Monocular Depth Estimation using Adversarial Training

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Pallavi Mitra

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Junaed Sattar

June, 2020

Acknowledgements

I would like to thank my advisor, Professor Junaed Sattar for his valuable guidance, support and patience throughout the course of this project. Without his help, this thesis would not have been possible. I would also like to thank the members of my committee, Professor Victoria Interrante and Professor Changhyun Choi. I am thankful to the members of the Interactive Robotics and Vision Laboratory as well for sharing their knowledge with me.

Dedication

To my parents, my sister, my brother and my other family here in the United States who helped me survive Minneapolis.

Abstract

Monocular depth estimation is a fundamentally challenging problem in Computer Vision and a widely researched topic. It is particularly useful for Robotics applications where design constraints prohibit the use of multiple cameras. Monocular depth estimation finds widespread use in autonomous driving applications. Since the task is to estimate the depth associated with each pixel from a single image, rather than two or more, a global perspective of the scene is required to estimate it correctly. Various pixel-wise losses like reconstruction loss, left-right consistency loss, disparity smoothness loss, etc., capture the local scene information. However, they do not take into account the global consistency of the scene. Generative Adversarial Networks(GANs) are highly effective in capturing the global structure of the scene and producing real-looking images, so they have the potential of depth estimation from a single image. This work focuses on using adversarial training for a supervised monocular depth estimation task in combination with various pixel-wise losses. Some variants of GANs like Vanilla GAN, LSGAN, Wasserstein GAN, etc. have been explored and their associated advantages and drawbacks for the training have been presented. We observe that with minimal depth-supervised training in both the generator and discriminator, there is a significant reduction of error in depth estimation in a number of GAN variants.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Monocular depth estimation is a fundamentally ill-posed problem in Computer Vision. Two significant reasons are projection ambiguity and scale ambiguity. On the application of geometric transformation on a scene, the points in the two different scenes may map to the same location on the plane, see figure 1.1. As such, many 3D scenes can explain a single 2D image. Humans possess the ability to perceive depth from one image using monocular depth cues like the size of objects, texture, and linear perspective. Figure 1.2 shows how the spatial arrangement of the trees and other objects develops our intuition of the depth associated with every object present in the scene.

In order to relate these monocular depth cues effectively, it is essential to have a global view of the scene. Eigen *et al.* [12] addressed this issue for the first time using two-stage deep neural networks, one for estimating the depth structure using a global view and the other for refining the coarse predictions within local regions. The work presented in the thesis aims to address the issue of capturing global scene context by making use of Generative Adversarial Networks(GANs) [13]. Since GANs are capable of producing globally coherent images and the success of monocular depth estimation can be measured by how well global information can be extracted from a single image, GANs are an attractive choice. Our work is inspired by that of Godard *et al.* [14]. In this thesis, elaborate experiments have been done to show the impact of using different variants of GANs on a monocular depth estimation network (in our case, Godard's) using a small amount of ground truth information. A novel loss function using a two-stage discriminator network

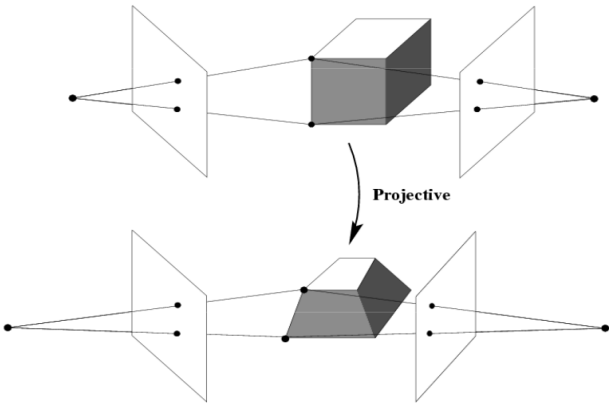

Figure 1.1: Figure shows how two points that have different 3D locations project on to the same 2D location as discussed above. Source: [4]

has been suggested to help improve depth estimation from a single image.

