

# **Speed Estimation Using Optical Flow**

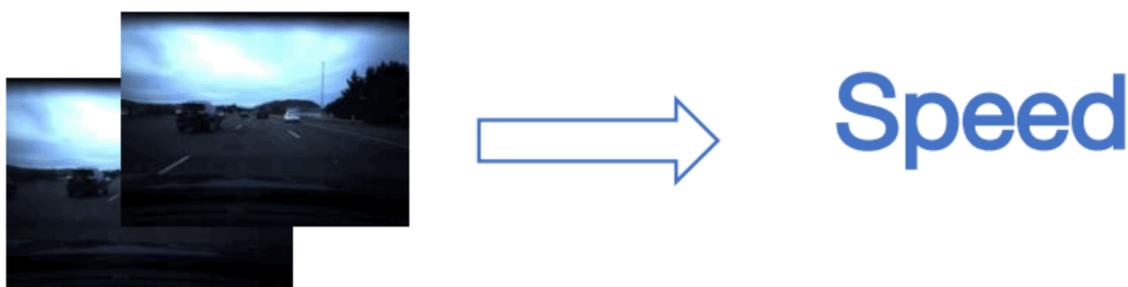
Group 11  
r10922026 吳勝濬  
r10922102 林正偉  
r10922161 黃正輝

# Outline

- Introduction
- Methodology
- Experiment
- Result
- Discussion
- Reference

# Introduction

在本次project中，我們嘗試研究僅使用由一般camera拍攝的第一視角的visual data能否很好地預測目標物體自身移動的速度。研究實驗中的輸入和輸出如下圖所示，我們將輸入的video按照一定的fps切成若干個frames，再又每兩個相鄰的frames預測出速度。



經過survey後，我們發現有些研究者使用例如ConvLSTM或Transformer這類可以同時處理不同時間點的frames的方法直接對原圖去預測速度。但我們認為相鄰的兩個frames與某個時間點的速度相關性更大。基於這個想法，我們認為可以先得到optical flow，再由此得到速度。而一些相關的工作也經過實驗證實了我們的想法。而且使用optical flow的方法也更貼合這門課程的內容，在這個project中我們將嘗試不同的optical flow方法，測試他們的效能並藉此更加深入學習optical flow。

## Methodology

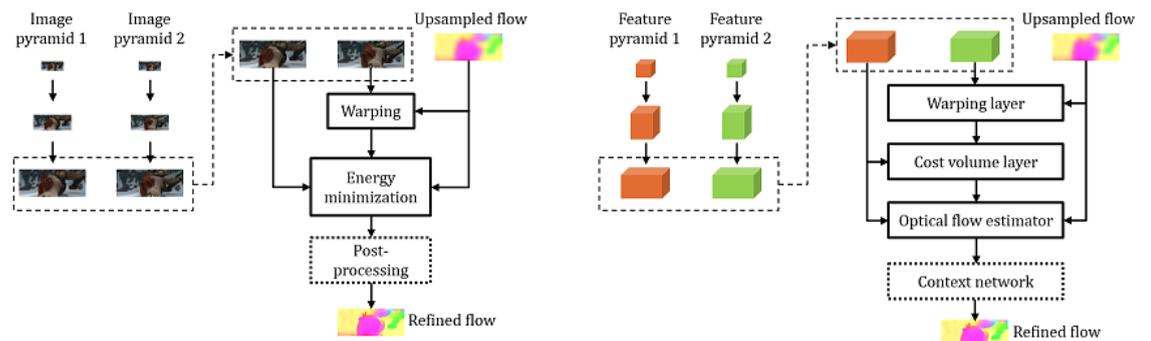
我們的方法主要分為以下幾個部分，先將輸入的video按照一定的fps切成若干個frames，再由相鄰的兩個frames算出optical flow，最後將得到的optical flow輸入neural network輸出預測的速度。



由於我們的方法中包含兩個獨立的部分，計算出兩兩frames之間的optical flow以及預測speed的model, 所以接下來我們會分別大致介紹這兩個部分我們所使用的方法及架構。

