

Computer Vision Final Project : Depth Map Generation on More Realistic Scenes

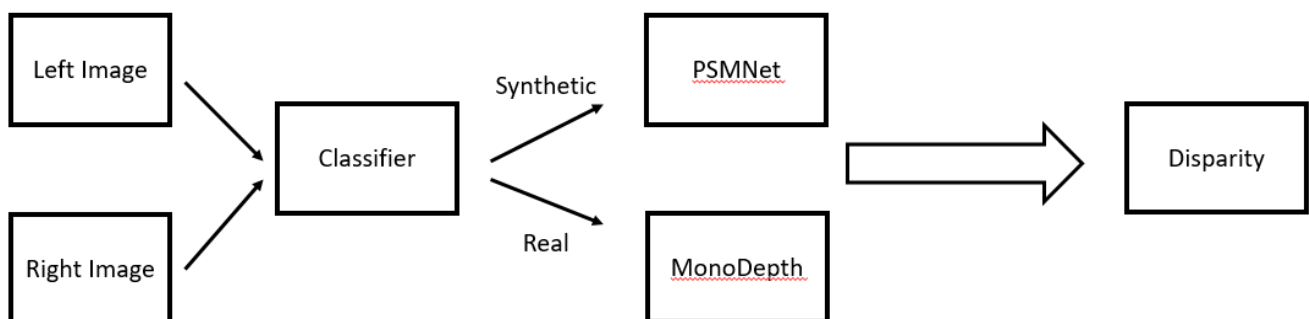
Name: 楊晟甫 Department: 電機三 Student ID: B05901082

Name: 許秉倫 Department: 電機三 Student ID: B05901011

1. Introduction

這次的期末專題是Depth Map的generation，其中包含了Synthetic image和Real scene image，我們嘗試了許多不同的方法，包含傳統的fast cost volume，以及用deep learning 的matching cost，ex. monodepth、PSMNet，我們起初嘗試用同一個model來inference我們的disparity map，但成效不彰，所以後來我們選擇先用一個CNN來判斷此圖是Real/Synthetic Image，再分別經過我們的PSMNet(Synthetic)、Monodepth(Real)，最後得到我們的Disparity Map。

2. Architecture



3. Implementation

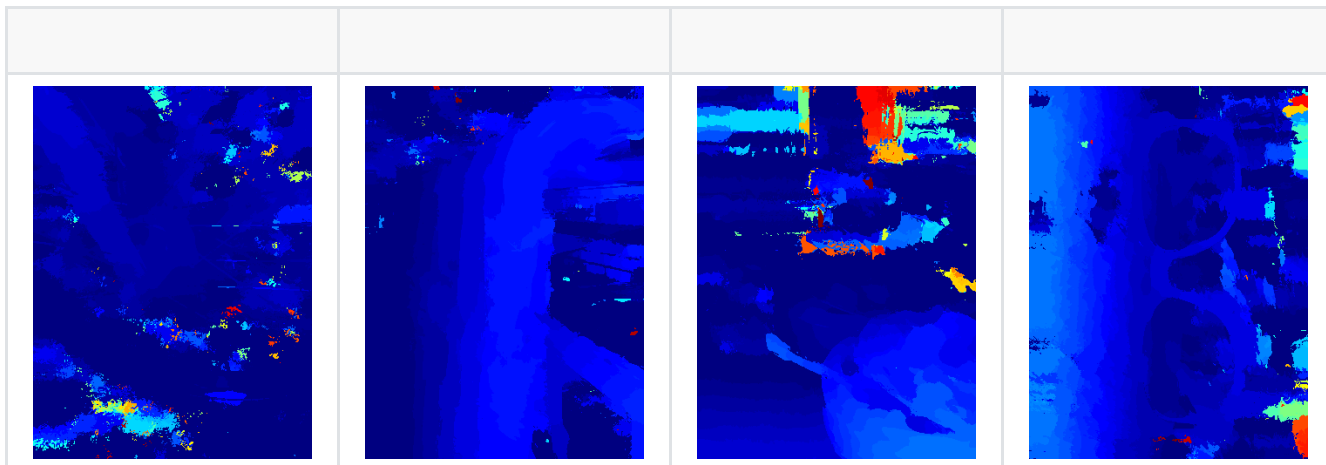
(0) Thought Process

一開始遇到這題目，我們打算承接hw4的架構繼續做下去，但是我們發現我們的input 需要RGB的image，我們硬把灰階轉成3個channel放進去但效果不佳。(如下圖)，雖然有點邊界，但是一方面由於我們在deep的synthetic表現很好，所以我們決定轉戰deep learning，另一方面是評分標準考量到運算速度的問題，deep在這方面遠優於傳統的方法。

