Disparity Estimation in Stereo Images

Shreyas Chaudhari
ECE Dept.
Carnegie Mellon University
Pittsburgh, PA 15213
`shreyasc@andrew.cmu.edu`

Asish Gumparthi
ECE Dept.
Carnegie Mellon University
Pittsburgh, PA 15213
`agumpart@andrew.cmu.edu`

Joshua Liu
ECE Dept.
Carnegie Mellon University
Pittsburgh, PA 15213
`jxliu@andrew.cmu.edu`

Dennis Loevlie
ChE Dept.
Carnegie Mellon University
Pittsburgh, PA 15213
`dloevlie@andrew.cmu.edu`

Abstract

Computer vision applications such as autonomous driving, 3D model reconstruction, and object detection and recognition rely heavily on depth estimation from stereo images. Recent state-of-the art work has shown that depth estimation can be formulated as a supervised learning task using Convolutional Neural Networks (CNNs) that process a pair of stereo images. There have been numerous recent proposed methods to tackle this challenge, but perhaps the most fundamental and well-known is the Pyramid Stereo Matching Network (PSMNet) (1). In this project we investigate the various architectural modules employed by PSMNet, and augment their findings through additional methods and experiments of our own. These include, but are not limited to, architecture modifications, parameter reduction techniques, and extending the work for disparity estimation from infrared (IR) images. A key novelty in our work is applying techniques learned from optimizing depth estimation from RGB sequences to IR sequences, since the latter has yet to be thoroughly addressed in literature. Specifically we achieve a 53.65 % reduction in parameters of the original PSMNet model and better performance on IR images, while consuming 14% less memory and 65% the inference time. Furthermore, we provide ablation study results that can guide future development of IR stereo disparity estimation architectures. Ultimately, the goal of our project is to provide a contribution to the scientific community by enabling others to use the results of our findings in their own projects.

1 Introduction

Accurate and robust 3-D object detection is critical for the advancement of autonomous vehicles. Traditional methods of object detection for autonomous vehicles rely on expensive LiDAR sensors to generate point clouds that can be processed by a deep learning model (2). The task of 3-D object detection inherently relies on accurate depth maps of the scene of interest, which can provide meaningful features regarding the geometry of the 3-D environment. Monocular RGB/IR cameras are an inexpensive way to obtain disparity maps; however, the depth estimation from a single camera might be highly unreliable in dynamic environments such as driving in cities. Disparity maps obtained through stereo images are more accurate than those obtained from monocular cameras, and are inexpensive compared to LiDAR sensors (3). IR Thermal images are more robust than their RGB counterparts as they are invariant to lighting conditions during the day or night, whereas RGB achieves optimal performance in well-lit conditions. Furthermore, thermal cameras can provide clear images in adverse weather conditions such as fog, smoke, and haze, and are more robust even in the presence of glare from oncoming headlights. These reasons provide a clear motivation to use IR thermal cameras as a surrogate for LiDAR and to solve the challenge of disparity estimation in IR images. In this work we propose an efficient deep learning architecture for disparity estimation that leverages previously reported state-of-the-art methods.

While researchers have investigated object detection via disparity estimation in stereo RGB images quite extensively (2), (3), (4), there is still significant room for improvement in developing accurate depth maps given stereo IR images (5). Disparity estimation in IR stereo images is in fact more challenging than in RGB images due to the lack of texture and feature points in thermal pictures. For this reason, we propose investigating the performance of depth estimation and disparity map generation for IR stereo images. We demonstrate results using architectures that augment current state-of-the-art methods, such as PSMNet, to understand how individual components influence the network's behavior. We observe the modified network's performance on stereo pairs of IR images, and attempt to further augment our modified network to obtain favorable results on an IR dataset.

Formally, we define the disparity estimation problem as follows. Let $I_L,\ I_R$ be a pair of stereo images. Given a pair of rectified stereo images, the goal of depth estimation is to compute the disparity d for each pixel in the reference image. We define disparity as the horizontal displacement between a pair of corresponding pixels between I_L and I_R. For a given pixel (x, y) in I_L, if its corresponding point is at $(x - d, y)$ in I_R, then the depth of the pixel is given by $\frac{fB}{d}$. Here f is the camera's focal length and baseline B is the distance between the two camera centers (1).

2 Literature Review

Stereo matching is an active field of research first instigated by Hirschmuller et. al (6). Semi-global matching (SGM) (6) is the underlying algorithm used for cost aggregation by many current state-of-the-art models for computing the disparity maps. SGM defines a matching algorithm which leverages pixel-wise mutual information to find the similarity among pixels/blocks in the stereo pairs, which is then used to compute the global cost volume and the dense disparity maps. SGM performs approximations by pathwise optimization from eight directions and thus the total time to compute the disparity is usually linear with respect to the number of pixels in the images. Thus the inherent dependence of run time of such traditional methods relying on hand-crafted schemes to find correspondences in the stereo pairs impedes their use for real time applications.