1.1 Depth Estimation using traditional Stereo Vision

Depth estimation refers to obtaining the distance from the camera of every object present in the scene. Like human eyes, two cameras are displaced horizontally from each other on the same plane to obtain two views of the scene. By comparing these two images, the relative depth information of the scene can be obtained as a *disparity map*, which measures the horizontal displacement in pixels of the corresponding image points. These disparity values are inversely proportional to the depth of the scene at the corresponding location of the pixel. Corresponding points in the left and right views can be obtained by comparing patches in the left image to those in the right. Dense matching algorithms are more popular and used in applications like image-based modeling [15], robot-vision, etc.
The major building blocks in stereo-matching algorithms are:

1. Matching Cost Computation
2. Matching Cost Aggregation
3. Disparity Computation and Optimization

Figure 1.2: Figure shows how the spatial arrangement of different objects, their varying texture helps humans perceive depth from a single image as discussed in section [1]. Source: [5]

4. Disparity Refinement

The dense algorithms can be of two types broadly, local (window-based), and global. Local algorithms, where the disparity computation at a given point depends only on intensity values within a finite window, make smoothness assumptions by aggregating support. Some of these dense algorithms can be broken down into steps 1, 2, 3 of the major building-blocks of a stereo-matching algorithm as described above. For example, the SSD (Sum of Squared Differences) [16] algorithm can be described as:

- At a given disparity, the matching cost is computed using the squared difference of intensity values.

- Matching cost aggregation is done by summing matching cost over square windows with constant disparity.
- Disparity computation is done by selecting the minimal aggregated value at each pixel.

Other local algorithms, however, combine the matching cost computation and aggregation steps and use a support region to compute the initial cost. These include normalized cross-correlation [17], rank transform [18]. These algorithms run very fast but suffer from low quality of obtained disparities.
On the other hand, global algorithms solve a global optimization problem by making explicit smoothness assumptions. These algorithms do not perform a cost aggregation step, but they perform a disparity assignment and minimize a global cost function consisting of a data term and a smoothness term. These algorithms differ in the type of minimization algorithm used, for example, loopy belief propagation [19] , expectation-maximization (EM) [20], graph cuts [21], simulated annealing [22],probabilistic (mean-field) diffusion [23], etc. The obtained disparities are better than those obtained from local methods. However, being iterative, it is time-consuming.
Apart from local and global algorithms, iterative algorithms form the third category of algorithms. These algorithms operate on an image pyramid where results from the coarser levels of computation are used for a more local search at the finer levels [24], [17].

1.2 Usefulness of CNNs in vision-based tasks

Convolutional Neural Networks play an important role in various computer vision tasks. The usage of CNNs increased greatly for vision-based tasks when they were first used by Alex Krizhevsky in 2012 for the ImageNet Classification challenge. The huge success of it, bringing down the classification error from 26 % to 15 % revolutionized the whole world of Computer Vision and the first Neural Network named AlexNet[25] was established. Eight years since then, many companies have used Neural Networks for their applications. While Facebook uses it for image tagging algorithms, Google uses CNNs for their Image searching algorithms, Amazon for product recommendations. Image Classification, Image Segmentation, Depth Estimation are among the few tasks where Neural Networks have proved to

be a champion and have helped generalize and outperform traditional image processing techniques for the above mentioned tasks.

AlexNet, the first of its kind, used a simple architecture consisting of 5 convolutional layers, max-pooling layers, dropout layers, and three fully-connected layers. It was used for the classification of images consisting of 1000 different categories. ReLU(Rectified Linear Unit, which was used to speed up training as compared to Tanh), usage of data augmentation, dropout layers, stochastic gradient descent for training were the main attractions of this model. Post that, many deep neural networks have been developed for various vision-based tasks [26], [27].

Among the shallow networks, VGGNet [28] has proved to be very efficient for Image Segmentation tasks. At the same time, ResNet [29], owing to its deep architecture, is well known in the computer vision community because of the concept of residual connections. VGG architecture was developed by the Visual Geometry Group at Oxford. In AlexNet, the kernel sizes are of size 11 and 5 in the first and second convolutional layers while in VGG, these are replaced by multiple 3×3 kernels stacked one after the other. Since multiple non-linear layers increase the depth of the network, it allows it to learn more complex features. The cost (number of parameters of the network) also decreases owing to a decrease in the kernel size. For instance, three 3×3 filters stacked together, with a stride of 1 has a receptive field size of 7, with the number of parameters being $3*9(C^2)$ while for a filter of size 7, it is $49*(C^2)$,(the receptive field size being constant) with C being the number of input and output channels. Three fully connected layers follow the convolutional layers. The width of the network is initially 64 and increased by a factor of 2 after every pooling layer. In VGG-Net, there are blocks of the same filter size, which are applied multiple times to extract more representative features. This concept of using blocks or modules became a common theme in the networks after VGG.