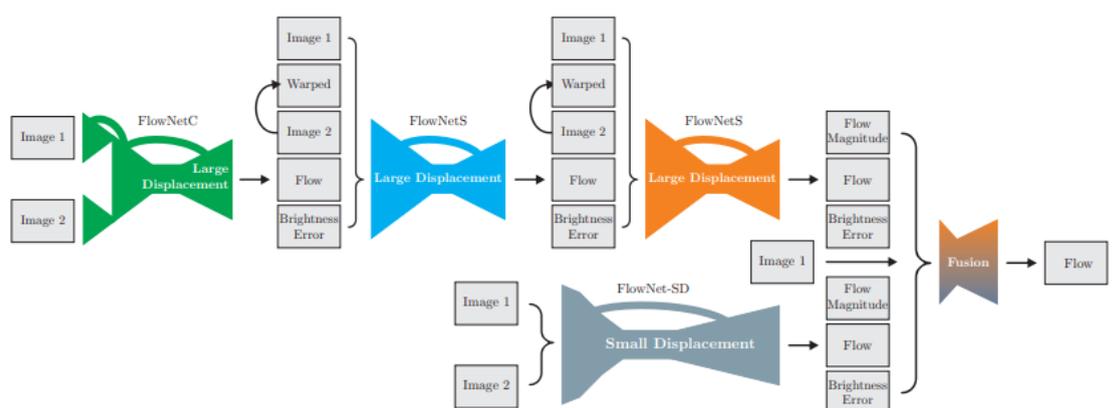
## ❖ Optical Flow

### 1. PWC-Net



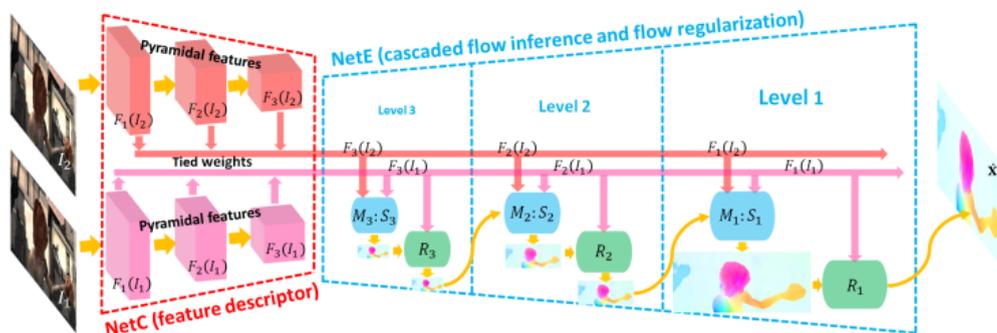
這演算法結合了Pyramid、Warping、Cost Volume。左圖是傳統的方法，通過構建金字塔處理不同尺度的 Optical Flow，再利用進行 coarse-to-fine 的方式得到 Refined Flow。右圖是 PWC-Net 的架構，基於傳統 coarse-to-fine 演算法改良，使用 CNN 來處理 Cost Volume 最後得到 Refined Flow。

### 2. FlowNet 2.0



FlowNet2.0 是 FlowNet 的增強版，在 FlowNet 的基礎上進行提升，在速度上只付出了很小的代價，使性能大幅度提升。通過結合不同的 FlowNet module，處理不同尺度的位移，最後將結果整合在一起，預測出 Optical Flow。