With the advent of deep learning, many methods have proposed the use of deep neural networks for computing the matching cost (7). These methods usually split the task into sub-tasks of cost computation, cost aggregation, disparity computation and refinement. Such methods have shown substantial improvements over the traditional methods. Deep networks leverage the convolutional layers as features extractors to compute patchwise cost values and then disparity refinement methods are used to obtain the final disparity maps. The resultant systems are not end-to-end solutions, however, and the performance is often limited to the performance of the cost aggregation steps. The other problem with not having end-to-end models is the difficulty in training as the common cost aggregation methods used such as SGM are not differentiable. DispNet (8) was the first end-to-end trainable network which showed that disparity estimation can be formulated as a supervised learning task that can be solved using CNNs. DispNet uses a correlation layer to compute the similarity between the extracted feature maps from the stereo pairs and computed the cost. This was followed by GC-Net(9) which incorporated contextual information by adding more 3D convolutions and proposed a differential soft argmin operation to train the network.

Chang et. al. (1) claims current depth estimation architectures rely on patch-based Siamese networks, which lack the means to exploit context information for finding correspondence in ill-posed regions. To tackle this issue, they propose a Pyramid Stereo Matching Network. The network consists of two main modules: spatial pyramid pooling (SPP) and 3D CNN. They state that the SPP module takes advantage of global context information at different scales to form a 4D cost volume. The cost volume is then fed to a 3D CNN for cost volume regularization and disparity regression. The only drawback of such networks is the computational cost and memory required to achieve favorable accuracy in terms of absolute disparity error, as more 3D convolutions are required. In order to tackle this we propose parameter reduction and network pruning techniques in this paper. We summarize the aforementioned works in Table 1 below, and also provide the latency and error rates they exhibit.

The error rate is given in terms of the percentage of pixel outliers (to 3px) averaged over all ground truth labels.

While stereo matching networks for RGB images have been thoroughly studied, disparity estimation in IR images has yet to be fully addressed. One of the reasons for the lack of extensive research in IR stereo matching is the lack of a comprehensive dataset and evaluation benchmark like the KITTI vision suite(10). Until recently, one of the only datasets to have thermal-thermal stereo pair datasets was the CATS Benchmark (11). The authors in (11) evaluate the performance of some of previous state-of-the-art stereo matching algorithms like (6),(12) etc. on thermal-thermal pairs and report the poor performance of almost all the methods with average accuracies (based on 3-pixel error) ranging from 9-24%. The exceptional performance of most of the current deep learning models on the KITTI dataset can be attributed partially to the large synthetic RGB-stereo corpus - SceneFlow Dataset. These models have been trained extensively on the SceneFlow dataset and fine-tuned on the KITTI to achieve their current results. Zhu et al.(13) in their recent work propose a new stereo matching algorithm for infrared images which leverages the Guided Image filtering along with weighted cost aggregation. The weighted cost aggregation allows them to calculate the effect of the effect of all the pixels of an image rather than a small window based block based matching and helps compensate the lack of texture in IR images, But this also increases the computation time by a order of magnitude(avg. of 14.75s on CATS) thus making it inefficient for real time applications.

Table 1: Summary of Currently Reported Methods (for RGB images)

Ref.	Name	Model Description	Latency	Error
(1)	PSMNet	Spatial Pyramidal Pooling + 3D CNN	0.41 s	2.32 %
(14)	AANet	Downsampled Feature Pyramids + Multi-scale Cost Volumes	0.062 s	2.55 %
(8)	DispNet	CNN + 1D correlation layer + introduces Sceneflow dataset	0.06 s	4.34 %
(15), (6)	OCV SGBM	SGM + Birtchfield-Tomasi sub-pixel metric + 5 directions	1.1 s	10.86 %
(9)	GCNet	Siamese Network + Multi-Scale 3D Conv. + 3D Deconv.	0.9 s	2.87 %

3 Contributions

3.1 PSMNet Model Description

We use the PSMNet as a reference since its architecture is the foundation for several other stereo matching networks. The PSMNet architecture begins with a CNN that consists of basic residual blocks for learning binary feature extraction. The Spatial Pyramid Pooling (SPP) module is then applied to gather context information to form a feature map. The left and right feature maps are then concatenated into a 4D cost volume, which is then fed into a 3D CNN for regularization. Finally, regression is applied to calculate the output disparity map. An architecture overview can be seen in Figure 1.

The main advantages of PSMNet can be summarized as follows:

- An end-to-end learning framework for stereo matching without any post-processing
- Pyramid pooling module for incorporating global context information into image features
- Stacked hourglass 3D CNN to extend regional support of context information into cost volume
- Achieves state-of-the-art accuracy on the KITTI dataset

3.2 Our Baseline

We referred to the PSMNet architecture to get an idea of what a good baseline would be. Using the open-source PSMNet GitHub repository as a reference, we re-implemented the PSMNet architecture with the basic submodule concept for cost aggregation and refinement. To verify the performance of our re-implementation we trained the author's implementation on the KITTI dataset and observed that the results were quite similar. We then conducted preliminary experiments in an attempt to improve upon the basic architecture described in (1), which are described in the following subsections. As a result of our findings, we settled on a baseline architecture that consists of ten $3\times3\times3$ convolutional

