# Joint stereo matching and semantic segmentation with CNN

CHEN Ran
Chinese University of Hong Kong
chenran1995@link.cuhk.edu.hk

## Abstract

*Leveraging the recent advances in convolutional neural networks, stereo matching is casted as a learning problem, surpassing the traditional approaches by a large margin. However, as the problem of stereo matching is inherently ambiguous at object occlusions, it is still challenging to correctly recover the depth of regions that are fully occluded in one input view. On one hand, known that disparity is a representation of the 3D scene, the disparity values within an object segment are strongly related, thus with semantic segmentation, the disparity on such regions can be greatly improved. On the other hand, the results of stereo matching can also improve semantic segmentation on their hard cases, e.g., texture regions. We believe this work may not only demonstrate a novel synergy between the areas of stereo matching and image segmentation, but may also inspire new cross-domain solutions between low-level vision and high-level vision.*

## 1. Introduction

Convolutional neural networks have become the method of choice in many fields of computer vision. There have been a lot of works done on semantic segmentation for per-pixel predictions. Depth estimation from joint stereo is another interesting topic which can widely be implemented in the networks.In this paper, we try to combine these two tasks with the idea of multi-task to improve the accuracy of the segmentation.

The reasons why we use multi-task to solve this problem: 1) *Attention focusing* - MTL can help the model focus its attention on those features that actually matter as other tasks will provide additional evidence for the relevance or irrelevance of those features. 2) *Eavesdropping* - Some features G are easy to learn for some task B, while being difficult to learn for another task A. This might either be because A interacts with the features in a more complex way or because other features are impeding the model's ability to learn G. Through MTL, we can allow the model to eavesdrop. 3) *Representation bias* - MTL biases the model to prefer representations that other tasks also prefer.

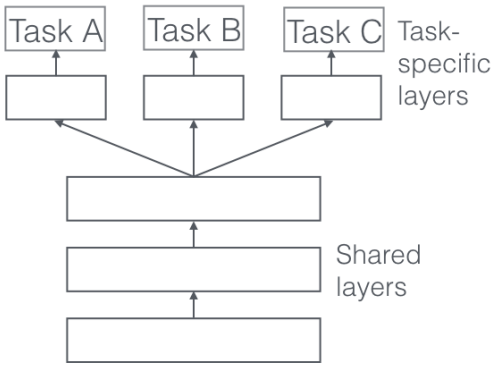


Figure 1. Hard parameter sharing for multi-task learning in deep neural networks

There are two most commonly used ways to perform multi-task learning in deep neural networks.In the context of deep learning, multi-task learning is typically done with either hard or soft parameter sharing of hidden layers. Since it was not clear which layer features are familiar we choose hard parameter sharing, because this structure (Fig.1) is much suitable and visualized for our networks.

Training such a network to predict the depth and segmentation requires a sufficiently large training set. Although data augmentation does help, the existing real-world dataset are still too small to train a network on par with state of the art. Trading in realism for quantity, we pretrain our model with a synthetic Flying Chairs dataset which consists of random background images.The benchmark of our multi-task network result looks reasonable.

## 2. Related Work

**Multi task.** In multi task learning for computer vision, approaches often share the convolutional layers, while learning task specific fully-connected layers[13] improve upon these models by proposing Deep Relationship Networks, which allow the model to learn the relationship between tasks.

Starting at the other extreme [8] propose a bottom-up

approach that starts with a thin network and dynamically widens it greedily during training using a criterion that promotes grouping of similar tasks.

[16] put forward an idea with two separate model architectures just as in soft parameter sharing. They use what they refer to as cross-stitch units to allow the model to determine in what way the task-specific networks leverage the knowledge of the other task by learning a linear combination of the output of the previous layers. We realize this idea in our model and will be introduced in section 4.

Besides learning the structure of sharing[8] take an orthogonal approach by considering the uncertainty of each task. They adjust each task's relative weight in the cost function by deriving a multi-task loss function.

There are also many other authors applied multi task learning in other area such as natural language processing and get the impressive result. We will not list here.