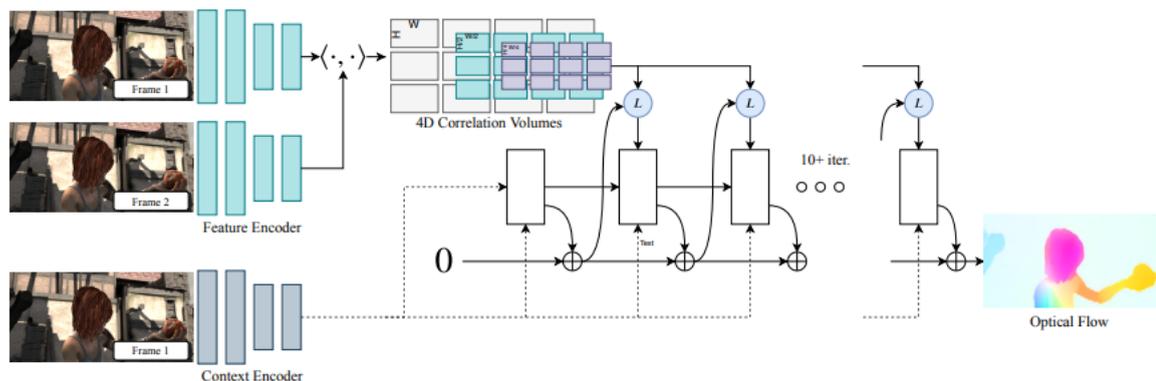
### 3. LiteFlowNet



基於 FlowNet 2.0 所做的改良, 透過對金字塔每層的特徵做更有效率的 inference 方法, 減輕了 FlowNet 2.0 龐大的網路架構, inference 速度也獲得有效的提升。

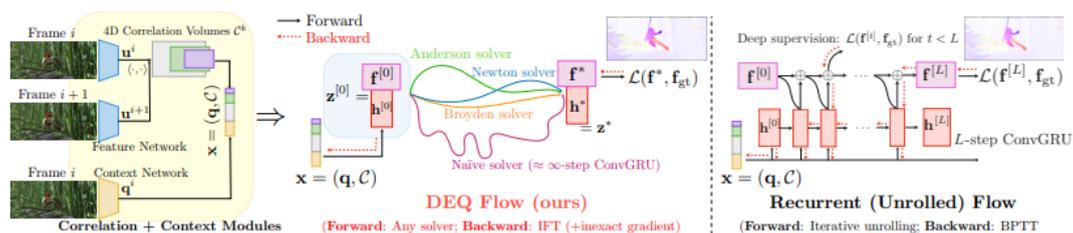
並且透過 Flow Regularization 來改善 Outlier 的問題。

### 4. RAFT (Recurrent All-Pairs Field Transforms)



RAFT包含三個主要部分: (1) Feature encoder 用來對兩張 Input frame 做特徵提取; (2) Correlation layer 使用所有feature vectors pairs 的內積, 建構一個 4D correlation volume ( $W \times H \times W \times H$ ); (3) Update operator 透過當前的估計來查 correlation volume, 利用 RNN 遞迴的更新 Optical Flow。

## 5. DEQ flow

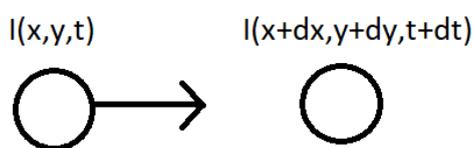


在得到兩張圖片的 correlation volume 後，採用任意的 Solver 直接解出一個穩定的 Flow 結果，取代了 RAFT 中使用的 RNN 架構。這改善了 RNN 計算量大以及不好收斂的問題，得到了更有效率的計算和記憶體使用。

## 6. Dense Optical Flow

首先假設一個物件 intensity 為  $I(x, y, t)$ ，經過  $dt$  時間過後，變成

$$I(x+dx, y+dy, t+dt)$$



$$\text{i.e. } I(x, y, t) = I(x+dx, y+dy, t+dt)$$

將RHS使用 Taylor approximation, 得

$$\frac{dI}{dx} \delta x + \frac{dI}{dy} \delta y + \frac{dI}{dt} \delta t = 0$$

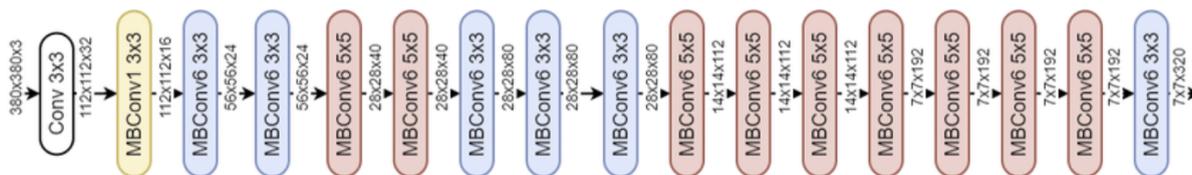
除以  $\delta t$ , 得到 Optical Flow Equation

$$\frac{dI}{dx} u + \frac{dI}{dy} v + \frac{dI}{dt} = 0$$

最後透過 Gunner Farneback's algorithm, 計算每個 Pixel 的 Optical Flow, 得到最後結果。

## ❖ Neural Network of Speed Estimation

EfficientNet是一個能夠同時兼顧速度與精度的架構。作者通過grid search的方法調整width、depth和resolution三個維度，尋找出能夠平衡速度和精度的結果。在這個project中我們希望最終的推理速度能夠即時，所以最終採用此架構。EfficientNet共有b0-b7八個版本，我們這裡實驗b0,b1,b2三個版本。以下是EfficientNetb0的architecture。