起初我們使用了PSMNet的架構，但是為避免overfitting我們有自己寫了transform的函數來同時對left、right、disp做相同的augmentation，表現如我們預期，在train了5000 epoch 後 training跟validation的error仍有下降趨勢，但是這個model在real data上表現不佳，歸根究柢我們發現由於real data的disparity range 在同一張圖裡有正也有負，所以也不能單把左右兩張圖做交換丟進network。

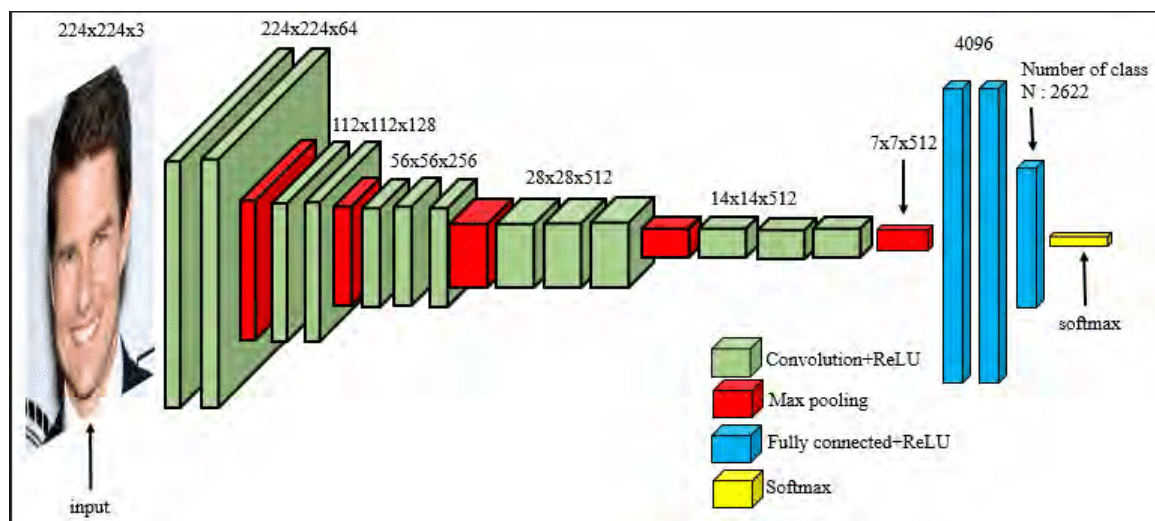
因此我們想到可以用left right consistency作為我們的loss，所以我們開始朝這方面做survey，發現monodepth這篇能夠在kitti dataset上與其他supervise的方法取得不相上下的結果，但是他只有提供pretrain model，所以我們去學了tensorflow來對於real data做finetune training，我們每10個epochs inference一次，可明顯看到邊界深淺，有越來越明顯的趨勢。

我們也試過用segmentation的mask讓每個區域的disparity更接近，但是現成的segmentation的方法似乎對real的辨識有它的極限在，不夠細緻。