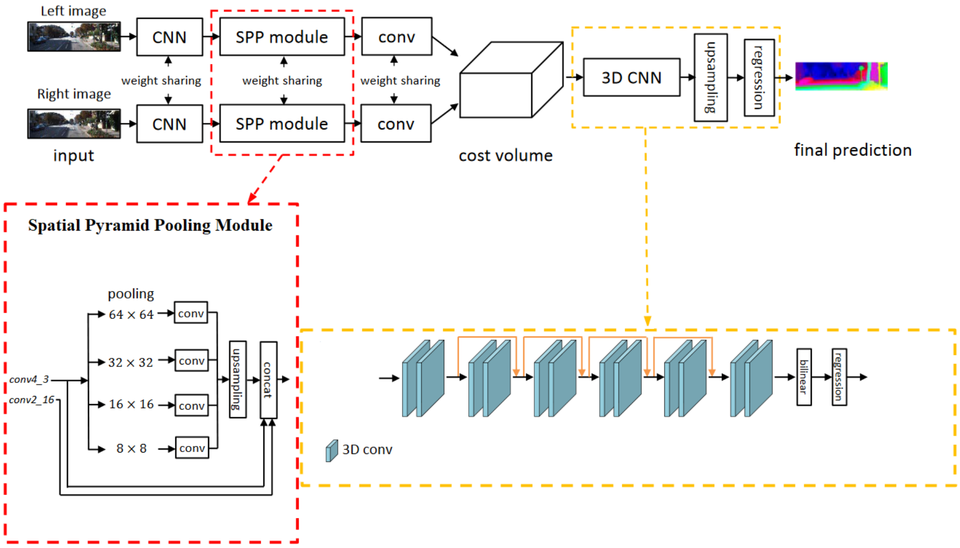


Figure 1: PSMNet Architecture (1)

layers for cost volume regularization and asymmetric convolutions for model parameter reduction. Note that while the model from (1) is trained on the Sceneflow dataset and fine-tuned on the KITTI dataset, we consider only the KITTI dataset to improve the rate of the experiments and evaluate the generalization capabilities of the model in the presence of limited representative data samples.

3.2.1 Reducing CNN layers

The basic architecture is built using residual blocks, containing twelve 3×3×3 convolutional layers. We conducted experiments in an attempt to both understand the architecture as well as improve upon it for our own baseline, and the results can be shown in Figures 2 and 3. We have tried both increasing and decreasing the number of convolutional layers by two, in order to observe the impact on L1 loss and 3-pixel accuracy. For the L1 loss, our experiments showed that all three variations achieved almost identical performance after roughly 20 epochs. However, we note that the variation with decreased convolutional layers actually converges at a much quicker rate, reaching an L1 loss of under 2 within 10 epochs. This is almost 50% faster when compared to the other two variations! Similar behavior is observed with the 3-pixel error: all three variations achieved nearly identical results after roughly 20 epochs, but the variation with decreased convolutional layers converges significantly quicker. We make note that the plots only show data for the first 30 epochs, as that was enough to portray the performance trends.

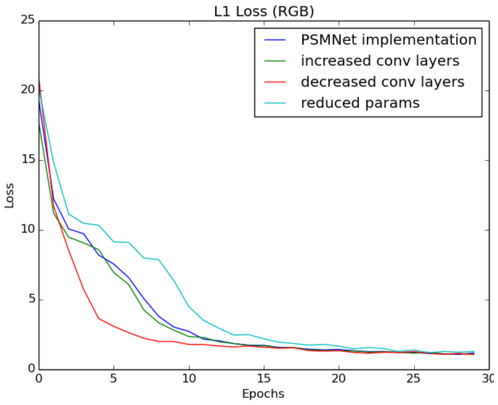


Figure 2: L1 Loss Findings

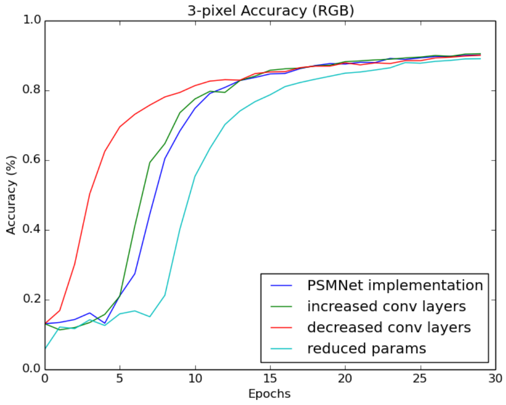


Figure 3: 3-pixel Error Findings

3.2.2 Parameter Reduction

When looking to improve the architecture of the PSMNet model from (1) one method that was considered was spacial reduction of the convolutions. This was heavily inspired by Szegedy et. al. (16). The models did not have any excessively large convolutions but did have a plethora of 3x3 convolutions. We could break them down into smaller 2x2 convolutions but an even better approach is to use asymmetric convolutions. The inception paper has shown that using a 3x1 convolution followed by a 1x3 convolution is equivalent to sliding a two layer network with the same receptive field as in a 3x3 convolution. This is shown in Figure 4. This two-layer method is 33% cheaper for the same number of output filters, if the number of input and output filters is equal (16). After implementing this methodology to our architecture it resulted in a reduction of parameters from 3,672,896 to 3,147,968 (a little over 14%). The modified network still reached the same test accuracy as the original but was a few epochs behind. An illustration of the original and modified architecture of a basic block in the model is presented in Figure 4.

