Stereo Analysis using Low Texture Thermal Data

Aditi Damle
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
aditimid@andrew.cmu.edu

Karan Tank
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
ktank@andrew.cmu.edu

Siddhant Jagtap
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
sjagtap2@andrew.cmu.edu

Vaidehi Joshi
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
vaidehij@andrew.cmu.edu

Abstract

Infrared Images (IR) images are known to be robust to ambient illumination, fog and incoming car headlights. These features should make them favourable for autonomous driving applications, however, they are less frequently used due to their noisy and textureless nature. In addition to this, IR Cameras embed the thermal component emitted by the objects which give them better visibility in extreme weather conditions or in low light scenes, and hence a better performance as compared to RGB cameras, in such situations. In this report, we have described the methods employed to estimate disparity from low texture, noisy IR data. For the baseline architectures, we have selected AANet (Adaptive Aggregation Network) and GA-Net (Guided Aggregation Net), which are state of the art architectures for disparity estimation using RGB stereo images. These models employ fewer 3D convolutional layers, which enables operation at real time speeds. In experiments, we show how these models perform on IR stereo data, after extensive tuning of parameters. For training and evaluation, we use the CATS (A Color And Thermal Stereo) dataset and yet to be published, CMU dataset. Finally we discuss the shortcomings of these models and describe our proposed method which improves upon the baseline architectures. This field of estimating disparity using IR images, remains unexplored, and we aim to solve this problem. The link to our GitHub repository, `https://github.com/karan2808/STEALTH.git`.

1 Introduction

The technique of stereo matching, also known as stereo correspondence, has become an integral part of several applications which include robotics and autonomous vehicle navigation. Measuring the disparity(horizontal distance) between corresponding pixels in rectified stereo images is the key idea behind stereo correspondence. For example, if a particular pixel at position (x, y) in the left image is identical or similar to a pixel at position $(x - d, y)$ in the right image, then the disparity between these pixels is d. These estimated disparity values for every pixel pair can be used to compute the depth z using the equation $z = fB/d$ where $B(Baseline)$ is the distance between the left and right cameras and f is the focal length of each camera. Although the concept of disparity estimation seems intuitive, it is difficult to compute this quantity in practice.

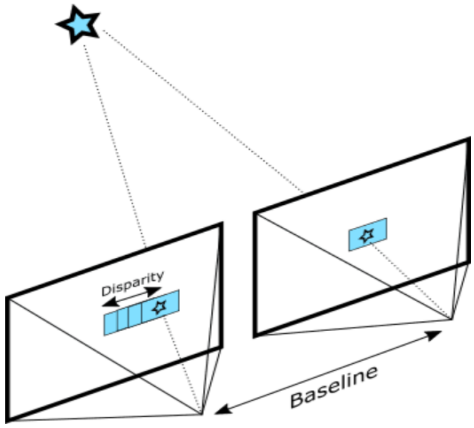


Fig 1: Diagram illustrating basic concept behind disparity estimation

Stereo matching often requires remote visibility with minimal distortion. RGB images are rich in texture and hence seem to be a better choice for stereo matching. But their quality suffers in low light and extreme weather conditions. Therefore, we explore the use of IR images for this purpose. IR images are fog tolerant, invariant to ambient illumination and are not blinded by incoming lights from other vehicles. Although they lack good texture and resolution, they can be improved by image pre-processing, for better compatibility with the stereo-matching algorithms.

While conventional state of the art methods have shown good performance, recent better performing algorithms, have been implementing Deep learning methods using mostly CNNs. However, these models add significant overheads on computation and memory requirements. There have been many attempts at sparsifying the computation to improve the speed. Moreover, to improve accuracy, some implementations have also used multiple camera setups and heavy data pre-processing which require more computational power.

We have addressed the problem of computing stereo disparity using fewer 3d convolutional layers and have used less pre-processing to improve the computational speed. We have also used a super resolution network as an extension to our disparity network to bring our disparity estimate to higher resolution.

2 Literature Review

We explored the following literature revolving around our problem statement:

Adaptive Aggregation Network for efficient stereo matching[11] proposes AANet which consists of a new sparse points based representation for intra-scale aggregation and adaptive multi-scale cross aggregation modules. This network uses a feature extractor similar to PSM-Net[1] to estimate the cost volume which enables intra-scale and inter-scale cost aggregation. This is one of the few networks that replace regular convolutional layer kernels with deformable kernels i.e kernels with varying size and shape. Furthermore, the AANet is known to have a low evaluation run-time of approximately 60ms which makes it suitable for use in autonomous driving applications.

Long range 3D with Quadocular Thermal camera[3] demonstrates a quadocular LWIR camera setup, calibration, image capturing and processing that results in long range 3D perception. This paper also aims at dealing with two major challenges of thermal images: lower image resolution and lower contrast of the textures using DNN approach for multimodal images. Images from all cameras are processed separately for sub-pixel matching using phase correlation. The depth map is generated using an iterative process. They use TPNET[2] architecture to train the model for disparity estimation. They use pseudo-ground truth for their error estimation and get good results.

Guided Aggregation Net for End-to-end Stereo Matching[12] is another successful model for disparity estimation of RGB images. It proposes two layers that can be used to replace the widely used 3D convolutional layer which is computationally costly and memory-consuming as it has cubic computational/memory complexity. These layers also help to account for low textured and occluded

regions in the images. This proposed method outperforms the state-of-the-art GC-Net on both Scene Flow dataset and KITTI dataset benchmarks.

Sub-pixel matching method for low-resolution thermal stereo images[14] proposes a stereo vision system using low-resolution 80×60 pixels thermal stereo images. It uses a three stage sub-pixel stereo matching framework. In the first stage the authors try to extract features from the two images, using phase congruency. This technique is found to be more stable and robust compared to conventional methods like SIFT and FAST, and is found to extract more number of feature points. In the second stage they perform a rough matching of these features, in order to get a rough estimate of the depth. They then refine the matched points using local phase coherence to obtain disparity estimate at a sub-pixel accuracy. Finally they use KNN (K Nearest Neighbour) to cluster the points together to obtain the disparity estimate. It has been demonstrated in the paper that this approach provides a more robust and reliable estimate of matching pixels compared to methods like ORB+KNN. We plan on using this frequency domain approach of phase congruency for the feature extraction stage in our model.

The Color And Thermal Dataset[10] dataset focuses on the stereo matching of various spectra, including those of RGB and thermal images. It consists of stereo thermal, stereo RGB, and cross-modality image pairs with high accuracy ground truth outcomes (<2mm) generated from a LiDAR system. We aim to use CATS as one of our datasets for this project.

Stereo Matching Using Gabor Convolutional Neural Network[7] proposes newly designed Gabor CNNs in the feature extraction part of the network to address the problem of geometric deformation caused by perspective projection by using Gabor filters. In this paper, the regular 2D convolution layers of the feature extractor are modulated by a Gabor filter bank. As a Gabor filter can be viewed as an edge detector along a particular orientation/angle, this strategy is successful in alleviating the problem of geometric deformation in the input stereo pair. The disparity cost volume is computed by correlating two Siamese networks which are produced by the feature extractor module. Finally, the cost aggregation is performed using a cascade of 3D convolutional layers followed by disparity regression by minimizing the smooth L1 loss function. This architecture is known to perform well on both RGB as well as grayscale images.