So far, we have seen that increasing the number of layers in a network increases the accuracy of the model; for example, AlexNet had five convolutional layers while VGGNet had 16. The accuracy of VGGNet was 90.6 %, while the accuracy of AlexNet was 84.7%. However, the gradients required to update the weights and biases in the initial few layers become extremely small, owing to repeated multiplications in back-propagation. So, the earlier layers are rarely trained. The fading away of gradients leading to loss of training is

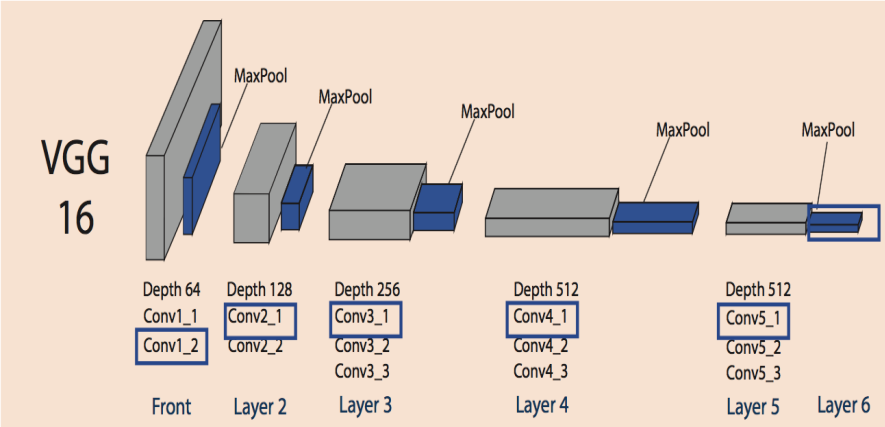

Figure 1.3: Block/Modular Layout of VGG-16. A variant of this, VGG-30 has been used as an encoder in the generator model as discussed in section 4.1
. Source: [7]

called the vanishing gradient problem.

Before ResNet, there were several ways to handle the vanishing gradient problem; for instance, [30] adds an auxiliary loss in a middle layer as extra supervision, but none of them completely resolved the problem. The basic idea of ResNet is adding an "identity shortcut connection" that skips one or more layers, as shown in figure 1.5.

Stacking up layers, in this case, should not degrade the network performance because it can be avoided by merely stacking identity mappings on the current network. Then, the resulting network would have a similar performance as the shallow network. It indicates that a deeper model should not produce a higher training error than its shallower counterparts. Through the changes mentioned, ResNets were learned with network depth as large as 152. It achieves better accuracy than VGGNet while being computationally more efficient than it. It achieves a top-5% accuracy of 95.51 in image classification.

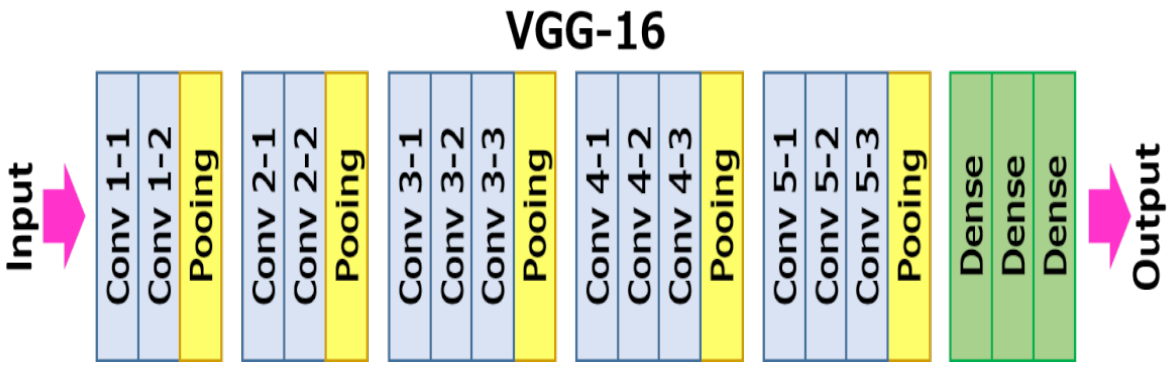

Figure 1.4: Layers of VGG-16 network in detail. Source: [6]