**Semantic segmentation.** Nowadays, semantic segmentation has widely been applied to 2D images, video, and even 3D data. And it is one of the key problems in the field of computer vision. Looking at the big picture, semantic segmentation is one of the high-level task that paves the way towards complete scene understanding.

We review some recent advances in semantic segmentation task, Driven by power deep neural networks[1, 11, 21, 8] pixel- level prediction tasks like semantic segmentation achieve great progress inspired by replacing the fully-connected layer in classification with the convolution layer [12]. To enlarge the receptive field of the neural networks, methods of [22, 21] used dilated convolution and got nice performance.

Other works mainly proceeds in two directions. One[10] is with multi-scale feature ensembling. Since in deep networks, higher layer feature always contains more semantic meaning and less location information. The Combination of multi-scale features can improve the performance.

The other direction is based on structure prediction. A tentative work [1] used conditional random field (CRF) as post processing to refine the segmentation result. However, our experiments show that this method is depends on type of database. [19] refined networks via end-to-end modeling. Both of the directions improve the segmentation in complex scenes.

## 3. Network Architectures

**Convolutional Networks.** Convolutional neural networks with the method of backpropagation have recently been shown to perform well on large-scale image classification by Krizhevsky *et al*. [9]. This gave the beginning to a surge of works on applying CNNs to various computer vision tasks.

There has been some works on estimating optical flow with CNNs, the FlowNet[4] is a recent work on estimating image flow between two frames in a video. The input of this network are two images with the small motions. They combine this two separate, yet identical processing streams for the two images and to combine them at a later stage as shown in Fig. 2.

We get the idea from multi task learning and the FlowNet. Construct a one encoder with two decoders network. As shown in Fig 3. To realize the stereo matching, we use the 'correlation layer' that performs multiplicative patch comparisons between two feature maps. An illustration of the network architecture 'FlowNetCorr' contains this layer which is shown in Fig. 2. Given two multi-channel feature maps *f1*,*f2*: $\mathbb{R}^2 \rightarrow \mathbb{R}^c$, with $w$,$h$, and $c$ being their width, height and number of channels, the correlation layer lets the network compare each patch from *f1* with each path from *f2*.

Now we explain it with only a single comparison of two patches. The 'correlation' of two patches centered at $\mathbf{x}_1$ in the first map and $\mathbf{x}_2$ in the second map is then defined as

$$\mathrm{c}(\mathbf{x}_1, \mathbf{x}_2) = \sum_{o \epsilon [-k,k] \times [-k,k]} \langle \mathbf{f}_1 \left( \mathbf{x}_1 + o \right), f_2 \left( \mathbf{x}_2 + o \right) \rangle$$

for a square patch of size $K := 2k + 1$. Note that this equation is identical to one step of a convolution in neural networks, but instead of convolving data with a filter, this layer convolves data with other data. So, it has no trainable weights.

Considering that it involves $c \cdot K^2$ multiplications. Comparing all patch combinations involves $w^2 \cdot h^2$ such computations, yields a large result and makes efficient forward and backward passes intractable. So, as the computational reasons, the correlation layer will do a limited comparisons.

Given a maximum displacement $d$, for each location $\mathbf{x}_1$ we compute correlations only in a neighborhood of size $D := 2d + 1$, by limiting the range of $\mathbf{x}_2$.

Back to our model, this two tasks share the encoder part to learn the features, and decode the features independently. At the backpropergation period, two tasks update the features, as a mutual correction to help each other to do better. Why we believe these two tasks can assist each other to get better performance?

The depth information can give us a clear border between different objects. In segmentation task, the difficulty that most of research meet is the confusion of the objects with similar features but different depth. So, we consider this model can do some optimization on this problem.