Attention Aggregation Encoder-Decoder Network Framework for Stereo Matching[13] proposes a attention aggregation encoder-decoder network framework for stereo matching, which the authors claim, can aggregate the cost volume to the utmost extent. The proposed model can be broken down into 3 modules. The first module consists of a sub-branch and cross-stage aggregation encoding module, which aggregates context information from different sub-branches and cross-stages to achive a mutual utilization of the cost volumes. The next stage consists of an attention recording module to obtain high to obtain a robust discriminative cost volume. This module uses a softmax function to get the similarity between the ith channel and the jth channel feature. The last stage consists of a step-wise aggregation decoding module to decode the cost function via the step-wise fusion up-sampling strategy. The dimensions of the high level features obtained are reduced in each branch. An upsampling operation is performed on low resolution features. The low resolution and high resolution features are hence brought to the same scale and they are fused together to obtain the final output.

Pyramid Stereo Matching Network proposes a PSM-Net[1] consisting of two main modules: spatial pyramid pooling and 3D CNN, to tackle the problem of current architectures which lack the means to exploit context information for finding correspondence in ill-posed regions. The spatial pyramid pooling module takes advantage of the capacity of global context information by aggregating context in different scales and locations to form a cost volume. The 3D CNN learns to regularize cost volume using stacked multiple hourglass networks in conjunction with intermediate supervision. This paper also focuses on something we want our method to achieve ie. to find better models than CNN for stereo matching; because the traditional CNN models do not give much context or semantic information in the images which is required for stereo matching. This paper proposes the PSM-Net which achieved good results on KITTI dataset and hence we propose to also take this network into consideration, in addition to the AANet and GA-Net networks, to extend it to the IR image datasets we are dealing with. PSMNets, explained in this paper, aim to exploit global context information for stereo matching, which we assume, could work for our thermal datasets.

Structure from Infrared Stereo Images[5] shows how to produce sparse disparity maps from uncalibrated infrared stereo images to produce a dense/semi-dense depth field, to deal with the problem

of discerning depth from stereopsis as the quality of un-cooled sensors is not sufficient for generating dense depth maps. The pipeline proposed in this paper starts by extracting a set of stable features from stereo pairs using the phase congruency model, then analysing those with a set of Log-Gabor wavelet coefficients, then refining the resulting sparse disparity map by triangular and epipolar geometrical constraints and finally a watershed transformation is applied to densify the disparity maps. Refining the ground truths, in our case, by this method can possibly give better results on the existing datasets.

End-to-End Learning of Geometry and Context for Deep Stereo Regression[6] proposes a novel approach to model disparity estimation as a regression problem. Disparity values are regressed from a multi-dimensional cost volume using the soft argmin operation. Unlike the traditional argmin operation, the soft argmin operation is differentiable and non-discrete making it suitable for backpropagation. As this method involves calculating the sum over weighted disparity values, it is also regarded as a soft attention mechanism.

Deep Learning for Image Super-resolution: A Survey [9] provides an in depth survey of advances in Super-resolution models and provides a detailed comparison between various architectural designs. The authors describe in detail, four main frameworks for supervised super-resolution, namely, pre-upsampling approach, post-upsampling approach, gradual-upsampling approach and iterative up-down sampling approach. We conclude from the paper that iterative up-down sampling method is the most well rounded method and models the low-resolution and high-resolution dependencies well.

3 Contribution

Our main goal is to predict disparity and subsequently the disparity error in the low resolution, noisy IR images. Out of all the models we implemented, we found that Pyramidal Stereo Matching network[1], with the necessary pre-processing steps and a modified architecture turns out to be better than the rest, for our problem statement. The original PSMNet was initially defined for RGB images to deal with the problem of other methods which lack the means to exploit context information for finding correspondence in ill-posed regions. Hence, as we are dealing with low textured thermal data, we decided to choose PSMNet to enable efficient feature extracion.

PSMNet consists of two main modules: Spatial pyramid pooling and 3D CNN. We used these two modules and modified some other modules from the original architecture along with certain pre-processing steps to fit our problem of computing stereo disparity for IR images. The entire model architecture and detailed description of the techniques used is briefed about in Section 3.1.

3.1 Netwok architecture

The detailed parameters and architecture of the modified PSMNet is given in the Table 1. Preprocessing of the left and right images is done as mentioned in Section 3.1. For the first convolutional layer, three small convolution filters (3×3) are cascaded to construct a deeper network with the same receptive field. The basic residual blocks are used for learning unary feature extraction. Following which receptive fields are enlarged using dilated convolutions. SPP Modules are then applied to gather context information. We concatenate the left and right feature maps into a cost volume, which is fed into a 3D CNN for regularization. Finally, regression is applied to calculate the output disparity map.

3.1.1 Pre-processing

For pre-processing of the left and the right images, we found out that Contrast Limited Adaptive Histogram Equalization works the best in retaining important features in the low texture, noisy IR images. A detailed description of this algorithm is given in Section 3.3.

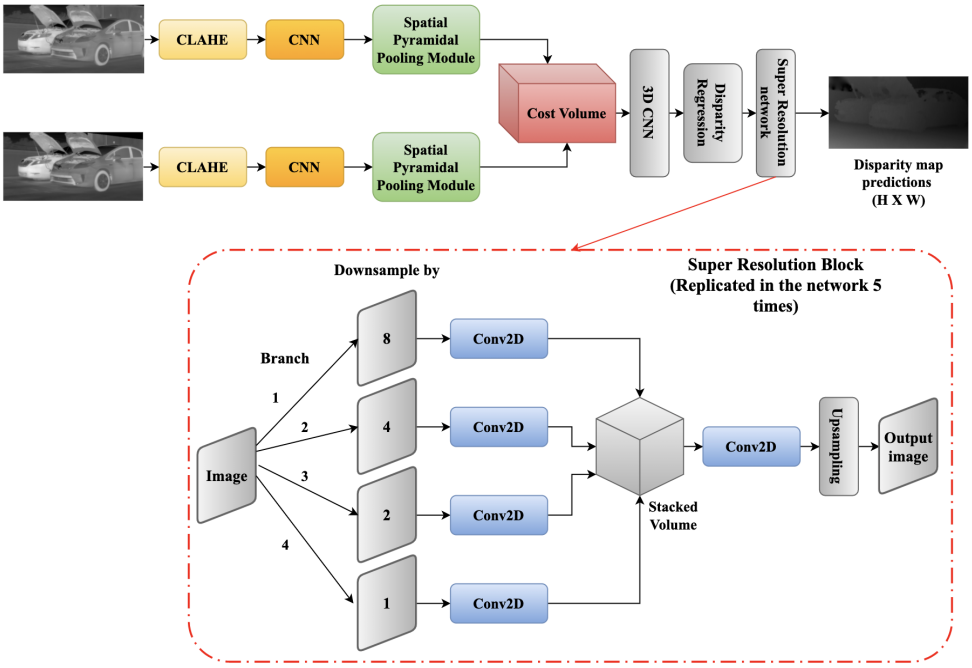


Fig 2: Network Architecture

3.1.2 Spatial Pyramidal Pooling (SPP) Module

Determination of context relationships solely from pixel intensities is difficult. Hence, image features rich with object context information can benefit from correspondence estimations, particularly for ill-posed regions. In the light of this, the Spatial Pyramidal Pooling (SPP) module as shown in Fig.2, incorporates hierarchical context information by learning relationships between an object and its sub-regions.