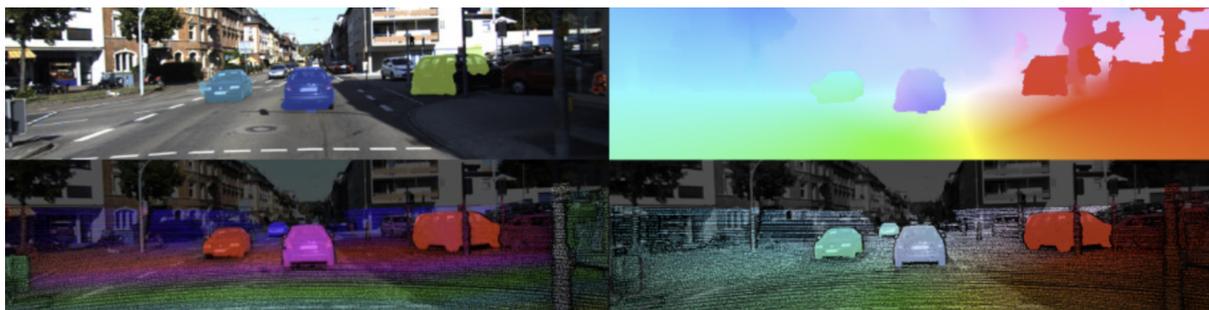


## Experiment

我們實驗了以上介紹的6種optical flow方法以及3種efficientnet。以下介紹訓練這些model使用到的dataset。

## ❖ Dataset

在optical flow中deep learning method的部分，我們大多使用KITTI dataset訓練。KITTI是目前自動駕駛領域最重要的測試集之一，包含市區、鄉村和高速公路等場景採集的真實圖像數據。整個數據集由289對立體圖像和光流圖等組成。因為這個數據集與我們後續預測的場景相似，所以我們預期在這個dataset上訓練好的optical flow的model在我們的預測場景中能有較好的效果。



而在speed estimation的部分我們則是選擇由comma.ai提供的SpeedChallenge。此dataset包括train.mp4和test.mp4, 根據fps為20的參數我們可以將train.mp4切成20400 frames。其中只有train.mp4的部分有提供label, 所以我們將其前80%作為training dataset, 後20%作為validation dataset, 在設定中最後的結果均呈現在validation dataset上。



## Result

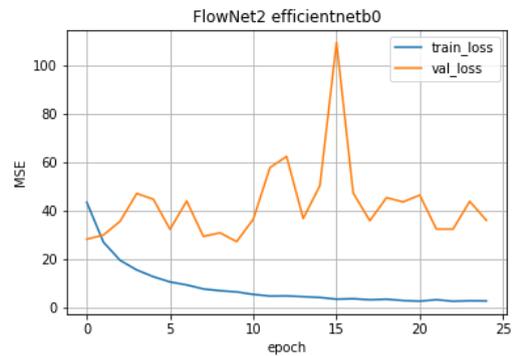
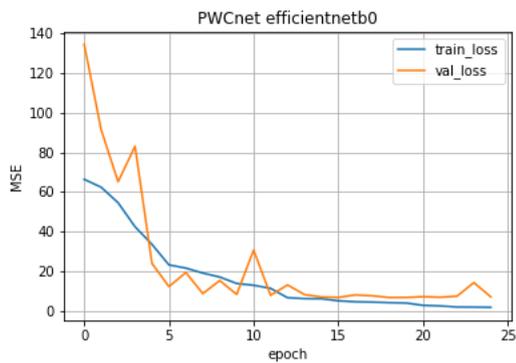
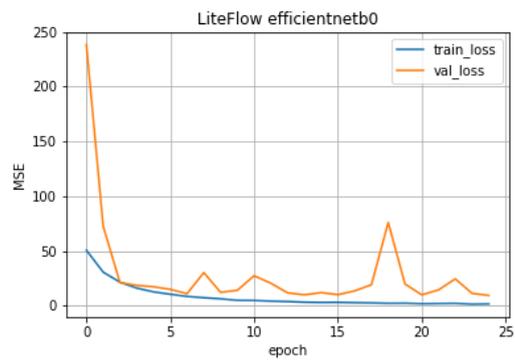
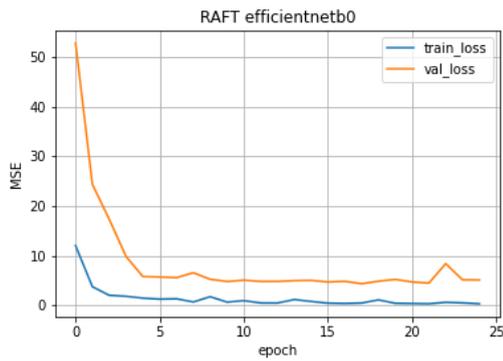
由於我們實驗的內容較多, 所以先固定使用RAFT得到optical flow, 實驗出speed estimation部分最佳的架構。在speed estimation的部分, 我們的epoch為25。通過實驗我們發現EfficientNetb0的效果最好, 且是3個架構中最輕量化的, 因此我們選擇此架構預測speed。