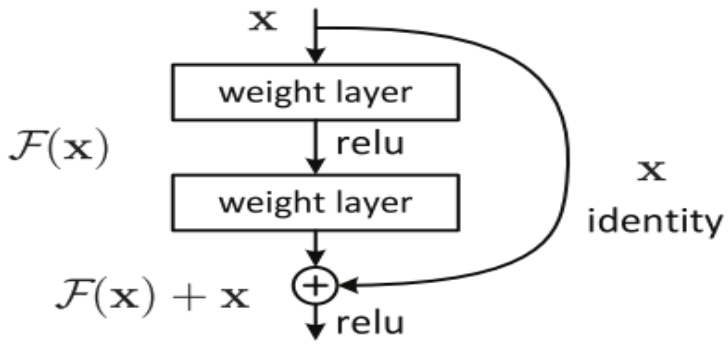

Figure 1.5: Residual block of the ResNet architecture which has proved to be more effective in solving deep learning problems due to its residual blocks. Source: [7]

1.3 Deep Learning vs. traditional Stereo Vision in Depth Estimation tasks

Deep learning based algorithms define a cost function and aim to learn the weights associated with a network by calculating the gradients of the cost function w.r.t the weights and updating the same using an optimization algorithm like SGD [31], RMSProp [32], ADAM [33] etc. The parameters (weights and biases) are tuned as per the gradients, and there are no hand-crafted features adopted by the cost function as it happens in traditional stereo-matching algorithms. Hand-crafted features limit the performance of these

algorithms. Traditional stereo-matching algorithms have discontinuous disparities and suffer from "bad pixels," which counts the number of differences between estimated disparity values and ground truth values that exceed a certain threshold. While these algorithms require post-processing, for example, the SGM algorithm requires a post-processing algorithm called Slanted Plane Smoothing (SPS) to smoothen out the obtained disparities. Most CNNs do not require any such post-processing step. A disparity smoothness loss term is integrated into the deep learning algorithm itself, which maintains high disparity changes in areas having high gradient changes and low disparity changes in areas having less gradient changes. This end-to-end training mechanism for depth estimation tasks helps obtain smoother disparities. On the contrary, stereo-matching based algorithms rely on hand-crafted features, which highly limits their performance. The figure shows that the SGM-method suffers from obvious bad pixels, while GC-Net provides a much smoother and more consistent depth map due to its end-to-end training mechanism.

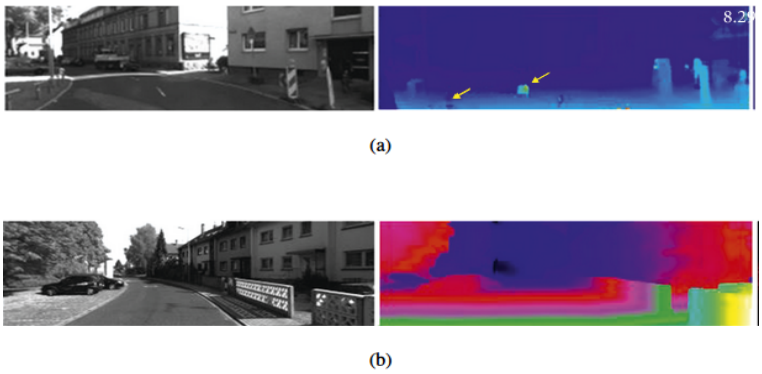

Figure 1.6: This figure shows how Deep Learning based methods produce smooth and consistent disparities, discussed above. The top disparity is that of GC-Net, and the bottom one is that of SGM. Source: [8]