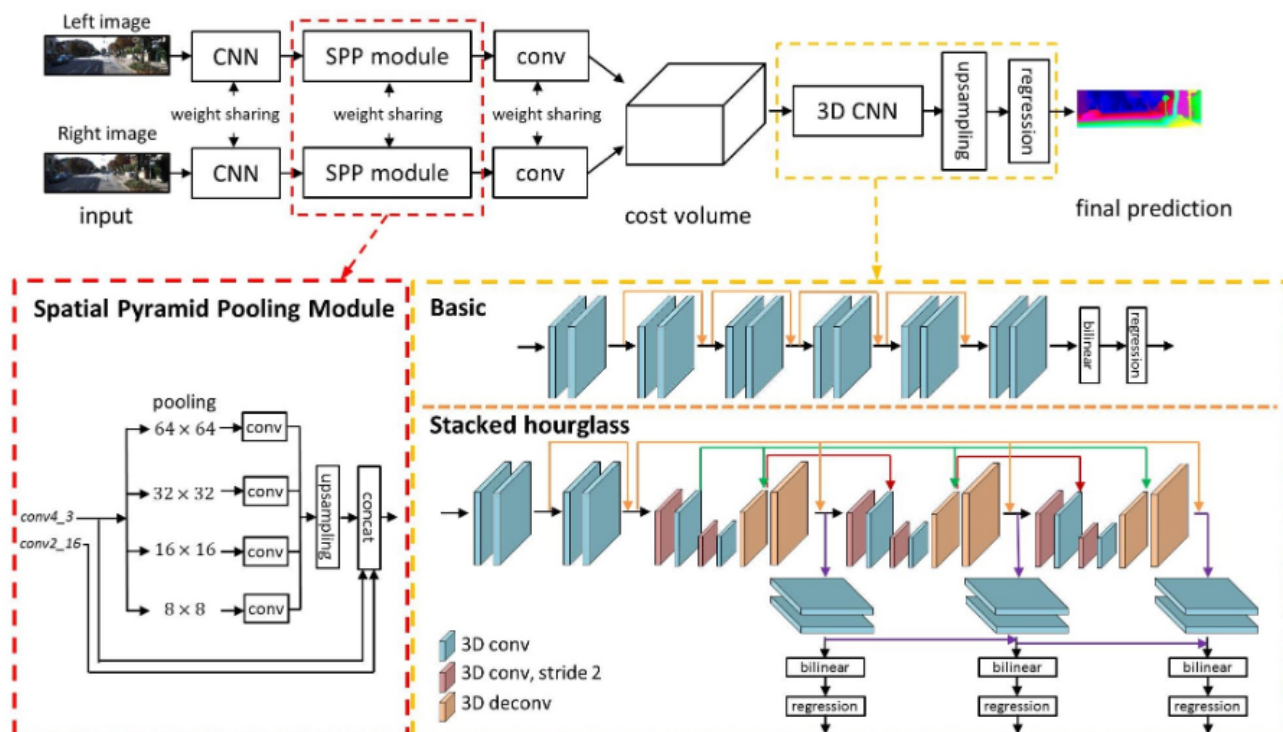


(1) Real/Synthetic Classifier

我們使用了類似VGG的模型架構，利用real和synthetic data，轉成灰階，並做augmentation，來做training，大約100個epoch內就可以達到100%的準確率。



(2) PSMNet



1. CNN:

作者首先用了比較小的kernel(3x3)串接多層，形成一個較深層的model來獲得跟其他論文相同大小的receptive field。

2. SPP Module :

SPP 可以把不同大小的feature map flatten然後餵進 fully-connected 的網路，聚合不同尺度的資料。

3. Cost Volume :

作者把left_feature跟right_feature concat 在一起，得到一個四維的vector(H X W X D X Feature)。

4. 3D CNN :

得到cost volume之後，利用作者架構的stacked hourglass模型(一個encoder-decoder的架構)，用此架構及SPP模型，作者宣稱可以讓model利用到global feature和local feature來regularize cost volume，進而處理到遮擋的區域、重複出現的特徵、反射區域等在影像處理上棘手的問題。