Method(RAFT)	EfficientNetb0	EfficientNetb1	EfficientNetb2
MSE	4.332	5.611	5.192

接著我們實驗了6種optical flow的方法, 並使用得到的optical flow預測速度。最終的結果如下圖, RAFT的效果最好。

Method(b0)	DEQ	PWC-Net	LiteFlowNet	RAFT	DOF	FlowNet 2
MSE	7.463	6.719	5.087	4.332	7.243	27.255

以下是不同model的loss curve, 可以看到在其中RAFT得到的結果最好, validation loss也基本有收斂。四種做法的train loss都有收斂, 但是在FlowNet2中的情況比較差原因可能在於其在optical flow inference的部分得到的結果不優。



最後我們將結果可視化，左上角為預測速度(mph)，中間為ground truth的速度，下面為兩者的差值。



## Discussion

車子在如畫面這樣平穩運行時，預測誤差值較小。但是當車子運行速度較慢或者停下來時會造成較大誤差。造成這樣現象可能的原因是，在車子停下時一些其他運動的物體會影響model的判斷，而且場景中一些不必要的noise可能也會降低最後的performance。而且我們這裡使用的validation dataset與training dataset使用的是同一video，所以待測data的變異度

不夠大，在實際場景中的表現無法預期。因此我們認為這個project還有很大的改進空間。以下是我們認為可能可以提高performance的方法：

1. 使用segmentation的方法去處掉畫面中某些可能誤導model的物體。以此降低noise的影響。
2. 在speed estimation的部分設計出更適合這個task的架構。
3. 收集不同場景的data, 增加model的泛化能力。並且建立test dataset, 以此得到model更加真實的預測能力。

## Reference

- [1] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." *International conference on machine learning*. PMLR, 2019.
- Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." *International conference on machine learning*. PMLR, 2019.
- [2] Bai, Shaojie, et al. "Deep Equilibrium Optical Flow Estimation." arXiv preprint arXiv:2204.08442 (2022).
- [3] Teed, Zachary, and Jia Deng. "Raft: Recurrent all-pairs field transforms for optical flow." *European conference on computer vision*. Springer, Cham, 2020.
- [4] Sun, Deqing, et al. "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [5] Ilg, Eddy, et al. "Flownet 2.0: Evolution of optical flow estimation with deep networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
- [6] Hui, Tak-Wai, Xiaoou Tang, and Chen Change Loy. "Liteflownet: A lightweight convolutional neural network for optical flow estimation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [7] Farneback, Gunnar. "Two-frame motion estimation based on polynomial expansion." *Scandinavian conference on Image analysis*. Springer, Berlin, Heidelberg, 2003.

# Appendix

Name	Contribution	Proportion
吳勝濬	實作optical flow(大部分)、實作speed estimation	1/3
林正偉	實作optical flow、實作speed estimation、撰寫report	1/3
黃正輝	survey、實作speed estimation、presentation、撰寫report	1/3