**ResNet.** After we verify the idea on our model. We applied our novel idea to the state-of-the-art convolutional network structure called ResNet [5] to check whether our methods can improve the best result at present. When deeper networks starts converging, a degradation problem has been exposed: with the network depth increasing, accu-

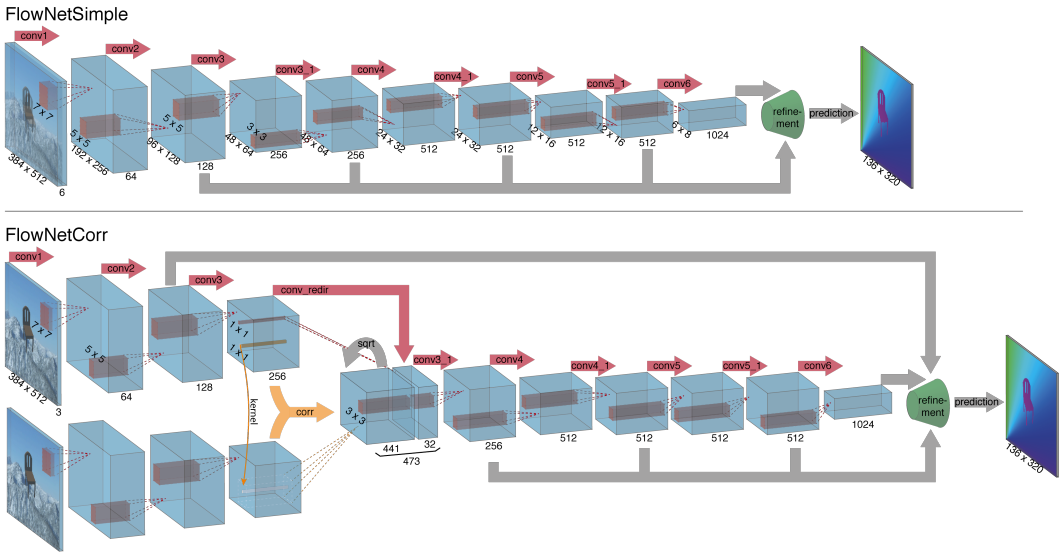


Figure 2. The two network architectures: FlowNetSimple (top) and FlowNetCorr (bottom).

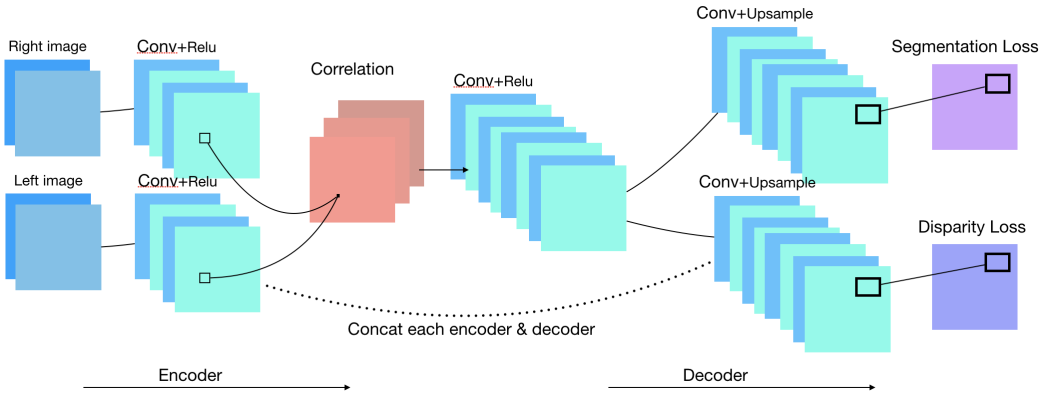


Figure 3. Our Single Encoder with double Decoders network.

racy gets saturated and then degrades rapidly. The residual structure, see figure 4, solved this problem and got widely used in recent computer vision tasks. We construct a multi-task ResNet to do our further research.

## 4. Training Data

As we mentioned before, the amount of dataset which can provide depth information and segmentation ground truth are quite small. Unlike traditional approaches, neural networks require data with ground truth not only for optimizing several parameters, but to learn to perfom the task from scratch. In general, obtaining depth ground truth is hard, because true pixel correspondences on stereo matching for real world scenes cannot easily be determined.The dataset we use to train is given in Table 1.

| Dataset | Cityscapes |
|---|---|
| Complexity | 19 classes |
| Volume | 5000 annotated images with fine annotations |
| Type of annotations | Semantic ,Instance-wise<br>Dense pixel annotations |
| Origin Size | 1024*2048 |

Table 1. Information of Cityscapes dataset

### 4.1. Cityscapes

The Cityscapes Dataset is a recently released dataset for semantic urban scene understanding.It focuses on semantic understanding of urban street scenes. It provide about 3000 image pairs for training and 500 for validation. We do not use the coarse images to do our training, because it cannot provide effective segmentation for pretrain and does not give us the ground truth of the depth.