1.3.1 Mono vs. Stereo

Deep learning based monocular depth estimation has been studied very widely owing to the rapid development of deep neural networks, and has achieved promising performance. MDE is widely used in SLAM or Visual Odometry to improve the mapping process, [34], [35], recover the absolute scale [36], [37], [38] and replace the RGB-D sensor in dense

mapping. It is commonly used in the self-driving car industry to measure the depth of an object and provide feedback about the location of the other objects with respect to the car. Since it is important to know the position, velocity, depth, thermal and more information about the surroundings of the self-driving vehicle, it becomes imperative to accommodate different kinds of sensors, thus producing space constraints. Mono cameras instead of a stereo one, help here. Also, monocular cameras are small, have low power consumption, and are easily accessible. Audi A8 uses a twin radar and monocular camera to implement its autonomous emergency braking system [39]. The company Autoliv provides automotive safety solutions to many car manufacturers in the world. Their mono vision system is capable of pedestrian detection, automated braking and traffic sign detection. [40]. Overall, the AV (Autonomous Vehicle) industry has been using monocular cameras for accomplishing different tasks like object, lane and traffic signs detection with good accuracy and its (AVs with mono cameras) sale is expected to take a surge by 2030 [41]. Another domain where it is useful is in the underwater autonomous vehicles, where space constraints, and power consumption plays a crucial role. Monocular cameras have also been beneficial in the surgical robotics domain where space constraints is a crucial factor [42].

1.3.2 Monocular depth cues

Research on Monocular depth estimation was first started by Eigen *et al.* [12], following which there has been active research in the field obtaining high accuracies in MDE which leads to the conclusion that neural networks can see depth in single images. In [43] four previously published networks are taken and investigated what depth cues they exploit. In their experiments, they found that these deep networks focus only on the vertical position of the object from the ground while ignoring the actual size of the objects present in the scene. Knowledge of the vertical position requires the network to estimate the camera pose. However, these networks can only partially estimate the camera pitch and the roll angles, and not the other pose parameters like the absolute translation and rotation of the second image with respect to the first. The inability to correctly estimate all parameters leads to inaccurate measurements in depth. The reason behind this inability is unknown in the literature, and further investigation about it can hugely improve the accuracy of

monocular depth estimation networks.

1.4 Usefulness of GANs over other methods in Monocular Depth Estimation

Very recently, the usage of GANs [13] in Monocular Depth Estimation has caught attention due to the capability of GANs to maintain global coherence in its generated samples. With the usage of GANs, the quality of generated data has increased over the years. GANs have been successfully used in image reconstruction [44], image segmentation [45], novel viewpoint estimation [46] and binocular depth estimation [47]. GANs are generative models, and they create new data instances that resemble the training data. These networks generate such realistic images by pairing a generator with a discriminator, which are both convolution neural networks. The generator learns to produce the target output given a latent variable while the discriminator learns to distinguish real data from the samples generated by the generator. A concept of an adaptable loss is introduced by the discriminator, which penalizes the samples generated by the generator that looks different from the real data.

Figure 1.7: Figure shows images of people generated by a GAN developed by NVIDIA. The realism of these fake images are extra-ordinary. Source: [9]

This confrontation between the generator G and the discriminator D helps the generator to learn which is based on the minimax problem as defined below:

$$\min_{G}\max_{D} E_{x\sim P_{gt}}[logD(x)] + E_{\hat{x}\sim P_G}[log(1 - D(\hat{x}))] \tag{1.1}$$

where x is the ground truth depth map, and $\hat{x}$ refers to the depth

$$L_{disp} = E_{d \sim P_{gt}}[logD(d)] + E_{\hat{d} \sim P_G}[log(1 - D(\hat{d}))] \tag{1.2}$$

$$L_{recons} = E_{I \sim P_{gt}}[logD(I)] + E_{\hat{I} \sim P_G}[log(1 - D(\hat{I}))] \tag{1.3}$$

map predicted by the generator.

1.5 Contributions

This thesis makes the following contributions:

- Adding Adversarial training to a Monocular Depth Estimation Model using a limited amount of ground truth depth information.
- Adding depth supervision to both generator and discriminator. Using a combination of image reconstruction loss and depth supervision loss in the discriminator and relatively weighing them to provide better feedback to the generator for efficient training.
- Experimentation with three different kinds of GAN frameworks, viz, Vanilla GAN, LS-GAN(Least Squares GAN), Wasserstein GAN, to study their impact on monocular depth estimation model training.

The rest of the thesis is organized as follows. Chapter 2 gives an overview of the Related Work in supervised, unsupervised, and semi-supervised Monocular Depth Estimation. The methodology, i.e., the different types of architecture and loss functions used, details on training, etc. are described in Chapter 3. Chapter 4 contains the results, experiments, and subsequent discussions. Chapter 5 contains discussions on the conclusion and scope for future work.