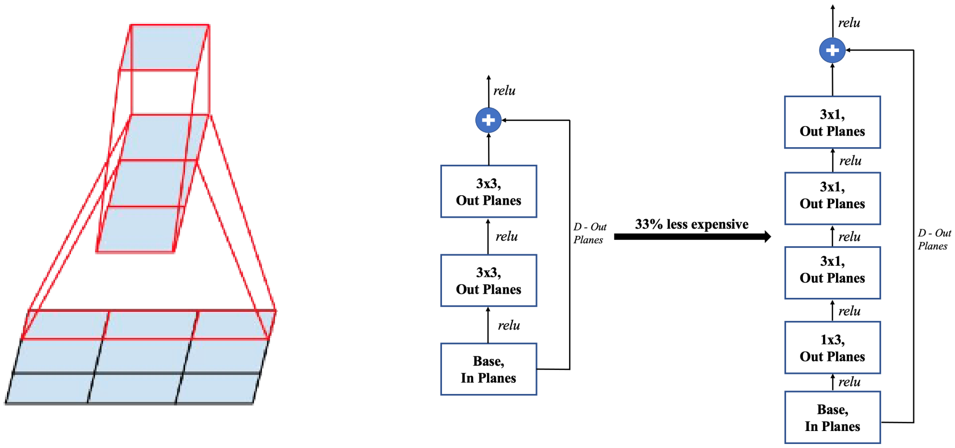


Figure 4: Left diagram depicts the mini-network replacing the 3x3 convolutions. Right diagram shows the original vs. modified architecture with asymmetric convolutions

3.3 Final Model

From the midterm report on-wards we focused our efforts on generating reasonable quality disparity maps from IR images, while ensuring a similar quality from the RGB dataset. Key areas of focus include experiments to reduce the number of feature extraction layers, applying more asymmetric convolutions to further reduce the total number of model parameters, and experimenting with a residual block-based SPP. Compared to our baseline model, we were able to achieve even better results on the IR dataset with fewer parameters. Methods and results will be discussed in the following sections.

3.3.1 Parameter Reduction Continued

Due to the positive impact the asymmetric convolutions in the basic block of the PSMNet architecture made on the midterm baseline model performance, further experiments included making all of the two dimensional convolutions asymmetric. This reduced the parameters from 3.67 million parameters (in the original PSMNet architecture) to 2.77 million parameters which was roughly a 25% reduction. This led to a reduction in the model complexity with a negligible change in the RGB image 3-pixel accuracy. This model is referred to as the "v1 reduced parameter" model. The three dimensional convolutional layers in the baseline model architecture contribute to a large majority of the computational complexity, as the cubic convolution computational complexity and high memory consumption make it quite expensive to deploy in real-world applications. This is the motivation behind the final parameter reduction alteration made to the model architecture. Three dimensional convolutions can be represented by three asymmetric convolutions in a similar manor to two dimensional convolutions. (17) as seen in Figure 5.

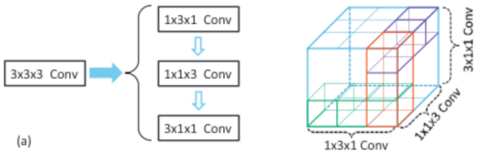


Figure 5: Approximation of a 3 x 3 x 3 3D convolutional layer by three asymmetric 3D convolutional layers (17)

3.3.2 SPP Module Modifications

We examined the performance of PSMNet on IR stereo images by first modifying the number of layers/branches in the feature extraction model to see if there was a significant impact on the model's performance. Our hypothesis was that the feature extraction module in PSMNet wouldn't work as well for generating disparity maps on the IR dataset as the RGB dataset since IR images generally have less texture and features than their RGB counterparts. Our baseline model originally used four layers of BasicBlocks in the feature extraction model, where a BasicBlock consists of a Conv2d layer followed by a BatchNorm2d and ReLU. We ran two modified models on the IR dataset: one consisting of only two layers of BasicBlocks, and the other consisting of five layers of BasicBlocks. The results can be seen in Figures 6 and 7. We observe that both variations achieve almost identical loss and error values in almost identical time periods, leading us to conclude that we're able to reduce the model's complexity without suffering from any reduction in performance.

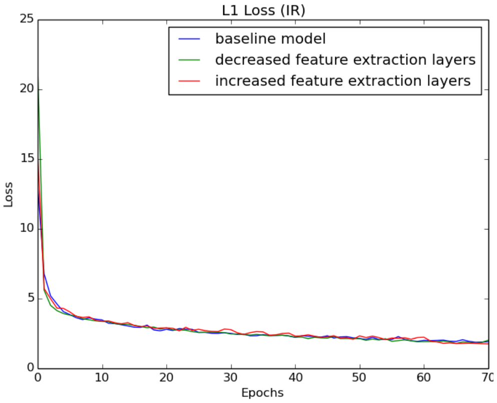


Figure 6: L1 Loss Findings

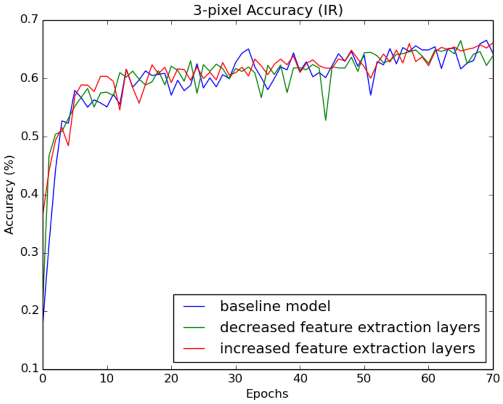


Figure 7: 3-pixel Error Findings

Using the insight gained from the aforementioned IR experiments, we redesigned the SPP module of PSMNet using residual blocks as shown in Figure 8 such that performance could be improved on IR images. The modifications described in this section, while tested primarily on IR images, may be applicable to RGB images as well. However, for the sake of this work we consider the architecture's performance on the more challenging problem of IR disparity estimation.