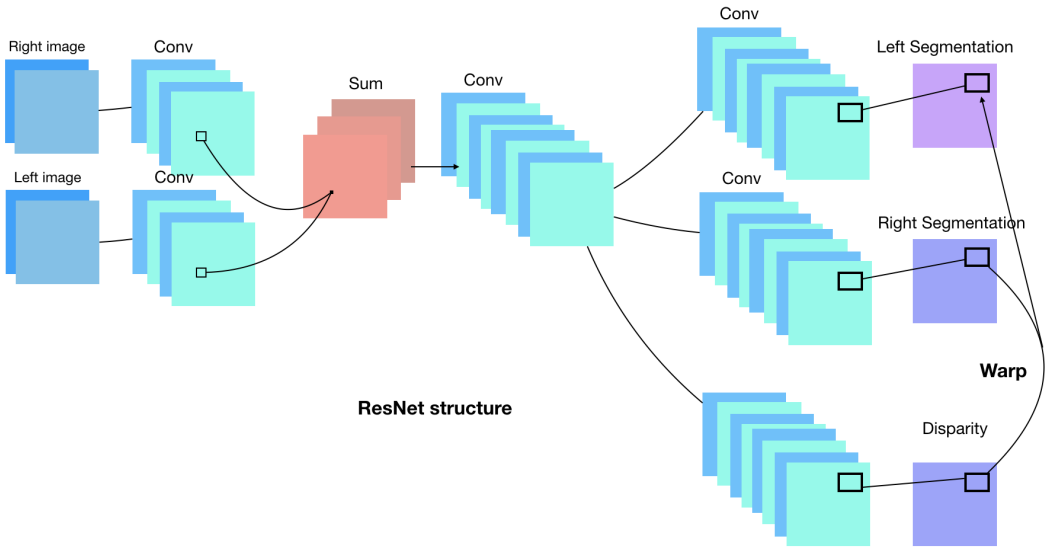


Figure 4. Our ResNet structure with warp methods

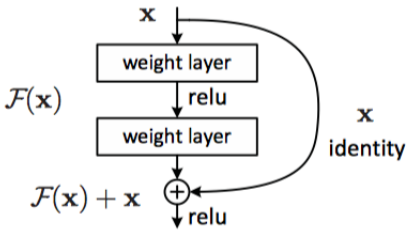


Figure 5. Residue structure

### 4.2. Data Augmentation

Data Augmentation is a widely used strategy to improve generalization of neural networks. We perform augmentation online during training the network. The augmentations we use include scaling, mirror,additive Gaussian noise and changes in brightness, constract,gamma and color. We do not use translation and rotation, because they do not conform to the real world data type. Limited by the GPU memory and batch size, we can not input the origin size of the data. We tried several input strategy to guarantee the training efficiency and exploring the way to get better performance: 1) crop the image into size of 512*512 or 1024*1024 and package them into origin size. 2) do random search on the origin size, enlarge the dataset manually. These two ways both do better than the pixel compression. High resolution do better in our task.

## 5. Experiments

Besides the regular training strategy and parameters adjustment. We also tried some algorithms to get higher mark or accelerate the training period. In our encoder-decoders network, we applied a feature sharing methods called alpha layer. In ResNet, we applied warp method, which compensate for some already estimated preliminary motion in the second image.The concept of image warping is common to all contemporary variational optical flow methods and goes back to the work of Lucas & Kanade[14]

### 5.1. Startup

Before we do experiments on Cityscapes dataset. We tried to implement our idea on a manual portrait dataset made by ourselves to check whether it works or not. And the result is perfect. See figure 6. and figure 7. The edge becomes much smoother with the disparity information.