Chapter 2

Related Work

2.1 Monocular Depth Estimation: In Detail

As explained in section 1.3.1, Monocular depth estimation can be particularly useful in industries where it is essential to deal with space constraints [42]. It can also be applied in SLAM to improve the mapping process [34], [35], recover the absolute scale of a scene, [36], [37], [38] and avoid the use of an RGB-D sensor in dense mapping [48]. Besides, due to the low cost and small size of mono cameras as opposed to RGB-D sensors, they are getting increasingly popular for use in robotics applications like drones. Furthermore, a dense depth map of a scene can be estimated from a single image in an end-to-end manner as opposed to using a LiDAR. However, estimation of depth of a scene from a single image is a fundamentally ill-posed problem and requires effectively relating many different cues. As explained in section 1, a global perspective of the scene is required to estimate the depth of a scene from a single image accurately. Texture variation, scene context, occlusions, etc. also need to be considered. Evidently, these cues cannot be obtained by examining local image patches.

2.2 Delving Deep into the Methods and Techniques of Monocular Depth Estimation

As explained in the previous section, the key idea in monocular depth estimation is a combination of both global and local cues of a scene. To this end, the first work was done by Saxena *et al.* [49], where the global context information was modeled by using manually engineered features. In their approach, they used two types of features: absolute depth features to measure the absolute depth at a particular patch and relative features to estimate the relative depths between two patches, after dividing the whole image into small patches. Eigen *et al.* [12] combined global and local features by using two deep network stacks: one was used to make a coarse global prediction based on the entire image, and the other was used to make local refinements to these predictions. Li *et al.* [50] proposed a refinement method based on Conditional Random Field(CRF) instead of using two separate deep neural networks for global and local prediction. Since CRFs can characterize information among neighboring pixels and has been widely used in image segmentation [51], they have proved to be effective in refining the predicted coarse depth values. Works like [52], [53] support this concept. Besides, there have been recent developments in the use of convolutional neural networks (CNNs) [10], recurrent neural networks (RNNs) [54], variational auto-encoders (VAEs) [55] and generative adversarial networks (GANs) [56] to address the problem of monocular depth estimation. These can be categorized into supervised, semi-supervised, and unsupervised estimation methods. The subsequent sections describe the basic idea of each type of network and discuss research applying them to the MDE problem.

2.2.1 Supervised Monocular Depth Estimation techniques

In supervised techniques, the supervisory signal is that of the ground truth depth maps. Deep neural networks are trained to predict depth maps from a single image. The difference between the predicted depth map and the ground truth acts as the driving force to train these networks. The loss can be defined as under:

$$L_2(d, d^*) = \frac{1}{N} \sum_{i}^{N} \|d - d^*\|^2 \tag{2.1}$$

where d and d^* are the real and the predicted disparities respectively. Shown below are a few techniques of using depth supervision for monocular depth estimation:

Techniques based on different architectures

The first work to solve the ill-posed problem of monocular depth estimation using deep learning was done by Eigen et al. [12] The architecture composed of two stacks, one global and the other local to predict the depth maps in an end-to-end fashion. The training loss function is defined as:

$$L(d, d^*) = \frac{1}{N} \sum_{i}^{N} y_i^2 - \frac{\lambda}{N^2} \left(\sum_{i}^{N} y_i \right)^2 \tag{2.2}$$

where d^* is the true depth, d is the estimated depth, $y_i^2 = \log(d) - \log(d^*)$ and λ refers to the balance factor which is set to 0.5. In [57], Eigen *et al.* [] proposed a multi-scale generic framework capable of performing depth map estimation, surface normal estimation, and semantic normal prediction from a single image. An additional loss function that takes into account the local structural consistency was also added to the loss described in 2.2.

$$L_s(d, d^*) = \frac{1}{N} \sum_{i}^{N} \left[(\nabla_x D_i)^2 + (\nabla_y D_i)^2 \right] \tag{2.3}$$

where $D_i = \log(d_i) - \log(d_i^*)$, ∇ is the vector differential operator. This function calculates the gradient of the difference between real and predicted depths in both horizontal and vertical directions.

