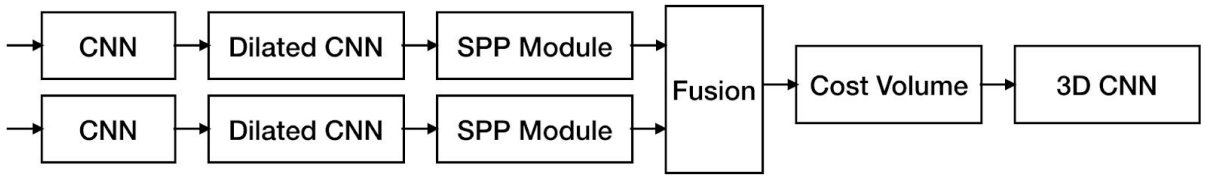
Final Project Report - Stereo Matching

R07921052 電機所碩一 劉兆鵬
R06921081 電機所碩二 張邵瑀
R06921058 電機所碩二 方浩宇

1. Synthesis Data



在這次的合成資料，我們採用 Deep learning 的方法進行預測，由於此次的合成資料 Max disparity number 較大，若採用作業四的 stereo matching 方法會造成計算時間較久的情情方法，故我們採用深度學習的方式使計算圖片的 Disparity map 的速度大幅降低，模型架構圖如上所示。

我們所採用的深度模型架構為 Pyramid Stereo Matching Network(PSMNet)，首先我們在特徵層的部分採用連續三個較小的 convolution filter 來取代一個大的 convolution layer，因為使用三個小的 convolution filter (3X3) 會等同學使用一個 (7X7) filter 的 receptive field，且能夠使模型更加深層化。在經過一般的 convolution layer 後我們接著採用 Dilated convolution layer 為的是能夠增加 receptive field 使得模型在特徵層能夠抓取更豐富的圖片特徵資訊。

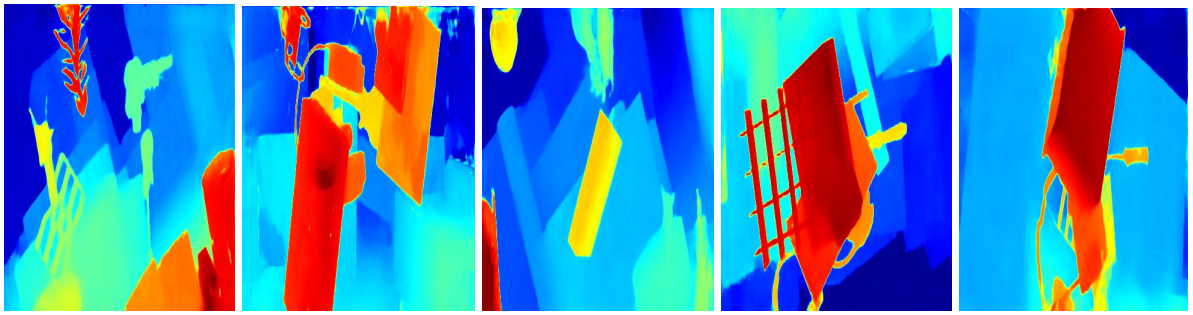
然而，在使用一般的 convolution layer 在越後面層所萃取的輸出維度會隨之減少，且在每一次的 MaxPooling layer 之後都會使得圖片的訊息跟著下降，為了要使得圖片能夠同時保留模型各特徵層所學習到的特徵，我們採用 Spatial Pyramid Pooling Module(SPP) 來串接前後的特徵。我們會先將特徵層的輸出分為四個分支，由於最後一層的 Channel 數量較多，為了減少特徵的維度我們採用 (1X1) Inception unit 來減少特徵的 Channel，接著將四個分支使用不同的大小的 Pooling layer 進行特徵萃取後，再使用 bilinear 的 Upsampling 來使得各分支的輸出維度相同，並將各分支以及先前的 Convolution layer 的輸出特徵 Concat 在一起讓模型能夠同時保留著各特徵層的輸出內容。最後將左圖以及右圖都經由相同的 Convolution feature extraction layer 後將此兩張圖片的輸出 Concat 在一起形成初步的 Cost volume。