### 5.2. Alpha layer

We get the idea of the alpha layer from the [16] Cross-stitch Networks. Alpha layer is a new unit layer which combines these two networks into a multi-task network in a way such that the tasks supervise how much sharing is needed, as illustrated in Fig 8.At each layer of the network, alpha layer learns a linear combination of the top blobs $x_A$,$x_B$ from the recent layer for both the tasks, we learn the linear combinations $\widetilde{x}_A$,$\widetilde{x}_B$ (Eq 1) of both the input activations and feed these combination is parameterized using $\alpha$. Specially, at

Figure 6. The output of segmentation image without disparity information.

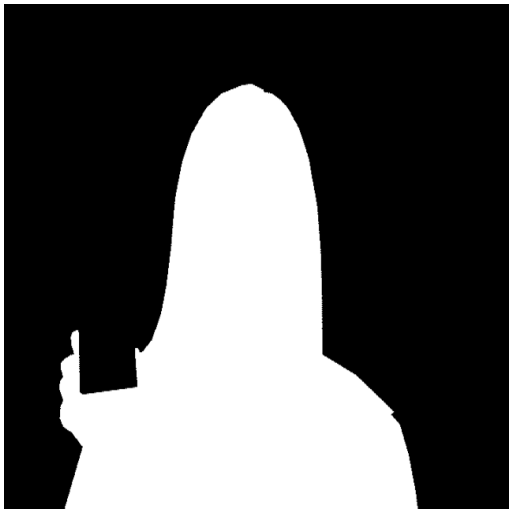

Figure 7. The output of segmentation image with disparity information.

the location $(i,j)$ in the activation map,

$$\begin{bmatrix} \widetilde{x}_A^{ij} \\ \widetilde{x}_B^{ij} \end{bmatrix} = \begin{bmatrix} \alpha_{AA} & \alpha_{AB} \\ \alpha_{BA} & \alpha_{BB} \end{bmatrix} \begin{bmatrix} x_A^{ij} \\ x_B^{ij} \end{bmatrix}$$

As our network is a symmetry structure, this layer can be directly applied in each layer at decoder. Avoiding the mismatching of the inputs.

**Backpropagating through alpha layer.** Since the alpha layer is modeled as linear combination, their partial derivatives for loss $L$ with tasks A,B are computed as

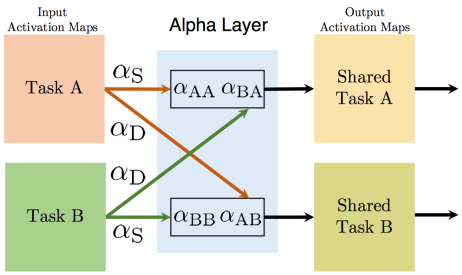


Figure 8. The Alpha Layer structure.

$$\begin{bmatrix} \frac{\partial L}{\partial x_A^{ij}} \\ \frac{\partial L}{\partial x_B^{ij}} \end{bmatrix} = \begin{bmatrix} \alpha_{AA} & \alpha_{BA} \\ \alpha_{AB} & \alpha_{BB} \end{bmatrix} \begin{bmatrix} \frac{\partial L}{\partial \widetilde{x}_A^{ij}} \\ \frac{\partial L}{\partial \widetilde{x}_B^{ij}} \end{bmatrix}$$

We denote $\alpha_{AB}$,$\alpha_{BA}$ by $\alpha_D$ and call them the different task values because they weigh the activations of another task. Likewise, we name the $\alpha_{AA}$,$\alpha_{BB}$ as $\alpha_S$, which means the same task values, since they weight the activations of the same task.As we cannot make sure the proportion between this two parameters. We can vary $\alpha_D$ and $\alpha_S$ values, the layer can freely move between shared and task-specific representations, and choose a middle ground if needed.

### 5.3. Conditional Random Fields

We also tried to combine the CNN with tradition method to see what it can bring to us. But in fact, after a tentative practice, the result of the CRF cannot match the benchmark of the Cityscapes dataset. We will not focus on this method at this moment.

### 5.4. Warp method

During the training with left, right and disparity ground truth, we can directly use the disparity map to pair the pixels in left and right images $\langle x, y\rangle \leftarrow \langle x+u, y+v\rangle$ ,$left\,\langle u, v\rangle$ is the value of the disparity map, in our experiments, $v$ is zero because there is no vertical movement.