In this architecture, we use four fixed sized average pooling blocks for SPP which is followed by a 1x1 convolution to reduce feature dimensions, after which the low-dimensional feature maps are upsampled to the same size of the original feature map via trilinear interpolation. Finally, the different levels of feature maps are concatenated as the final SPP feature map.

3.1.3 Cost Volume

We adopt the GC-Net[6] approach by using SPP features to form a cost volume by concatenating the left feature maps with their corresponding right feature maps across a disparity level, resulting in a 4D volume (height x width x disparity x feature size).

3.1.4 3D CNN

The SPP module captures the necessary features and facilitates stereo matching by involving different levels of features. To aggregate the feature information along the disparity as well as spatial dimensions, we use a 3D CNN architecture for cost volume regularization. We use residual blocks to build this architecture. The output of the final convolutional layer is upsampled back to the original size of the image ie. H X W X D by using trilinear interpolation. Finally, we apply regression to calculate the final disparity map with size H x W. Once, this disparity map is generated, we use it with the ground truth to calculate the D1 error or the disparity error.

3.1.5 Disparity Regression

Disparity regression is used to estimate a continuous disparity map from the disparity map resulting from the CNN layer by calculating the probability of each pixel using a softmax operation $\sigma(\cdot)$ on the predicted cost. This is proven to be more robust than classification based stereo matching methods. A more detailed explanation can be found in Section 3.2.1.

3.1.6 Super Resolution Module

Image super-resolution refers to the process of recovering high-resolution images from low resolution images. It has become an important class of techniques for a variety of applications such as surveillance and medical imaging. This problem is ill-posed since there can be multiple high resolution images for a low resolution image.There are four main frameworks for supervised super-resolution, namely, Pre-Upsampling, Post-Upsampling, Progressive Upsampling and Iterative Up-Down-Sampling, as mentioned in Zhihao et al[9].

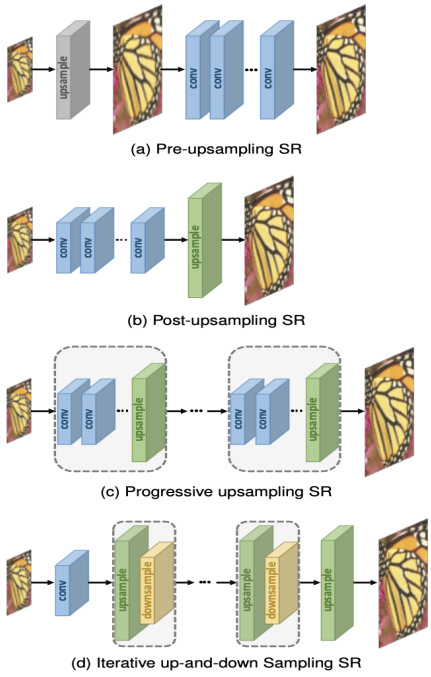


Fig 3: Diagram illustrating the four main frameworks for supervised super-resolution.[9]

Problem definition: The low resolution image can be modeled by the following equation,

$$I_x = (I_y \otimes k) \downarrow + \mathcal{N}(\mu, \sigma^2)$$

where the image I_x is the blurred down-sampled image, I_y is the sharp, high resolution image, k is the blur kernel and we have also added gaussian noise after blurring and down-sampling the image. Our goal is to obtain an estimate of I_y, say $\hat{I}_y$. The objective function can hence be modeled as,

$$\hat{\theta} = argmin_\theta L(\hat{I}_y, I_y) + \lambda\Phi(\theta)$$

Pre-Up-Sampling is a versatile method in which we first up-sample the low resolution image and then use a CNN architecture to obtain the high resolution image. Since the up-sampling operation is already done in the beginning, the model has to just refine the coarse image. It is versatile because it can work with a wide range of input sizes. However, the up-sampling operation in the beginning introduces side effects like noise amplification and blurring. It is also computationally inefficient as we need to work with full size images.

Post-Up-Sampling improves computational efficiency by up-sampling after most computation has been performed, in the low dimensional space. Since the feature extraction process and most computationally expensive operations are done in the low dimensional space, this method has become one of the most mainstream methods. However, since the up-sampling occurs only at one step, it greatly increases learning difficulty for large scale factors like 4, or 8. In those cases, we need to train different models for each scale factor and that becomes computationally expensive and time consuming.

Progressive Up-Sampling tries to improve on these drawbacks by progressively refining the images. In this framework a cascade of CNN blocks are used. The image is up-sampled, then refined using

CNNs and this process is carried out until the required resolution is obtained. This approach greatly reduces the learning difficulty and improves multi scale super-resolution without significant increase in spatial or temporal cost. However, these models are significantly difficult to design and face problems of training stability.

Iterative Up-Down Sampling Super-Resolution is a relatively new method which tries to capture the mutual dependencies of Low Resolution and High Resolution images in an efficient iterative procedure. As shown by Haris et al.[9], in SRFBN [9], and in RBPN[9], models that employ these methods can better mine the deep relationships between the low resolution and high resolution images providing higher quality results compared to the other frameworks. We have employed our own architecture based on this method for obtaining high quality images from our low resolution predicted images.

Stereo Disparity estimation requires smoothness constraints to better obtain the Disparity regions. The Smooth L1 Loss 3.2.2 is hence considered the default choice for this task. However, it does not model the sharp high resolution details that are required and tends to produce soft results. To deal with this problem, we have used a super resolution network ahead of our primary disparity computation network. To train the super-resolution module we have used the Mean Squared Error loss. The intuition behind decreasing the Mean Squared Error loss comes from the definition of PSNR or Peak Signal to Noise Ratio. The formula for PSNR is given by,

$$\text{PSNR} = 10Log_{10}(MAX^2/MSE)db$$

Where, MAX represents the maximum intensity value which is 255 in our case, and MSE stands for Mean Squared Error. Thus by decreasing the mean squared error, we effectively try to increase the PSNR and try to remove the effects of noise and up-sampling defects.