5. Disparity Regression :

$$\hat{d} = \sum_{d=0}^{D_{max}} d \times \sigma(-c_d).$$

d hat 是計算得到的disparity，c_d是predict的cost，d是原本每點的disparity

6. Loss :

$$L(d, \hat{d}) = \frac{1}{N} \sum_{i=1}^N smooth_{L_1}(d_i - \hat{d}_i),$$

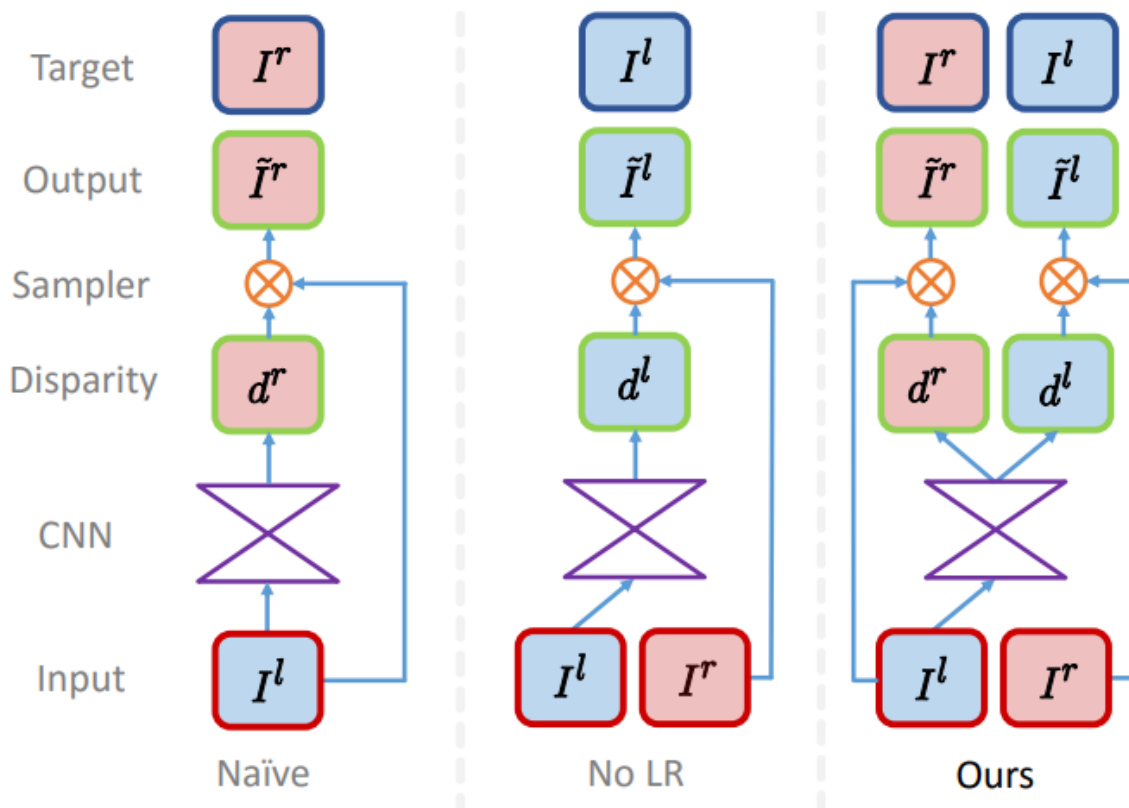
$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases},$$

使用L1 loss的原因是因為其比L2更加強健，對於異常點也比較不敏感，其中d是ground truth disparity，d hat則是predict disparity。

7. Training :

我們將max_disp設為64(實驗後的最佳disp)，分別觀察data augmentation的有無，並以經過transforms的圖片當作validation，我們發現，沒有augmentation的model雖然能在TL0~TL9上取得較佳的表現，但是對於transforms的圖是有overfitting的趨勢，但後者雖然error不如前者低，但也是在可接受範圍，並在transforms的圖片上表現良好。

(3) Monodepth



1. Overview:

此模型為unsupervised learning，意味著他並不需要ground truth就能做訓練。他主要的特色為，透過disparity重建出原圖，並計算其中差距當作loss，能近乎與supervised learning的方法並駕齊驅。

2. CNN:

主要由兩個部分組成：

1. 編碼器：conv
2. 解碼器：deconv

產生左右圖的disparity map

3. Sampler

使用the spatial transformer network (STN)來sample出原圖，STN使用bilinear sampling，其中產生的pixels會是輸入的pixel的weighted sum

4. Loss設計

1. Cap: 重建後與原圖的影像差距
2. Cds: disp整體的平滑性
3. Clr: left-right consistency

5. Training

我們載下kitti的pretrained model後，在助教給的資料做fine tune，train了倆百個epochs左右，已經明顯可看出物體的輪廓

(4) Self-reflection

聽完大家的報告後我們覺得我們preproces 的部分是比較缺乏的，我們試過edge enhancement，以及cv2 的threshold、erosion等，但效果不彰，但報告某些組別的preprocess感覺有顯著提升performance，這方面是我們想去多加鑽研的，另外postprocess的部分我們也曾使用過weight median filter，但在我們的edge以及object都還沒有很顯著的情況下，效果並不明顯。

4. Results

Running Time (Evaluate one image)