最後我們採用論文中 3D CNN 的 Stacked hourglass 的 3D CNN 架構進行 Disparity 的 Classifier，為了能夠抓取更多特徵內容的資訊，在此使用 Encoder-Decoder 的做法來得出適當的 Disparity map。在此 Encoder-Decoder 層串接了三個相同的維度的 Convolution layer，而每個輸出都作為一個 Disparity map 的輸出，此作法能夠加速模型的訓練，且能夠在模型訓練的過程中了解最後的 Encoder-Decoder 層在每一個階段所學習到的內容為何。而這三個輸出會以 weighted sum 來計算我們的 loss，而在最後測試時我們是以最後一個輸出作為我們最終的 Disparity map。

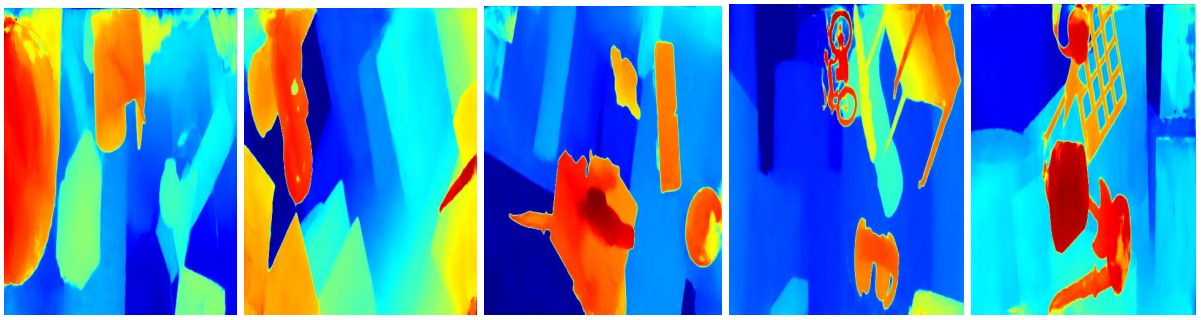
在訓練過程中，為了能夠使我們的模型能夠學習更多元的資訊，首先我們先使用 SceneFlow dataset 的 FlyingThing3D 進行模型的訓練，因為此資料集與我們的 Synthesis data 較為相像，做為我們的 pretrained weight，接著我們也使用 KITTI stereo 2015 進行訓練，為了讓模型能夠更多種不同的資訊內容。最後我們再以此 pretrained weight 作為 model 的 initializer，將我們 Synthesis dataset 進行訓練，而為了使得模型能夠保留先前所學習的資訊，在此我們只以我們的 Synthesis dataset 進行 fine-tune，我們

將 learning rate 設置為 0.001 並採用 learning rate decade 的方式進行參數調整，只為了能夠使得模型最後輸出值的範圍能夠與我們 data 的 label 相近避免計算 error 時產生太大誤差。