**Algorithm 1** Warp method

**input** Left, Right image pairs & Disparity map.
In last layers before outputs
**for** each pixel $\langle x, y\rangle$ in right image features **do**
$left\,\langle x, y\rangle = left\,\langle x, y\rangle + right\,\langle x+u, y+v\rangle$
$output = softmax(left\,\langle x, y\rangle)$

The reason why we use this method is that we believe the right image can provide further information and confidence to the left image and improve the result.

| Method | road | sidewalk | building | wall | fence | pole | traffic light | traffic sign | vegetation |
|---|---|---|---|---|---|---|---|---|---|
| 1024*1024 Seg_Disparity . IoU | 0.97 | 0.773 | 0.895 | 0.355 | 0.451 | 0.574 | 0.557 | 0.674 | 0.904 |
| 1024*1024 Seg_only . IoU | 0.963 | 0.733 | 0.878 | 0.281 | 0.358 | 0.522 | 0.474 | 0.605 | 0.894 |
| 1024*1024 Seg_Disparity . iIoU | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 1024*1024 Seg_only . iIoU | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |

| Method | terrain | sky | person | rider | car | truck | bus | train | motorcycle | bicycle |
|---|---|---|---|---|---|---|---|---|---|---|
| 1024*1024 Seg_Disparity . IoU | 0.543 | 0.930 | 0.711 | 0.416 | 0.912 | 0.388 | 0.580 | 0.410 | 0.286 | 0.679 |
| 1024*1024 Seg_only . IoU | 0.502 | 0.929 | 0.652 | 0.287 | 0.888 | 0.311 | 0.495 | 0.235 | 0.221 | 0.625 |
| 1024*1024 Seg_Disparity . iIoU | NaN | NaN | 0.519 | 0.275 | 0.827 | 0.178 | 0.366 | 0.241 | 0.186 | 0.451 |
| 1024*1024 Seg_only . iIoU | NaN | NaN | 0.464 | 0.186 | 0.800 | 0.141 | 0.290 | 0.156 | 0.141 | 0.392 |

Table 2. Per-class result of our multi task network and single segmentation baseline network.

* IoU commonly known as the PASCAL VOC intersection-over-union metric $IoU=\frac{TP}{TP+FP+FN}$, where TP, FP, and FN are the numbers of true positive, false postive, and false negative pixels.
* iIoU is used to evaluate the semantic labeling using an instance-level intersection-over-union metric $iIoU=\frac{iTP}{iTP+FP+iFN}$. Again iTP, FP, and iFN denote the numbers of true positive, false positive, and false negative pixels,respectively.

## 5.5. Results

### 5.5.1 Our model

Table 2. shows the IoU and iIoU of our networks and the baseline network. We can see that each class has got promoted after the hard parameter sharing with the FlowNet. Especially, some low score classes got great improvement: truck - 6.9%,traffic light - 8.3%,traffic sign - 6.9%,fence - 9.3% *et al*. Our network gets 6.1% on average IoU better than the baseline model 57.1%.

We also applied the Alpha Layer into our decoder side at last convolution layer with different initial value of $\alpha_D$ and $\alpha_S$ to see whether an intuitive information interchange can help our model. From the result, the alpha layer raise up the training period to half and got the same level as multi task network.

### 5.5.2 ResNet model

We use the disparity ground truth directly to combine warped right result and the left result which generated by the ResNet. This is the expected upper bound of this task. But we found that the fused result got 0.67% lower than origin single left image result, which means the features from the right images do not work.

# 6. Problem Analysis

## 6.1. Mismatched Error

After analysis, we find out that if we use the ground truth depth map to warp the image, there maybe some mismatch error. The following image is the sample result of the warped right output. See figure 9. The white parts are the ignore points of the depth ground truth. These ignore points are always the hidden part which cannot be matched in stereo images. Most of them appear on the left border of an object. In this image, the area in the left circle is the wrong prediction warped from the right image's output. About 70 columns of pixels on the left of the right image are random and there are some meaningless depth values of one at the corresponding columns in the disparity map which copy the random error directly. After warping, these cause the wrong prediction.