	Real	Synthetic
Time	1.5 sec	0.5 sec

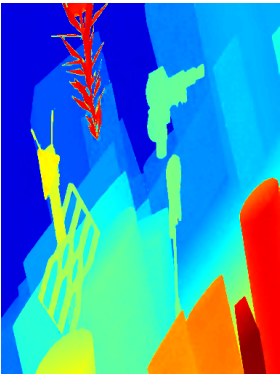
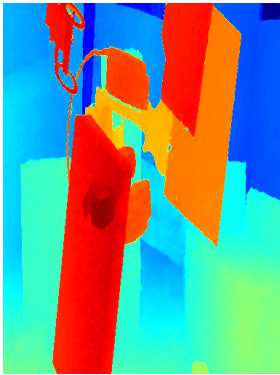
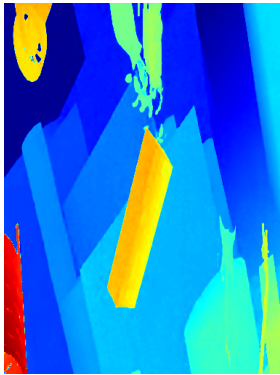
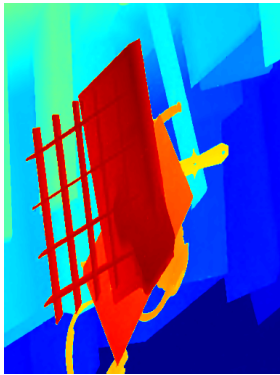
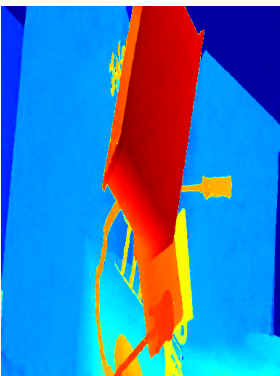
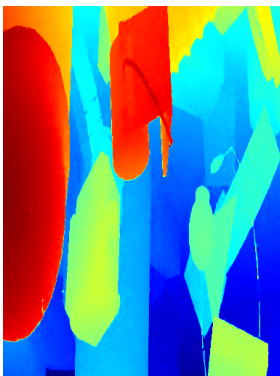
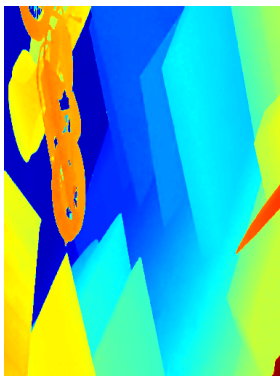
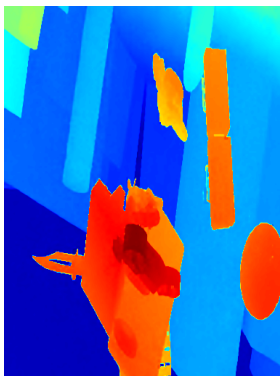
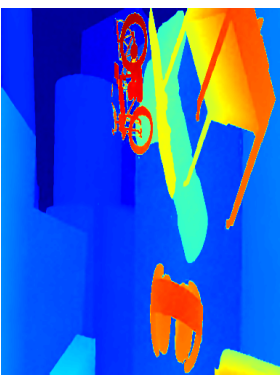

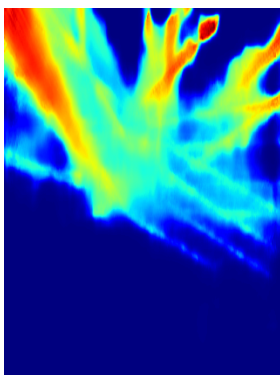
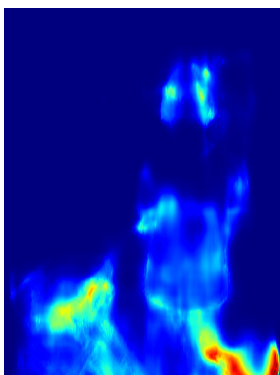
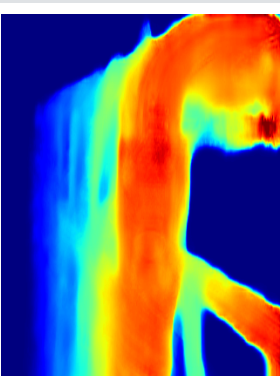
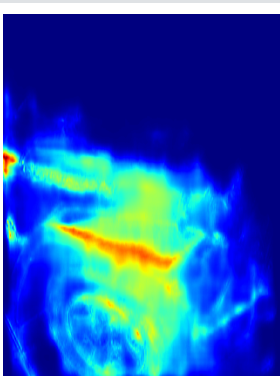
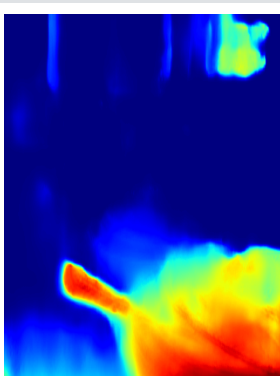
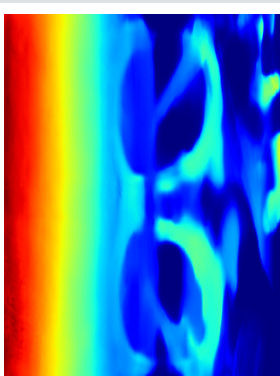
(1) Synthetic : Without Augmentation

