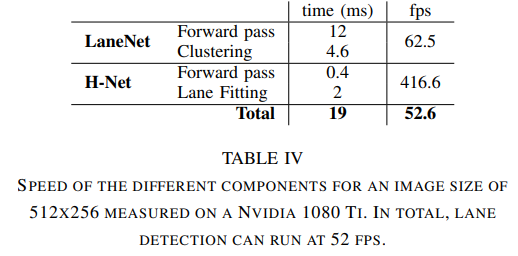
式(1-9)



* 比較結果

論文最後有利用2次式和3次式去擬合車道線，以及比較有無使用H轉置矩陣如下圖



並且輸入大小512\*256圖像，得出結果為19ms，每秒最多處理52偵