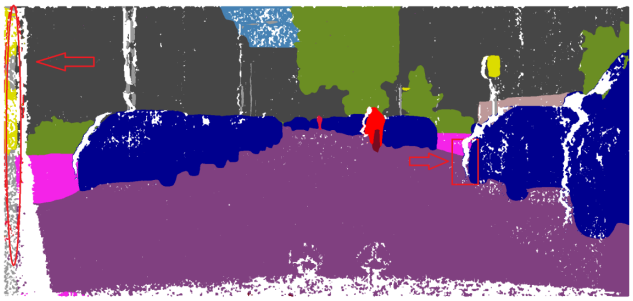

Figure 9. Sample of the warped right output

The area in the right rectangle is the hidden regions which cannot be matched. We find that near the ignore part, there are isolated blue points which represent cars. This may be caused by the inaccuracy of the right part.

To solve the problems, we mask the first 60 rows of the image and use dilation method to expand the ignore regions and reduce errors. The fused result is almost the same as the origin but no benefit. The results are showed in the following table.

As we mentioned before, we assumed that the stereo

| Network | Method | Score(%) | Difference(%) |
|---|---|---|---|
| ResNet | left only (Baseline) | 69.91 | |
| | right only | 67.13 | -2.78 |
| | bilinear warp on right image | 69.55 | -0.36 |
| | warp right with mask first 60 columns | 69.77 | -0.14 |
| | warped right with 8*8 dilation + mask 60 | 70.72 | 0.81 |
| | fused result with 8*8 dilation + mask 60 | 69.88 | -0.03 |

Table 3. The result of warp method with masking

matching benefits the contour or boundary of the segmentation, but comparing to our manual portrait dataset, the realistic scenes are more complex and with much more miss matched part. And after the dilation operation, we focus less on boundary so that less extra features and information can be transferred from the right image.

### 6.2. Ground truth of disparity map

The ground truth of the dataset is also a potential problem we considered during the research. The depth ground truth we got from the Cityscape dataset is not robust. So we tried many different algorithms such as CPM algorithm[6], SVS[15] and CRL[17] to generate disparity map and do the test again, but all of them cannot provide further improvement. The mismatch points and bias still exist.

## 7. Conclusion

Although the hard parameter sharing works on our model, we cannot make a conclusion that this idea take effects on those state of the art model like ResNet. It's reasonable to consider that the multi-task on segmentation and disparity is still an intractable problem as there are still some prior problem we have to solve. If we want to make progress on this topic, we should solve the occlusion issue first. The conclusion from [23] also verify our guess.

## Acknowledgments

Thanks for Xiaoxiao Li's suggestions and advice during our research. This project trained on GTX1080 with Caffe[7] framework.

## Reference

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, June 2016.

[3] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016.

[4] A. Dosovitskiy, P. Fischery, E. Ilg, P. Husser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pages 2758–2766, December 2015.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Yinlin Hu, Rui Song, and Yunsong Li. Efficient coarse-to-fine patchmatch for large displacement optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5704–5712, 2016.

[7] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[8] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv preprint arXiv:1705.07115*, 2017.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] Sulin Liu, Sinno Jialin Pan, and Qirong Ho. Distributed multi-task relationship learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 937–946. ACM, 2017.

[11] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1377–1385, 2015.

[12] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[13] Mingsheng Long and Jianmin Wang. Learning multiple tasks with deep relationship networks. *arXiv preprint arXiv:1506.02117*, 2015.

[14] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[15] Yue Luo, Jimmy Ren, Mude Lin, Jiahao Pang, Wenxiu Sun, Hongsheng Li, and Liang Lin. Single view stereo matching. *arXiv preprint arXiv:1803.02612*, 2018.

[16] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 3994–4003, June 2016.

[17] Jiahao Pang, Wenxiu Sun, JS Ren, Chengxi Yang, and Qiong Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *International Conf. on Computer Vision-Workshop on Geometry Meets Deep Learning (ICCVW 2017)*, volume 3, 2017.

[18] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[21] Jonas Uhrig, Marius Cordts, Uwe Franke, and Thomas Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. In *German Conference on Pattern Recognition*, pages 14–25. Springer, 2016.

[22] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[23] Amir Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. *arXiv preprint arXiv:1804.08328*, 2018.