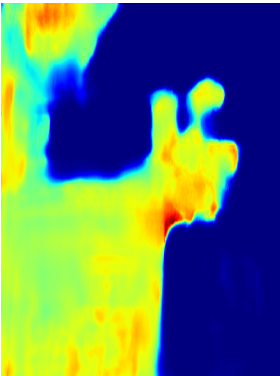
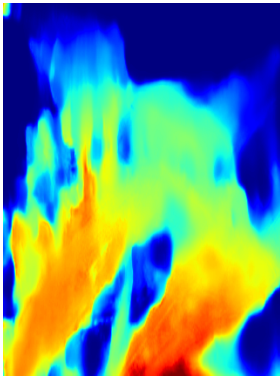
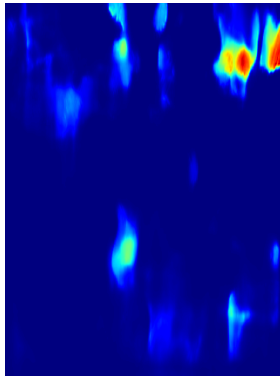
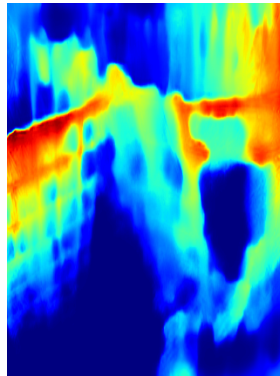
	TL0	TL1	TL2	TL3	TL4	TL5	TL6	TL7	TL8	TL9	Avg
Error	0.67	0.98	0.67	0.66	1.21	0.64	0.43	0.64	0.80	1.68	0.84

(2) Synthetic : With Augmentation

	TL0	TL1	TL2	TL3	TL4	TL5	TL6	TL7	TL8	TL9	Avg
Error	1.56	1.02	1.25	1.11	1.13	0.96	1.55	1.04	1.60	1.09	1.23

(3) Disparity Map

Synthetic_TL0	Synthetic_TL1	Synthetic_TL2	Synthetic_TL3
			
Synthetic_TL4	Synthetic_TL5	Synthetic_TL6	Synthetic_TL7
			
Synthetic_TL8	Synthetic_TL9	Real_TL0	Real_TL1
			
Real_TL2	Real_TL3	Real_TL4	Real_TL5
			

Real_TL6	Real_TL7	Real_TL8	Real_TL9
			

5. Reference

- (1) Godard, Clément, Oisín Mac Aodha, and Gabriel J. Brostow. "Unsupervised monocular depth estimation with left-right consistency." *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.
- (2) Jia-Ren Chang and Yong-Sheng Chen. "Pyramid Stereo Matching Network." *2018 CVPR*

6. Execution

0. Machine Spec

```
Centos 1080Ti x 1
```

1. Install the pretrain model

```
bash install.sh
```

2. Requirement

```
torchvision==0.2.1
tensorflow==1.1.0
opencv_python==3.4.4.19
matplotlib==3.0.1
numpy==1.15.4
torch==0.4.1
scipy==1.1.0
```

3. Compute disparity!

```
python main.py --input-left <path to left image> --input-right <path to right image>
--output <path to output PFM file>
```

7. Experience

歷經6天的CV final camp，我和楊甫每天過著晚上通話到兩三點、早上九點準時起床上GRE的日子，在睡前固定開啟tmux讓他train，期待早上看到的成果，睡覺的時候心裡總想著我們的模型到底有什麼問題，有什麼地方弄錯了。過程真的很辛苦，好幾次成果都讓我們很絕望，懷疑到底要不要回歸傳統的方法，但奇蹟總是在我們陷落谷底時出現，往往一個小調整，整個performance就會大爆衝。很感謝我的隊友還有助教、老師，我從剛踏進教室時是個連numpy都不會的菜雞，到現在可以跟別人嘴的一口好CV了xD！