Similar to PSMNet, we first perform spatial pooling at scales 4×4, 8×8, 16×16, and 32×32. The outputs of each spatial pooling operation are sent to a convolutional block (CB) whose architecture is provided in Figure 9a. Specifically CB1 accepts 3 feature maps from the provided image and outputs 32 feature maps. The outputs from CB1 are passed to a series of 4 identity blocks. The design of each identity block (IB) is shown in Figure 9b. Note that the number of feature maps is unchanged by the identity block. The outputs of the identity block are passed through another set of convolutional (CB2) and identity (IB2) blocks. In the figure, CB2 accepts 32 feature maps and outputs 64 maps. The outputs from each spatial pooling branch are upsampled to a common size, concatenated, and passed through a final set of convolutional and identity modules. In Figure 8, CB3 takes in 512 feature maps and outputs 128 maps, while CB4 contains 64 filters. The final Conv layer contains 32 filters and performs a convolution with kernel size and stride both set to 1×1.

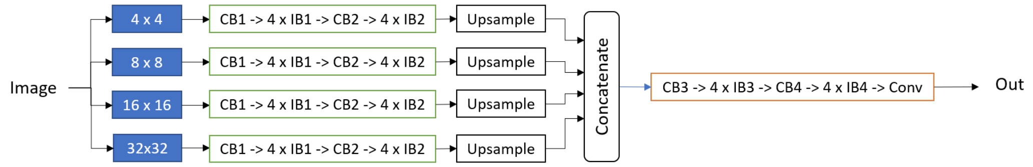
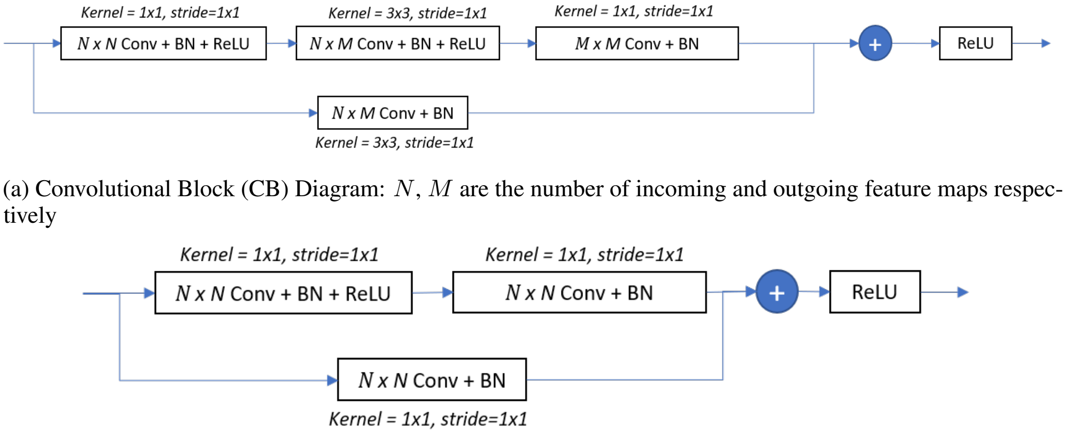


Image
4 x 4
8 x 8
16 x 16
32x32
CB1 -> 4 x IB1 -> CB2 -> 4 x IB2
CB1 -> 4 x IB1 -> CB2 -> 4 x IB2
CB1 -> 4 x IB1 -> CB2 -> 4 x IB2
CB1 -> 4 x IB1 -> CB2 -> 4 x IB2
Upsample
Upsample
Upsample
Upsample
Concatenate
CB3 -> 4 x IB3 -> CB4 -> 4 x IB4 -> Conv
Out

Figure 8: Modified SPP Module



(a) Convolutional Block (CB) Diagram: N, M are the number of incoming and outgoing feature maps respectively

(b) Identity Block (IB) Diagram: N is the number of incoming feature maps

Figure 9: Diagrams of convolutional blocks (CB) and identity blocks (IB) used in the modified SPP module

4 Experimental Setup

4.1 Datasets

We use the stereo2012 (10) and stereo2015 (18) datasets provided in the KITTI corpus. The KITTI datasets contain RGB stereo images for disparity map generation. Each stereo image provided has size 375×1242. The stereo evaluation benchmark from KITTI'15 provides 200 training and 200 testing scenes, while KITTI'12 gives 194 train and 195 test scenes in the set. The KITTI benchmark suite does not make the ground truth for the testing data public, hence we re-purpose the combined datasets formed from the training sets and evaluate the performance of our models on a split of the combined dataset. The combined set includes 393 image pairs along with their disparity maps, we use a 80-20 split for the train -evaluation partitions.

While the KITTI dataset is widely considered the standard benchmark for disparity estimation in RGB images, there does not exist a widely accepted benchmark dataset for disparity estimation in IR stereo images. While the CATS Benchmark(11) does contain thermal-thermal stereo pairs, the scenes captured along with the semantics of the dataset were not similar to the KTTI dataset. The number of outdoor scenes in CATS was just 79 which is very low even compared to the KITTI dataset. Hence, we use a custom urban scene dataset with close semantics to the KITTI for performing disparity estimation in stereo IR images. The dataset contains 323×429 stereo images captured by thermal cameras along with dense disparity maps constructed from AANet(14). The dataset was curated at Carnegie Mellon University and contains images of representative driving scenes in Pittsburgh, PA. There exist 244 train and 100 test IR stereo image pairs in the dataset. We employ an 80-20 train-validation split on the train data in consistency with the KITTI dataset.