3.2 Mathematical Details

3.2.1 Soft Argmin Operation

As our project aims to estimate the disparity between a pair of rectified IR stereo images, we have modelled our deep learning pipeline as a regression task. The PSMNet architecture computes a 4D cost volume (height x width x disparity x feature size) which is subsequently used to estimate the disparity using cost aggregation. This cost volume is constructed by concatenating each feature from the SPP module with their corresponding feature from the opposite stereo image across each disparity level[6]. Our model includes the hyperparameter 'Maximum Disparity' which can be used to set the number of disparity levels for constructing the cost volume.

The cost aggregation operation is then performed by passing the cost volume through a series of 3D Convolutional layers. The output from these convolutional layers is then upsampled to generate the raw disparity predictions. Finally we make use of Disparity Regression to make our final predictions.

The Disparity Regression techniques uses a soft argmin operation which is differentiable and showcases the ability to regress a smooth estimate[6]. The predicted cost volume c_d is converted to a probability distribution by negating it $-c_d$. A softmax operation is then applied across the disparity dimension to generate logits for the different disparity levels. The final regression output is computed by taking the sum of each weighted disparity level d where the logits act as weights. This operation can be expressed mathematically as follows.

$$\text{soft argmin} = \Sigma_{d=0}^{D_{max}} d \times \sigma(-c_d)$$

3.2.2 Smooth L1 Loss

We have used the Smooth L1 Loss for computing the divergence for backpropagation. Although traditional L1 and L2 loss are widely used for regression tasks, the Smooth L1 Loss has proved to be advantageous in predicting the disparity map from stereo images. Fig 2 illustrates the mathematical details of the same.

$$smooth_{L1} := (x) \rightarrow piecewise\left(\text{abs}(x) < 1, 0.5 \cdot x^2, \text{abs}(x) - 0.5\right)$$

$$x \rightarrow piecewise\left(|x| < 1, 0.5\, x^2, |x| - 0.5\right) \quad \textbf{(1)}$$

Where x will be the L1 distance between 2 vectors.

$$smooth_{L1_{plot}} := piecewise\left(\text{abs}(x) < 1, 0.5 \cdot x^2, \text{abs}(x) - 0.5\right)$$

$$\begin{cases} 0.5\, x^2 & |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \quad \textbf{(2)}$$

→

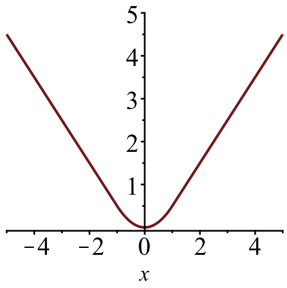


Fig 4: Diagram illustrating the Smooth L1 Loss along with mathematical equations (source: https://mohitjainweb.files.wordpress.com)

It is evident from the above figure that the Smooth L1 Loss is a combination of both L1 and L2 loss functions. The L1 component of the loss function ensures limited sensitivity to outliers. On the other hand, the L2 component of the loss makes the function strongly convex in a uniform neighborhood($|x| < 1$ in this case) around its minimum and therefore differentiable at the minimum. To summarize, the Smooth L1 Loss, also known as Huber Loss, is able to capture the sensitivity of the L2 loss as well as the robustness of the L1 loss. These characteristics make it suitable for use in disparity regression.

3.3 Algorithms Used

3.3.1 Contrast Limited Adaptive Histogram Equalization

For any grayscale image $\{x\}$ and n_i is the number of occurrences of gray level *i*, the probability of an occurrence of pixel of *i* in the image is given by

$$p(x_i) = p(x = i) = \frac{n_i}{n}, \quad 0 \leq i \leq L$$

where L is the total number of gray levels in the image (256), n is the total number of pixels in the image, and $p(x_i)$ is in fact the image's histogram for pixel value *i*, normalized to [0,1].

The c.d.f corresponding to $p(x)$, which is also in fact the image's accumulated normalized histogram, can be defined as

$$cdf_x(i) = \Sigma_{j=0}^{i} p_x(x = j)$$

The following properties of c.d.f. allows certain transformations

$$cdf_y(y') = cdf_y(T(k)) = cdf_x(k), \text{where k is in the range [0, L]}$$

Using them, we create a transformation $y = T(x)$ to produce a new image $\{y\}$ with a flat histogram, it would have a linearized c.d.f. across the value range i.e.

$$cdf_y(i) = iK, \quad \text{K being a constant}$$

Here, T maps the levels into range [0,1], since we used normalized histogram of $\{x\}$. To map the values back to their original range, a simple transformation is applied on the result.

$$y' = y \cdot (max\{x\} - min\{x\}) + min\{x\}$$

Ordinary histogram equalization uses the same transformation derived from the image histogram to transform all pixels. Using this method, the contrast in regions where there is an abrupt change in intensities does not get enhanced sufficiently. For infrared images, these abrupt changes are abundantly present. For dealing with such images, an extension of histogram equalization called Adaptive Histogram Equalization (AHE) is used.

Adaptive histogram equalization (AHE) improves on this by transforming each pixel with a transformation function derived from a neighborhood region. The transformation function is derived in a similar manner as original histogram equalization, however, it is proportional to the cumulative distribution function (CDF) of pixel values in the neighbourhood.

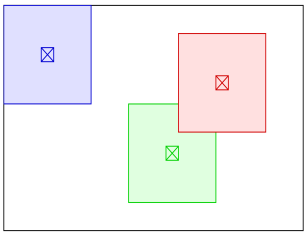


Fig 5: Diagram illustrating the regions of interest for adaptive histogram equalization. The tiles represent the regions used for computing histograms.d (source: Wikipedia).

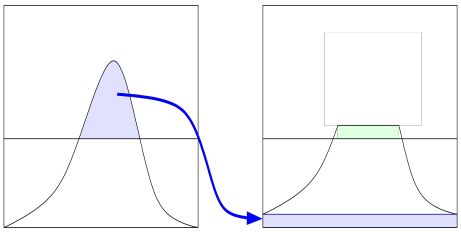


Fig 6: Figure illustrating the clipping of histograms.(source: Wikipedia).

Ordinary Adaptive Histogram equalization however has a drawback. It tends to over amplify the contrast in near-constant regions of the image, since the histogram in such regions is highly concentrated. Hence, it may cause the noise to be amplified as well. To deal with this issue, we have use contrast limited adaptive histogram equalization (CLAHE).

In CLAHE, the contrast amplification in the vicinity of a given pixel value is given by the slope of the transformation function, and this slope is proportional to the slope of the neighborhood cumulative distribution function (CDF). CLAHE limits the amplification by clipping the histogram at a predefined value before computing the CDF. This limits the slope of the CDF and therefore, of the transformation function. After experimenting with different values of the tile grid size and clip limit, we found the optimal parameters for our use case to be (8, 8) and 2.0 respectively.