Disparity map 結果圖



TL0 TL1 TL2 TL3 TL4



TL5 TL6 TL7 TL8 TL9

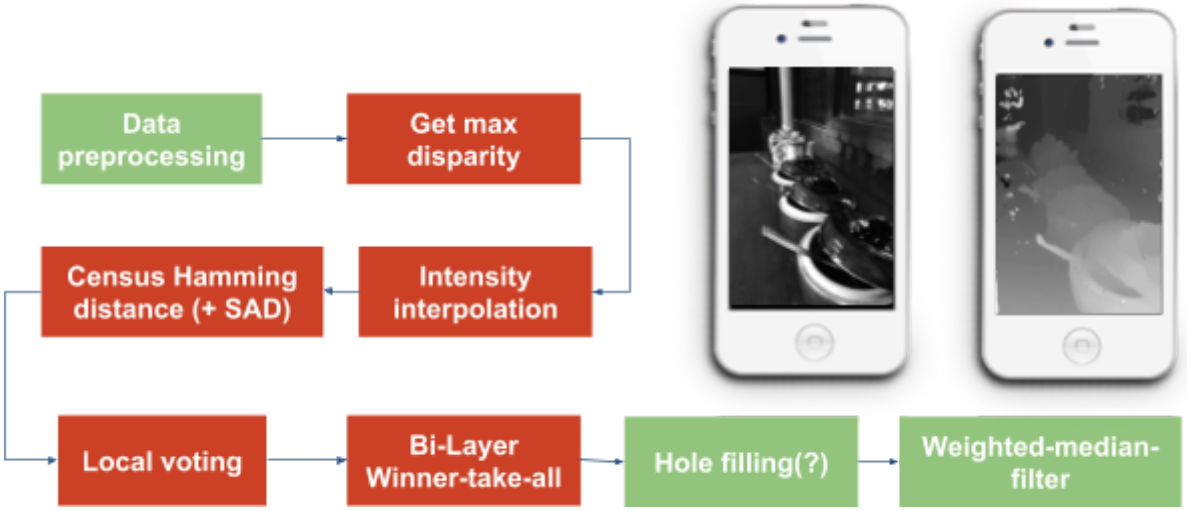
Disparity map bad pixel ratio

Img Name	TL0	TL1	TL2	TL3	TL4
Bad pixel ratio	0.673	0.967	0.720	0.719	0.767
Img Name	TL5	TL6	TL7	TL8	TL9
Bad pixel ratio	1.005	0.623	0.575	0.862	0.750
Average ratio	0.860				

Disparity map run time

Img Name	TL0	TL1	TL2	TL3	TL4
Run time	12	12	13	13	12
Img Name	TL5	TL6	TL7	TL8	TL9
Run time	12	13	12	13	12
Average time	12.4				

2. Real Data

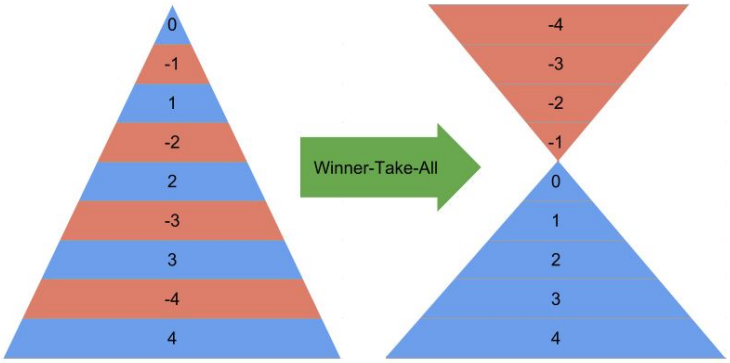


在 Real data 的部分，由於沒有如 SceneFlow 與 Synthesis data 等相像的訓練資料，故在此部分我們採用與作業四相像的方式進行 Disparity Map 計算。由於此次題目並沒有給予我們各張圖片所對應的 Max disparity number，故我們必須先自己計算出每一張圖片應計算多少 Disparity map 才足夠。由於原始的兩張圖片為黑白的，其各自的亮度都有稍顯誤差，為了使兩張圖的亮度能夠對應，我們首先將兩張圖都進行 Histogram equalization，將圖片的對比進行校正，接著又兩張圖片的明亮分佈不相同，我們計算出左圖以及右圖的 mean 與 variance 並改變右圖的分佈使得兩張圖片的擁有想同的 mean 以及 variance，接著我們使用 SIFT 來找出兩張圖片所相同的特徵位置，並統計圖片中相同的兩個對應點所在的 X 位置為何。而其中我們發現在做 SIFT 時會有對應點的距離為負的，這代表著說圖片在近景以及遠景的焦距是不同的，且由於兩張圖片的無窮遠處並非位於同個位置，導致在做 Stereo matching 時不能夠只考慮往同一個方向計算，必須要左右查看來得出結果。且有些圖片的特徵對應點的距離都為 0，這代表著兩張圖片的差距 過小，且小於一個 pixel 的距離，這會使得我們在計算 cost 時所有的對應點都於第一個 Disparity 時就得出最小的 cost。為了解決此問題，我們採用 Interpolation 的方式解決此問題，根據 SIFT 所得出對應的差距進而判斷此輸入圖片需要放大幾倍才能夠使得 Disparity 能夠正常計算。