4.2 Evaluation Metrics

We use the KITTI evaluation benchmark to measure the disparity error of our models. The evaluation metric is 3 pixel error, where a pixel is noted to be correctly estimated/predicted if the absolute difference between the predicted pixel disparity and the value from the corresponding index in the ground truth is < 3px. Furthermore there is an additional constraint of considering only the pixels where the counterpart in the ground-truth has a value > 0. This can also been as the D1-all error on the KITTI leaderboard for stereo 2015. We specifically use the 3 pixel error in this work since it is widely regarded as the standard evaluation metric for density estimation in the literature. Hence, since most reported works report their results in terms of the 3 pixel error, we use the same metric so that we can directly compare the performance of our model with others.

4.3 Training Methodology

For the RGB images, we employ the KITTI dataset for training the disparity map generation model. Specifically, the KITTI'15 and KITTI'12 datasets are combined and 80% of the samples are used for training. The remaining 20% of image pairs are reserved for validation. While the aforementioned models are trained on 256×512 random crops extracted from the stereo images for the RGB images, the IR images had a random crop of 256×416. The random crops are normalized by the ImageNet means. We use Adam optimizer with learning rate 0.0001 and beta values (0.9, 0.999). Due to the size of the PSMNet model, we use a relatively small batch size of 4 such that we could run experiments using a single GPU. We use a Tesla T4 for training and testing all our models.

5 Experimental Results

As discussed in Section 4, we use the 3 pixel disparity error to evaluate our models and compare them against the original PSMNet(1) performance. A comparison of each model's total number of parameters used, error on the RGB dataset, and error on the IR dataset can be seen in Table 2.

Table 2: Performance Comparison

Name	Params	RGB Error	IR Error
PSMNet-Our Implementation	3.6 mil	6.4%	25.9%
Baseline	3.1 mil	6.9%	31.2 %
v1 reduced param	2.77 mil	6.7%	33.3%
v2 reduced param	2.58 mil	9.7%	36.8%
Final model	1.7 mil	8.4%	23.7%

5.1 RGB Results

As described in the Experimental Setup section, we used the KITTI dataset to generate the disparity maps from RGB images. We compare our re-implementation of the PSMNet model with a few models of our own, and the results can be seen in Figures 10 and 11. We note that all of our models are able to achieve competitive results with the PSMNet model within 80 epochs while using significantly fewer parameters. One especially significant contribution of ours is reflected in the visible decrease in training time required. This is depicted in Figure 12, as we make note that the v2 reduced param model requires less than half the total training time compared to the PSMNet model. Two examples of generated disparity maps from RGB images can bee seen in Figure 14, one for an ideal environment without objects and the other with an environment cluttered with objects. Even though the following modifications were made in order to accommodate for better feature extraction in IR images when tested on RGB images we observed that it was able to perform fairly well. We hypothesize that the small loss in accuracy might because of the reduction in pooling sizes.