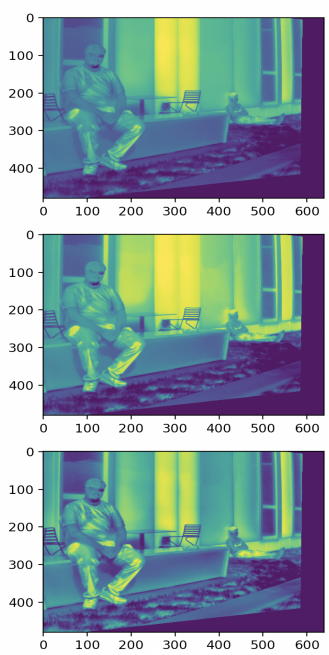


Fig 7: Figure illustrates the effect of histogram equalization and contrast limited histogram equalization on the infrared image. The top most image corresponds to the original image, the center image corresponds to the image obtained after histogram equalization and the bottom most image corresponds to the one obtained after contrast limited adaptive histogram equalization.

3.4 Implementation details

We implemented our model as described in Section 3.1 and Table 1. However, we made certain modifications to it which helped us decrease the disparity error on the CATS IR image dataset. In addition to the pre-processing steps mentioned in Section 3.1, we included more ReLU activations between the residual connections in order to capture the non linearities present in the stereo images. We reduced the average pooling dimensions in SPP module to capture fine features from the low resolution IR images. We divided the maximum disparity by 8 for the disparity dimension in the cost volume. Trilinear interpolation was used for upsampling the before the disparity regression block The CATS dataset was split in the following way. We used 150 images for training , 20 images for validation and 10 images for testing. We used the Adam optimizer with initial learning rate of 0.001 to train the model. The learning rate was scheduled using a StepLR scheduler($\gamma = 0.1$ and step size of 300). The model was trained for 500 epochs on an Nvidia Tesla P-100 GPU.

To train the super-resolution network, we down-sampled the ground truth disparity images by a factor of 20 and used them as inputs to the network. This was done for the first 250 epochs. After which, we used the disparity predictions from the disparity network to train the super-resolution network further. We used this approach because before 250 epochs the predictions of the Disparity network looked distorted and too noisy. After 250 epochs, the disparity predictions slowly started looking like down-sampled versions of ground truth disparity. We used the smooth L1 loss function for the disparity network to obtain a smooth, less noisy version of the disparity map and for the super-resolution network we used Mean Squared error loss to obtain a sharp image.

Our work can be easily re-implemented by referring to Table 1. Details regarding the super-resolution block can be obtained from [9]. Interested readers can also try to train our model on other popular IR stereo datasets. Furthermore, we have also provided a link to our GitHub repository for the reader's reference.

Table 1: Implementation details

Name	Layer details	Output dimensions
Input		H X W X 1
CNN		
conv2d-1	3 x 3, 32	0.5H x 0.5W x 32
conv2d-2	3 x 3, 32	0.5H x 0.5W x 32
conv2d-3	3 x 3, 32	0.5H x 0.5W x 32
conv2d-x1	[3 x 3, 32] x 3	0.5H x 0.5W x 32
conv2d-x2	[3 x 3, 64] x 16	0.25H x 0.25W x 64
conv2d-x3	[3 x 3, 128] x 3	0.25H x 0.25W x 128
conv2d-x4	[3 x 3, 128] x 3	0.25H x 0.25W x 128
SPP Module		
branch1	16 x 16 average pool; [1 x 1, 32]; Bilinear interpolation	0.25H x 0.25W x 32
branch2	8 x 8 average pool; [1 x 1, 32]; Bilinear interpolation	0.25H x 0.25W x 32
branch3	4 x 4 average pool; [1 x 1, 32]; Bilinear interpolation	0.25H x 0.25W x 32
branch4	2 x 2 average pool; [1 x 1, 32]; Bilinear interpolation	0.25H x 0.25W x 32
concat[conv2d-x2, conv2d-x4, branch-1, branch-2, branch-3, branch-4]		
fusion	[3 x 3, 128], [1 x 1, 32]	0.25H x 0.25W x 32
Cost Volume		
Concat left and shifted right		0.25D x 0.25H x 0.25W x 64
3D CNN		
conv3d-1	[3 x 3 x 3, 32] x 2	0.25D x 0.25H x 0.25W x 32
conv3d-2	[3 x 3 x 3, 32] x 2	0.25D x 0.25H x 0.25W x 32
conv3d-3	[3 x 3 x 3, 32] x 2	0.25D x 0.25H x 0.25W x 32
conv3d-4	[3 x 3 x 3, 32] x 2	0.25D x 0.25H x 0.25W x 32
conv3d-5	[3 x 3 x 3, 32] x 2	0.25D x 0.25H x 0.25W x 32
conv3d-last	[3 x 3 x 3, 32]; [3 x 3 x 3, 1]	0.25D x 0.25H x 0.25W x 1
Disparity regression		
Upsampling	Trilinear interpolation and disparity regression	H x W

4 Experimental Evaluation

4.1 Experiments

As described in the previous section, we developed our model using the basic building blocks of the PSMNet architecture. Furthermore, we used certain data pre-processing techniques, filtering techniques, color coding techniques and made some modifications to the model architecture, which we describe here. In order to maintain consistency across all our experiments, we added an offset of 70 to the raw ground truth images and clipped the disparity maps to constrain the disparity values to $[0, 192]$(maximum disparity value for our model). We trained all our models, with 150 images for training, 20 for validation and 10 for testing, for the CATS dataset. For the CMU dataset, we used 7 images for training, 3 for validation and 2 for testing. We conducted all our experiments by training the model for 500 epochs. We found that Adam optimizer worked the best, after experimenting with several optimizers like SGD, RMSProp. The learning rate was changed from 0.001 to 0.0001 at the end of 300 epochs using the StepLR scheduler.

To start with our experimentation, we considered training the model with raw input left and right images i.e without any pre-processing. Due to the inherent low-contrast, low-texture and noisy nature of IR images, the predictions obtained were not up to to the mark and resulted in high loss values. We believe that the CNN module of the PSMNet architecture was unable to capture important features from the input images due to noise and lack of texture.

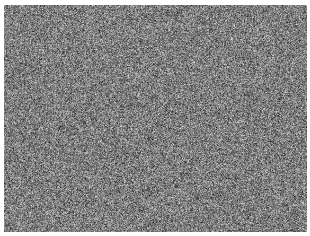
In our second attempt, we tried to color code the grayscale IR images using Matplotlib's Inferno colormap. By doing so, we wanted to explore if the PSMNet architecture(originally meant for RGB images) performs well on RGB versions of the IR images. However, since the color coded images also lack texture and contrast and display a limited range of colors, the model failed to capture the underlying trends in the disparity. Fig 2. shows an example of a color coded image using Matplotlib's Inferno colormap.

(a) (b)

Fig 8(a): The original gray scale image; Fig 8(b): The image encoded in inferno color scheme.

We also tried using frequency domain transformations like the Discrete Cosine Transform(DCT) followed by an MLP to extract the frequency components from the images. The reason behind using these was the basic property of frequency transformation i.e convolutions in spatial domain become multiplications in frequency domain. We computed the DCT of the left and right images and also of the ground truth and applied equal cropping to all these images. The element-wise product of the DCTs of left and right images was passed through a 3-layer fully connected network and smooth L1 loss function was used to compute the loss. This method failed to give good results. This could be due to the assymetric nature of DCT which prevents us from using filtering operations that we use in the fourier domain.