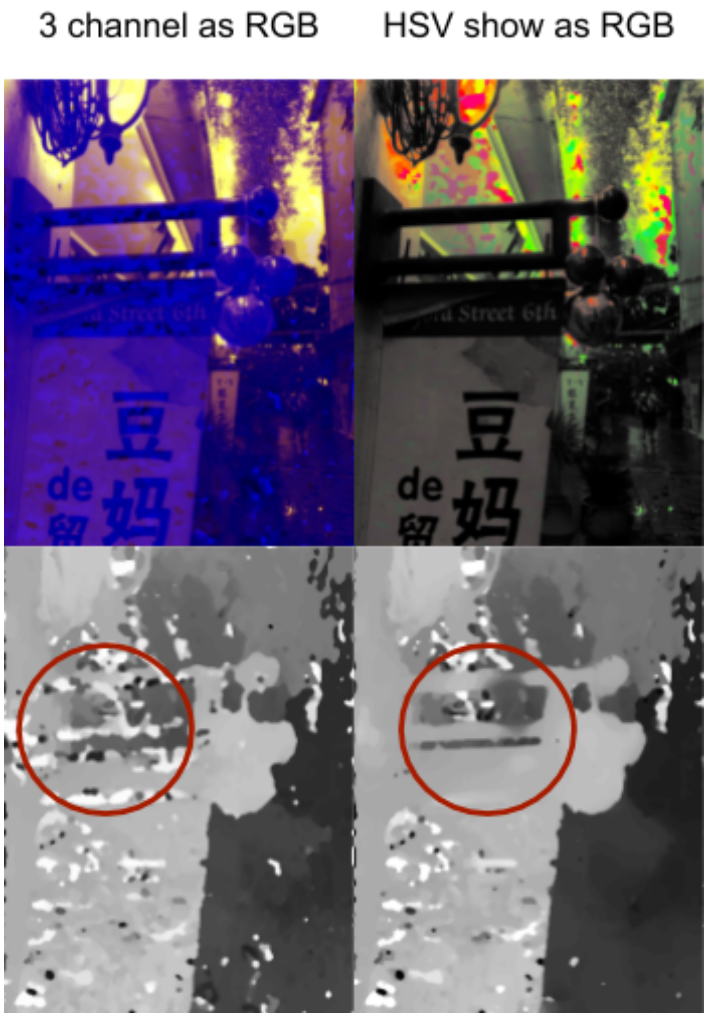
在將圖片放大後我們也根據原先計算的 SIFT 最大以及最小值作為我們向右查看與向左查看的 Disparity 數量。再將圖片放大後，我們也會在對圖片進行顏色的矯正，使得兩張圖片的色彩分佈較為相近，便於根據 cost 找出對應的物件資訊。

由於此次的 Real data 為黑白的，並非如作業四使用彩色的圖片，故會導致圖片的在計算 cost 時候數值範圍縮小，導致在做 winner take all 時發生錯誤。為了改善此問題我們採用 Census 以及 Hamming distance 作為我們的 cost 計算，如此一來能夠排除掉使用圖片數值所得出的資訊，並且能夠以圖片各物件的結構資訊作為 cost 的計算，就能夠解決數值範圍較小的問題。

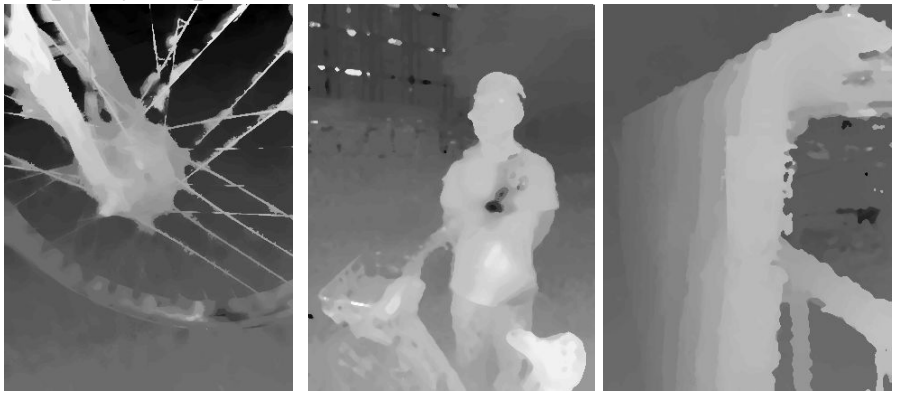
由於我們在計算 cost 時採用向左以及向有掃描，故在 winner take all 的部分我們也採用兩者所產生的 cost 的整合計算，因為右圖為右邊的相機所拍攝，故向右掃描可視為左相機以及右相機的交點處前段，而向左掃描則為兩者的交點後段，成像的部分視為相反的，故我們將兩著的 cost concat 在一起做 WTA 時，需判別最小值是來自於哪一段 cost，若為向左掃描而來，我們則判定此為交點後段的影像，故我們會將此 Disparity 乘以 -1，若為交點前段的 Disparity 則為其對應的 Disparity number。最後我們將此 WTA 的結果根據其最小值將之推到 0 的位置，則表示為我們將兩圖的交點重設至無窮遠的位置。如下圖所示。