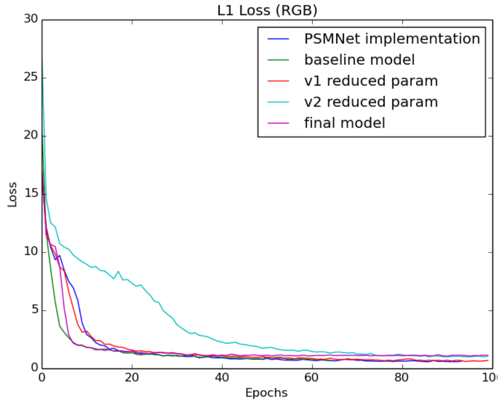


Figure 10: L1 Loss Comparisons for RGB

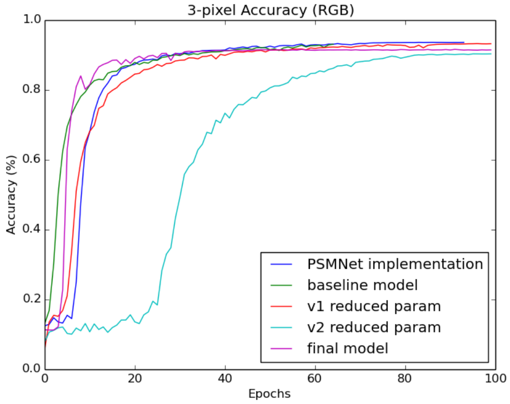


Figure 11: 3-pixel Error Comparisons for RGB

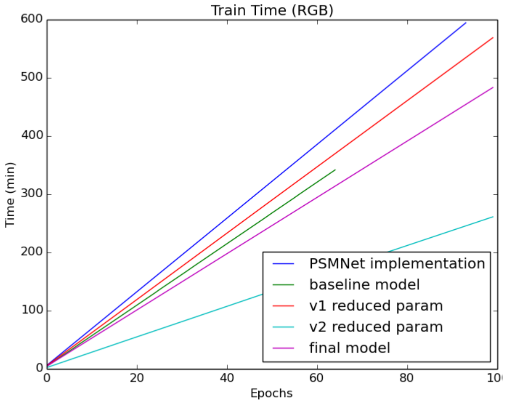


Figure 12: Train Time Comparisons for RGB

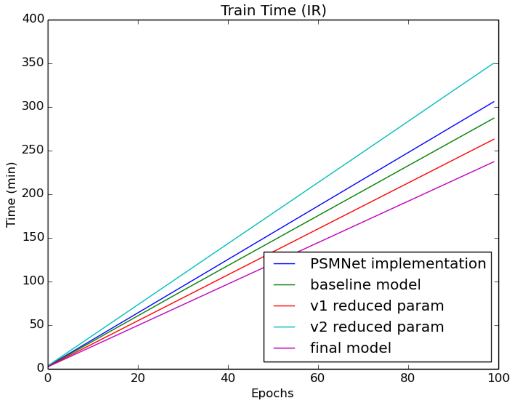


Figure 13: Train Time Comparisons for IR

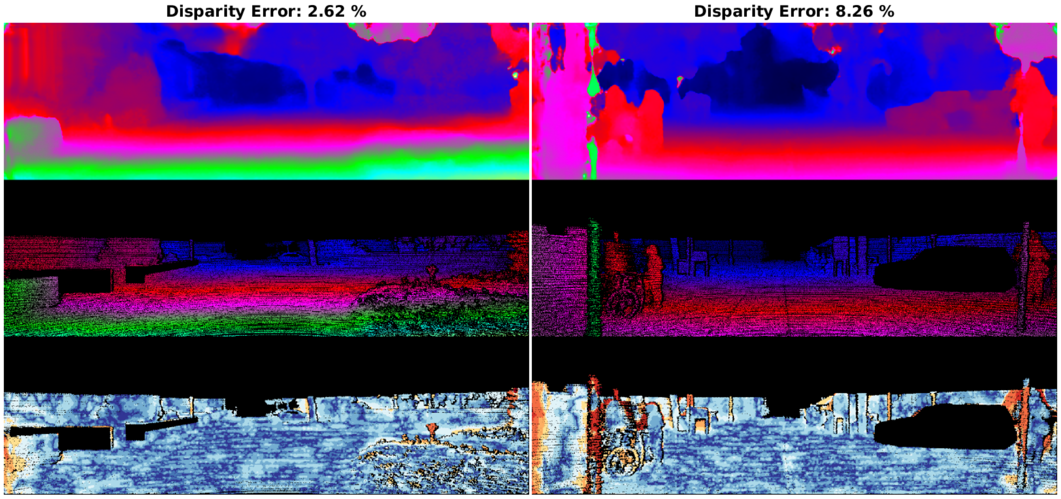


Figure 14: Disparity error visualization. Top row is the generated disparity map, middle row is the GT, and the last row is the error visualized on the GT. Left image depicts generated disparity map with low disparity error in absence of multiple objects, right image depicts a relatively high disparity error

5.2 IR Results

Similarly as for the RGB results, we compare our re-implementation of the PSMNet model with a few models of our own on the IR dataset. The results can be seen in Figures 16 and 17. We note that our final model actually performs better on the IR dataset than the PSMNet model, while also requiring significantly less time during training as shown in Figure 13. The disparity maps are visualized in Fig 15. The left image depicts a stationary scene with sparse objects, resulting in a performance quite comparable to that of the RGB images. The right image on the other hand, captures a scene with multiple moving objects, causing the resulting disparity map to have a high disparity error concentrated on the moving objects and along the edges.

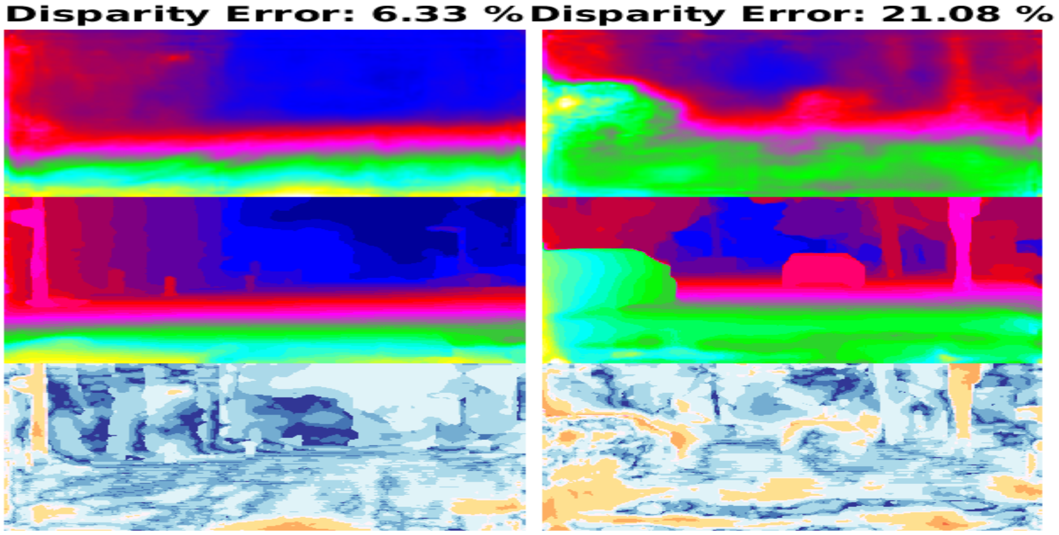


Figure 15: IR-Disparity error visualization. Top row is the generated disparity map, middle row is the GT, and the last row is the error visualized on the GT. Left image depicts generated disparity map with low disparity error in absence of multiple objects, right image depicts a relatively high disparity error

The loss comparisons for our models described in Section 5 are shown below. Clearly our final model, with the modified SPP module that contains residual blocks, outperforms the PSMNet implementation on IR images.