We then focused on spatial domain techniques, like Histogram of Oriented Gradients(HOG) and Laplacian of Gaussian filters(LOG), for pre-processing the input data to the PSMNet. These methods clearly did not work as can be seen in Fig 9 and Fig 10 (b). A huge amount of information was lost as these techniques mainly serve as edge detectors and do not account for the information present within edge boundaries. After trying several approaches mentioned so far, we understood that enhancing the contrast of the noisy IR images might help us achieve better results. For this task, we decided to apply Contrast Adaptive Histogram Equalization 3.3.1 to the input images. As this technique increases the local contrast of an image, the CNN layers were able to extract features efficiently, giving us much better results.

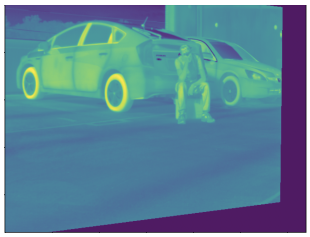


(a)

(b)

Fig 9(a): DCT of the gray scale image; Fig 9(b): HOG of the gray scale image.

Last but not the least, we would also like to discuss a few changes we made to the model architecture. In order to account for low resolution IR images(640 x 480), we used smaller kernel sizes in the branches of the SPP module(average pooling layers) as opposed to the original PSMNet architecture[1]. Furthermore, we also added a super-resolution module after the disparity prediction to avoid loss of information due to upsampling/trilinear interpolation. We also tried to make a few modifications to the model in hopes of adding regularization. One approach was to include dropout after each of the 2D convolutional layers in the feature extraction module. However, the training as well as the validation loss did not converge to a low value in the presence of dropout. On the other hand, we replaced the Adam optimizer with AdamW which is known to provide efficient weight decay. The results remained similar irrespective of the optimizer used.



(a)

(b)

(c)

Fig 10(a): The original image.; Fig 10(b): The image filtered using Laplacian Of Gaussian Filter (Sigma = 5).; Fig 10(c): The image processed with CLAHE.

We had previously used the AANet and GA-Net deep network architectures by modifying the max disparity to be compatible with the low resolution(640 x 480) IR images. One of our experiments included halving the max disparity to account for images having comparatively smaller dimensions. We used the Adam optimizer with a learning rate of 0.0001 and trained the baseline models for 1000 epochs in both cases. Our experimentation makes it evident that the PSMNet has a much better performance on IR images than both baseline architectures. 5 illustrates these results.

For conducting all our experiments, we developed the codebase from scratch by taking inspiration from the PSMNet repository on GitHub. We have also cited open-source code snippets that we have used in our codebase.

4.2 Dataset

We have used 2 datasets for the project; the CATS Dataset and the CMU Dataset.

The Color And Thermal Stereo (CATS) benchmark, is a dataset consisting of stereo thermal, stereo color, and cross-modality image pairs with high accuracy ground truth (< 2mm) generated from a

LiDAR. It has over 100 indoor and 80 outdoor scenes in various lighting conditions. The thermal images which we are currently using have been captured using Xenics Gobi-640-GigEs long wave infrared cameras at 640 x 480 resolution at 50 mK thermal sensitivity, and the ground truth has been captured using a Trimble GX Ad- vanced TLS LiDAR.

The CMU Dataset (Not yet published) consists of a million time synchronized Lidar, RGB Stereo and Thermal Stereo frames. The training data consists of a few selected thermal images captured across various lighting conditions. We have generated the ground truth stereo disparity using the RGB stereo images and by reprojecting the lidar frames onto the thermal images. The thermal images have been captured using FLIR ADK cameras at 640 x 480 resolution, the RGB stereo images have been captured using FLIR Blackfly cameras at 640 x 480 resolution and the Lidar point clouds have been captured using a Velodyne 16 beam Lidar.

4.3 Evaluation Metrics

We have chosen to use the Kitti D1-error as an evaluation metric for evaluating the quality of our model.

The D1-error represents the percentage of outliers in the predicted disparity map. Disparity can be computed by determining the shift to the left image of an image feature or a pixel, when seen in the right image. This shift can be measured using the difference between the location of the feature in the left image and it's location in the right image, along horizontal scan lines, in terms of the number of pixels between the two. Our goal while computing the stereo disparity is to compute this shift or the number of pixels between the matching pixels or features in the pair of stereo images. The evaluation metric (D1-error) puts a bound on this shift. We classify a predicted disparity value as an inlier if it's value falls within the range of 3 pixels from the ground truth disparity. The rest are classified as outliers, and D1-error provides a percentage of these outliers, taking into account all the valid disparity values. To compute the D1-error, we first compute the absolute value of the difference between the predicted and the ground truth disparity.

$$E = |D_{gt} - D_{pred}|$$

We then compute the number of outliers by imposing the following conditions:

- The ground truth disparity has to be positive.
- The absolute difference E has to be greater than a predefined threshold T1.
- The normalized value of E($\frac{E_{i,j}}{|D_{gt\,i,j}|}$) has to be greater than a predefined threshold T2.

All the points in the image satisfying the above conditions are counted as outliers $N_{outliers}$ and the number of points with ground truth disparity value greater than 0 are counted as valid disparity points N_{total}. The D1 error is hence given by,

$$D1 = \left(\frac{N_{outliers}}{N_{total}}\right) * 100$$

The values for T1 and T2 are set to 3 and 0.05 respectively, to ensure the evaluation is as strict as possible.

4.4 Description of baseline

We have used the following deep network architectures as baseline models for our experiments:

4.4.1 Adaptive Aggregation Network (AANet)

The Adaptive Aggregation Network(AANet)[11] is a recent deep network architecture which makes use of stereo correspondence for disparity estimation. Unlike most networks, the AANet utilizes the technique of deformable convolution for cost aggregation as opposed to conventional 3D convolutional layers. The deformable convolutional layers have the ability to reshape and resize the convolution kernels/filters as per the texture content at different locations in the input images. Furthermore, the AANet incorporates a 3-level pyramidal feature extractor similar to PSMNet[1].

Since we aim to compute the disparity using IR stereo pairs, the AANet is a reasonable baseline model. As discussed in the previous sections, IR images are known to have limited texture and irregular contrast. The adaptive sampling ability of the AANet can be leveraged to account for low texture regions in IR stereo images. Moreover, the AANet requires comparatively lesser time for both training and inference as compared to other disparity estimation DNNs. We have implemented the baseline model by modifying certain files present in the AANet public repository available on GitHub[11]. The AANet model was trained from scratch using the CATS dataset[10] and several hyper-parameters such as transformations on images, the max disparity value and data loader arguments were changed to facilitate training using IR images.

We made the following assumption while training this baseline model on IR images. As the AANet was originally designed to estimate the disparity using RGB images, it requires the input pairs to have three channels as well. However, IR images, being grayscale, are essentially single channel images. Therefore, we assume that IR images can be represented as 3 channel images by replicating the grayscale channel across 3 channels.