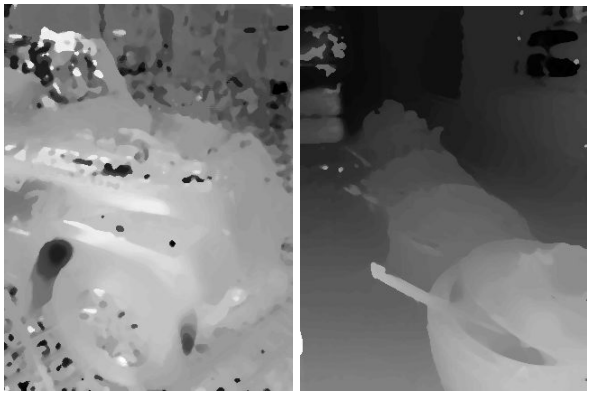


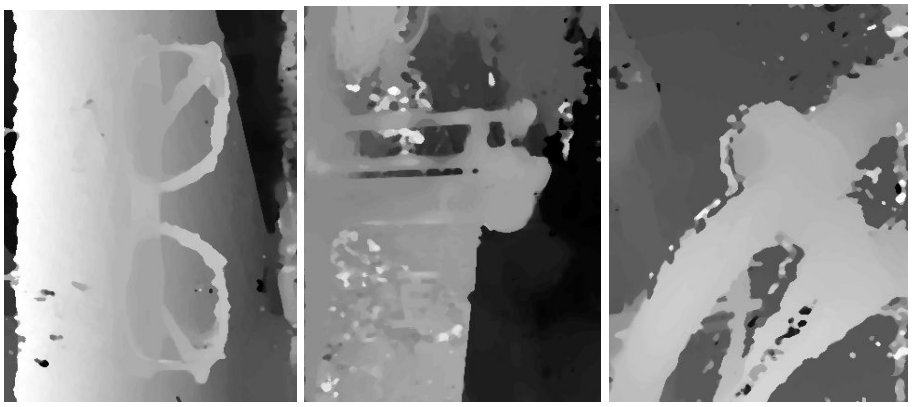
最後我們透過 weighted median filter 進行 refine，因為在做完 winner take all 後會有許多雜訊的產生，為了能夠消除此些雜訊並保留著原先物件得邊緣資訊，故在此需要根據不同的 weight 來進行 median filter，在此我們採用 guided filter 作為 weight 的資訊來進行運算使得輸出圖片能夠更加平滑且包含其各物件的深度資訊，但由於weighted median filter是對物件的顏色差異較為敏感，用題目提供的灰階照片並不能達成相同目的，所以我們採用的方式是對我們手上擁有的資訊來產生新的color space guide 影像，對於影像做weighted median filter下圖為做color space轉換前與轉換後的同一個guide image的結果。

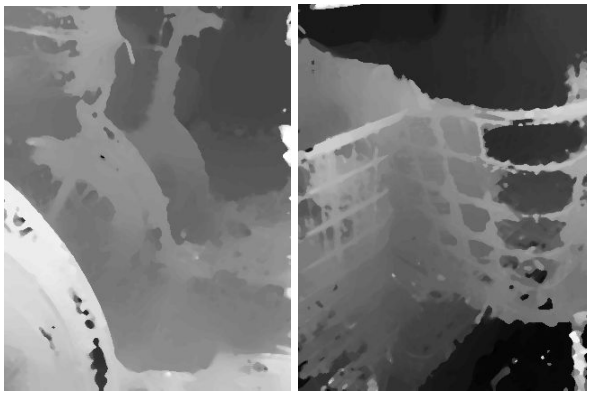


Disparity map 結果圖









Real data disparity map 運算時間表

Disparity map run time

Img Name	TL0	TL1	TL2	TL3	TL4
Run time	164	617	110	601	72
Img Name	TL5	TL6	TL7	TL8	TL9
Run time	61	243	72	183	74
Average time	219.7				

3. contribution:

R07921052 劉兆鵬：Deep model training, Researching, interpolation 構想

R06921081 張邵瑀：主要Real data演算法, refinement

R07921052 方浩宇：文件編寫， segmentation