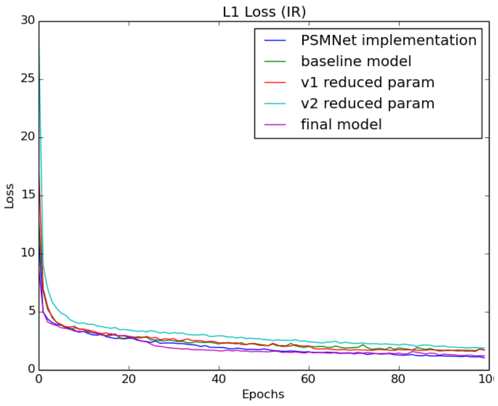


Figure 16: L1 Loss Comparisons for IR

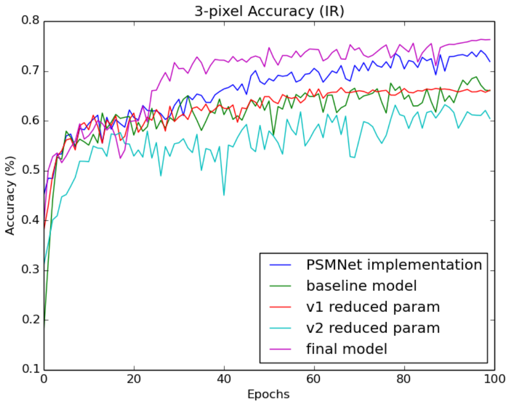


Figure 17: 3-pixel Error Comparisons for IR

5.3 GitHub

Our GitHub repository can be found here.

6 Division of Work

All duties (literature review, coding, debugging, and drafting of the report) were evenly divided amongst the group. All team members were instrumental in contributing to the development of models, debugging the code, and contributed evenly to the report. Hence we reiterate the fact that contributions were equal.

7 Conclusion & Future Work

The key contributions of our project include parameter reduction methods that reduced the total number of parameters from the PSMNet model by 53.65 %. We were able to achieve comparable results on RGB images but even better results on IR images. Our final model consumes at least 14% less memory and takes only 65% inference time when compared to the PSMNet model. By the midterm deadline, we had implemented a baseline model that is competitive with the basic architecture described in (1). Notable modifications made included a substantial reduction in the number of network parameters, as well as reduced number of CNN layers while still achieving comparable results. Since then, we further improved our baseline model by continually reducing parameters via 3D asymmetric convolutions, and achieved even better results on the IR dataset than the basic architecture from the PSMNet model. While the disparity maps generated from the IR dataset don't quite match the quality of disparity maps generated from RGB images, we believe our results provide a significant contribution to others in the scientific community who are also researching the potential of utilizing IR images to generate disparity maps. We propose as future work to combine asymmetric convolution methods with a residual block-based SPP module to further reduce model parameters while improving disparity generation from IR images. We also propose the design of more robust feature extraction and prepossessing methods such as guided image filtering to further improve disparity map estimation in IR images.

References

[1] J.-R. Chang and Y.-S. Chen, “Pyramid stereo matching network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5410–5418, 2018.

[2] J. Sun, L. Chen, Y. Xie, S. Zhang, Q. Jiang, X. Zhou, and H. Bao, “Disp r-cnn: Stereo 3d object detection via shape prior guided instance disparity estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10548–10557, 2020.

[3] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, “3d object proposals using stereo imagery for accurate object class detection,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1259–1272, 2017.

[4] A. D. Pon, J. Ku, C. Li, and S. L. Waslander, “Object-centric stereo matching for 3d object detection,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8383–8389, IEEE, 2020.

[5] S. J. Krotosky and M. M. Trivedi, “A comparison of color and infrared stereo approaches to pedestrian detection,” in *2007 IEEE Intelligent Vehicles Symposium*, pp. 81–86, IEEE, 2007.

[6] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[7] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, “Ga-net: Guided aggregation net for end-to-end stereo matching,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 185–194, 2019.

[8] N.Mayer, E.Ilg, P.Häusser, P.Fischer, D.Cremers, A.Dosovitskiy, and T.Brox, “A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation,” in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134.

[9] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, “End-to-end learning of geometry and context for deep stereo regression,” *CoRR*, vol. abs/1703.04309, 2017.

[10] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[11] W. Treible, P. Saponaro, S. Sorensen, A. Kolagunda, M. O’Neal, B. Phelan, K. Sherbondy, and C. Kambhamettu, “Cats: A color and thermal stereo benchmark,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 134–142, 2017.

[12] J. Zbontar and Y. LeCun, “Computing the stereo matching cost with a convolutional neural network,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[13] C. Zhu and Y. Chang, “Stereo matching for infrared images using guided filtering weighted by exponential moving average,” *IET Image Processing*, vol. 14, no. 5, pp. 830–837, 2020.

[14] H. Xu and J. Zhang, “Aanet: Adaptive aggregation network for efficient stereo matching,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[15] S. Birchfield and C. Tomasi, “A pixel dissimilarity measure that is insensitive to image sampling,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, pp. 401–406, 1998.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” *CoRR*, vol. abs/1512.00567, 2015.

[17] “Asymmetric 3d convolutional neural networks for action recognition,” *Pattern Recognition*, vol. 85, pp. 1 – 12, 2019.

[18] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.