Finally, there is a major caveat to the original AANet architecture. The network was designed to operate on images from the KITTI 15[8] dataset which are almost twice the size of 640 x 480 IR images. Thus, the AANet architecture has a maximum disparity value of 192 pixels in the horizontal direction. As the IR images are much smaller in size, we lowered the maximum disparity value of the model to accommodate lower disparity values.

4.4.2 Guided Aggregation Network (GA-Net)

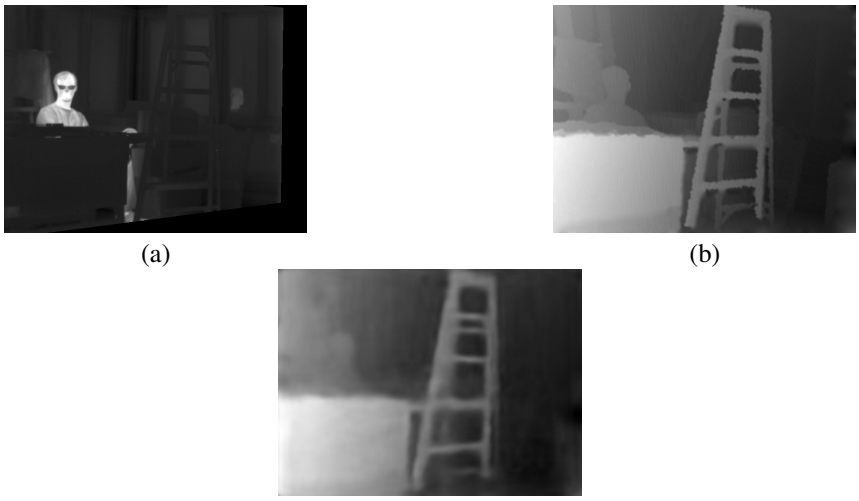
The Guided Aggregation Net(GANet)[12] is another top performing deep neural network based model for the KITTI 2015 dataset. We assume the model will generalize to IR images as well as it performs for the RGB images. The semi-global aggregation layers in it recover the data from low textured regions as well. We think this will serve to our advantage.

There are many versions of the GANet model that are included in the public GitHub repository[12]. We used the GA-Net11 architecture as the GANet-deep model is too large to load on a single GPU machine. We trained the model for 1000 epochs that took us about 2 days with one GPU. We used a batch size of 2 and to train and test the dataset on this model. The maximum disparity was set to 96 as it gives best results. As the dataset was low-resolution compared to the KITTI dataset, we set the crop size of 640 * 528. This is a caveat of the GANet model that the crop sizes and the maximum disparities have to be multiple of 48, since the model is tuned to perform well on the high resolution Kitti Dataset. Similar to AANet, we also needed to replicate the IR images across 3 channels to account for the RGB analysis of the existing model.

Unfortunately, we weren't been able to achieve decent results on both models due to some inherent characteristics of IR images mentioned earlier. Moreover, both the models are huge and are designed to be trained on distributed frameworks. This prevented us from training them using a batch size greater than 2. We believe this was a major drawback and prevented us from training the models to their potential. Following which we decided to shift to PSMNet which served to be a feature extractor in the AANet as well. We tested our data by trianing with the PSMNet baseline architecture as well. It performed a bit better than AANet and GANet with the preprocessed images. However it was not enough. Hence we made the necessary modifications to the PSMNet blocks as per our requirement as explained in section 3.

5 Results

Results from our Stereo Disparity Network.



(a) (b)

(c)

Fig 11(a): Left Camera Image.; Fig 11(b): Ground truth Disparity.; Fig 11(c): Predicted Disparity.

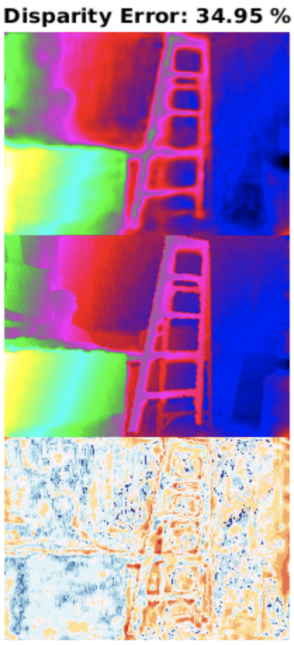
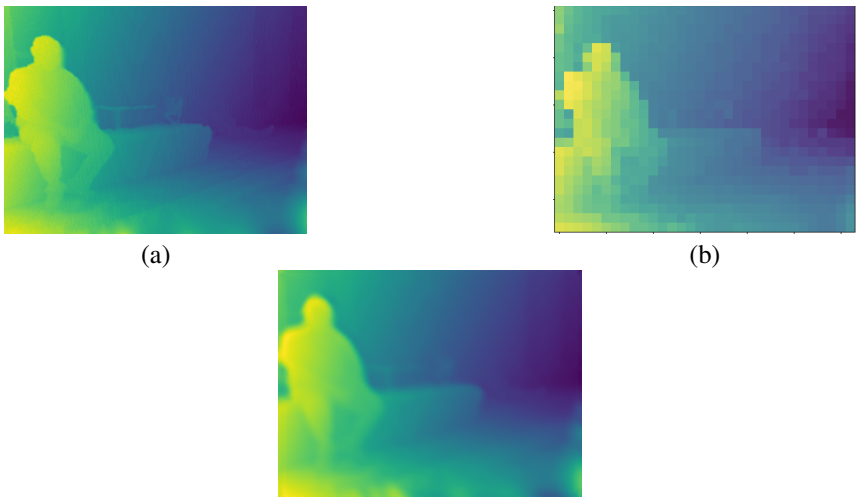


Fig 12: The D1 error calculation results. The top most image corresponds to the predicted disparity. The center image corresponds to the ground truth disparity. The bottom most image corresponds to the error visualization. Here, the cool colors indicate regions of low error and warm colors indicate regions of high error. The threshold for error evaluation is set to 3 pixels to keep the evaluation as strict as possible.

Results from the iterative Up-Down Sampling Super Resolution Network.



(a) (b)

(c)

Fig 13(a): Ground truth image.; Fig 13(b): Ground truth down-sampled to H/20, W/20.; Fig 13(c): Image reconstructed using super resolution network.

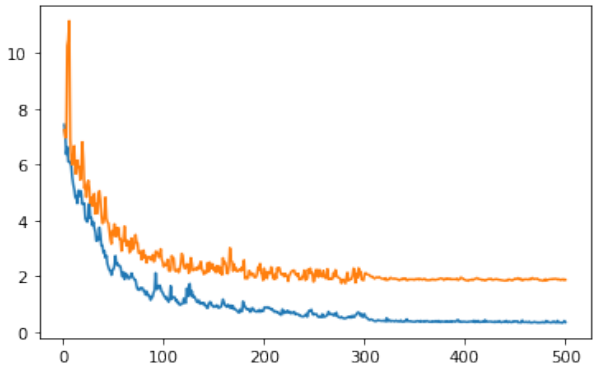
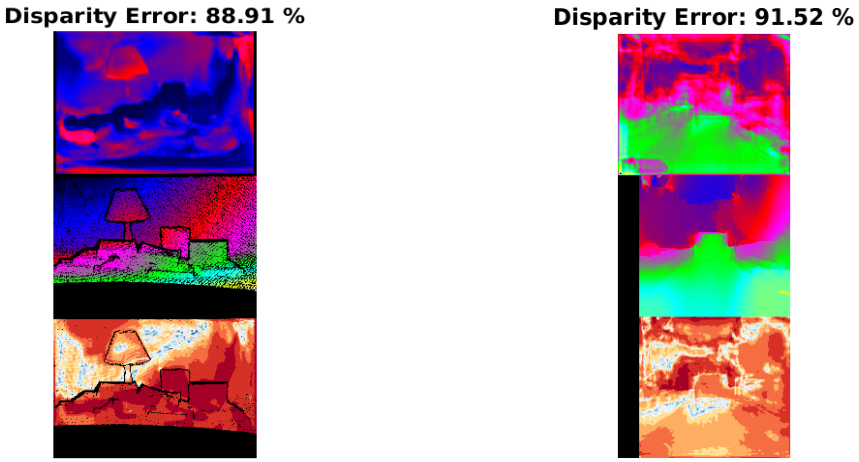


Fig 14: The training and validation loss plots for CATS dataset.
(Orange plot: Validation loss; Blue plot: Training loss)



(a) (b)

Fig 15(a): D1 Error Obtained Using Baseline Implementation, GANET, on CATS dataset.; Figure 15(b): D1 Error Obtained Using Baseline Implementation, AANET, on CMU dataset.

Table 2: Predicted disparity/ D1 error for different Models and/or Algorithms and/or preprocessing steps on CATS and CMU Datasets

Model/Algorithm	Dataset	Max Disparity	D1 error
AANet (Baseline)	CATS	192	90.22
GANet (Baseline)	CATS	192	96.6
PSMNet	CATS	192	77.35
Modified PSMNet	CATS	192	65.3
CLAHE + Modified PSMNet	CATS	192	58.5
Inferno + Modified PSMNet	CATS	192	74.5
CLAHE + Modified PSMNet + Super resolution	CATS	192	34.9
AANet (Baseline)	CMU	192	92.68
GANet (Baseline)	CMU	192	93.76
PSMNet	CMU	192	93.10

6 Conclusion and future work

We evaluated three state of the art models (AANet, GANet and PSMNet), on the CATS and CMU thermal dataset. These models are top performers on stereo disparity estimation using KITTI stereo dataset (RGB image dataset). However, we saw that these models failed to perform well on the thermal images. So we constructed our models using the building blocks from the above networks to extract the features from IR data and construct a disparity map.

Therefore, from the results shown in section 5, we can conclude that the modified PSMNet, along with CLAHE pre-processing and a super resolution network, outperforms all the other networks and generalizes well to the IR image dataset. We can also infer that using the model architectures designed explicitly for 3 channel RGB images, do not extend well to the one channel, low texture, low resolution IR images. We also observed poor performance of the models with spatial filters as these filters take away most of the useful information from the image. Frequency domain techniques also proved to be helpless due to excessive noise present in the images, at various frequencies. Considering specific features of the thermal data, model architectures have to be designed in a way that will fit in the framework of IR image datasets. CLAHE showed an improvement in the extraction of meaningful features from the IR images which proved to be an important pre-processing step. Super resolution network proved to be a meaningful addition to the network as well.

In the future, we plan to implement these modified models on the CMU data set and also plan to modify the existing model architecture to further improve the disparity estimation and reduce the D1 error.

Acknowledgments

We would like to acknowledge the assistance and guidance of Iljoo Baek, a PhD student working with Cylab in the ECE department at CMU and thank him for providing us with the CMU dataset.

References

[1] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network, 2018.

[2] Andrey Filippov and Oleg Dzhimiev. See far with tpnet: a tile processor and a cnn symbiosis, 2018.

[3] Andrey Filippov and Oleg Dzhimiev. Long range 3d with quadocular thermal (lwir) camera, 2019.

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] K. Hajebi and J. S. Zelek. Structure from infrared stereo images. In *2008 Canadian Conference on Computer and Robot Vision*, pages 105–112, 2008.

[6] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression, 2017.

[7] Shangzhen Luan, Chen Chen, Baochang Zhang, Jungong Han, and Jianzhuang Liu. Gabor convolutional networks. *IEEE Transactions on Image Processing*, 27(9):4357–4366, Sep 2018.

[8] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[9] Zhihao Wang, Jian Chen, and Steven C. H. Hoi. Deep learning for image super-resolution: A survey. *CoRR*, abs/1902.06068, 2019.

[10] Scott Sorensen Abhishek Kolagunda Michael O'Neal Brian Phelan Kelly Sherbondy Chandra Kambhamettu Wayne Treible, Philip Saponaro. Cats: A color and thermal stereo benchmark. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] Haofei Xu and Juyong Zhang. Aanet: Adaptive aggregation network for efficient stereo matching, 2020.

[12] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip H. S. Torr. Ga-net: Guided aggregation net for end-to-end stereo matching, 2019.

[13] Y. Zhang, Y. Li, Y. Kong, and B. Liu. Attention aggregation encoder-decoder network framework for stereo matching. *IEEE Signal Processing Letters*, 27:760–764, 2020.

[14] Yannick Wend Kuni Zoetgnande, Geoffroy Cormier, Alain-Jérôme Fougères, and Jean-Louis Dillenseger. Sub-pixel matching method for low-resolution thermal stereo images, 2019.

Division of Work

- Aditi Damle
  Worked on the implementation of GA-Net and the corresponding documentation. Also researched on frequency domain technique of DCT and developed the model for DCT-MLP model pre-processing methods trained the model on CATS dataset. Also genereated HOG pre-processed data for PSMNet model.
- Karan Tank
  Worked on the implementation of AANet and the corresponding documentation, and data pre-processing. Also worked on tuning and tweaking of PSM-Net architecture to better suit the Infrared Images. Implemented the super resolution model employing the iterative up-down sampling architecture. Developed parts of code for training and validation pipeline.
- Siddhant Jagtap
  Worked on the implementation of AANet and the corresponding documentation. Worked on developing code for the PSMNet building blocks from scratch and tried to make modifications to the existing architecture. Tried various hyperparameter tuning methods on the network architecture. Implemented average pooling in the residual connections to account for skip connections between tensors of unequal dimensions.
- Vaidehi Joshi
  Worked on the implementation of GA-Net and the corresponding documentation. Worked on modifying architecture of PSMNet with the team. Worked on generating inferno maps for CATS dataset and trained the baseline PSMNet and modified PSMNet architectures on the generated inferno dataset. Also, worked on the training and evaluation of CMU Dataset on baseline PSMNet and modified PSMNet architectures.