Owing to the outstanding performance of Resnet [29], Laina *et al.* [58] used Resnet to learn the mapping between a single RGB image and its corresponding depth image. Residual learning helped obtain higher accuracy. Also, the up-sampling blocks which replaced the fully connected layers in ResNet improved the resolution of the predicted depth map. Their training used the BerHu(reverse Huber) loss as shown below :

$$L_{Berhu}(d, d^*) = \begin{cases} |d - d^*|, & \text{if } |d - d^*| \leq c \\ \frac{(d-d^*)^2 + c^2}{2c}, & \text{if } |d - d^*| \geq c \end{cases} \tag{2.4}$$

where c is a threshold set as $\frac{1}{5}max_i(|d-d^*|)$. So, within a range defined by c, Berhu is equal to L_1 loss, while outside the range it is equal to L_2 loss. By using ResNet and the improved loss function as described above, this work achieved better results when compared to [59]. To improve the portability of a depth network on different cameras, Facil *et al.* [60] introduced the camera model into the depth estimation network, thus improving their generalization capabilities.

All the above methods described above achieve accuracies comparable to the ground truth. However, their capabilities are limited in embedded systems because of a large number of parameters in deep neural networks. The non-real-time performance of these networks can be improved by designing lightweight auto-encoder networks, as shown by Woft *et al.* [61] or by applying network pruning techniques.

Techniques based on Conditional Random Field

This section talks about using Conditional Random Fields in order to obtain continuous disparities as opposed to having a separate network for refining coarse disparities. Conditional Random Field has been widely used in solving image segmentation problems following the work of Jayasumana et al. [62]. They work towards applying similar labels to pixels having similar properties. This concept was first introduced by Li et al. [50] in the domain of monocular depth estimation. In their work, they designed a network to regress the depth map from multi-level image patches at the super-pixel level. The obtained depth map is then refined from the super-pixel level to the pixel level through CRF. The energy function which is minimized as below:

$$\mathbf{E(d)} = \sum_{i \epsilon S} \phi_i(d_i) + \sum_{(i,j) \epsilon \varepsilon_s} \phi_{ij}(d_i, d_j) + \sum_{c \epsilon P} \phi_c(d_c) \tag{2.5}$$

where S refers to the set of super-pixels, ε_s stands for the set of super-pixels that have a common boundary, and P is the set of pixel-level patches. $\mathbf{E(d)}$ is composed of three terms: (i) a data term for calculating the quadratic distance between the ground truth depth, d and the one predicted by the network $\bar{d}$ (ii) a smoothness term for relating neighboring super-pixels and (iii) an auto-regression model for taking into account the local relevant structures in the depth map. This information is used to improve the accuracy of depth maps due to the structural consistency between the depth maps and the semantic labels.

Wang et al. [53] presented a framework to jointly estimate the pixel-level depth map and semantic labels given a single image.

2.2.2 Unsupervised Monocular Depth Estimation Techniques

Instead of using the ground truth depth data for supervising the training, these methods use the underlying epipolar geometry present between the left and right pairs of images as their supervisory signal.

Techniques based on view synthesis

Unsupervised methods commonly treat MDE as a view synthesis problem where a particular frame in a given sequence of frames is reconstructed based on the knowledge of the camera's intrinsic parameters and the spatial transformation between consecutive frames. Zhou *et al.* [63] leveraged this idea to combine a depth network and a pose network to predict the depth and the transformation between corresponding frames, respectively. They estimated the corresponding pixels in the next frames based on the output of the two networks and used the reconstruction error between the predicted and the original frame as the supervisory signal. Other than using L_1 loss, they also incorporated SSIM loss to weigh their reconstruction error. Godard *et al.* [64]in their recent work on MDE showed that it is beneficial to use the minimum error between consecutive frames instead of the mean. This leads to improved error rates.

View reconstruction between consecutive frames, however, works only when the position of objects between consecutive frames do not change. This highly limits the efficiency of unsupervised MDE networks to real-world data. To address this issue, researchers use a mask to minimize the effect of dynamic objects present in the scene. Vijayanarasimhan *et al.* [65] designed an object mask network to model the dynamic objects present in the scene. They regress the motion of these objects along with the camera parameters to calculate the optical flow. However, usage of these masks increases the computation cost of the network and makes the training difficult. To relax the network of the difficult task of predicting the mask, Bian *et al.* [66] used geometry-based masks instead of estimating it by the deep network. This replacement reduced the error metrics.

Techniques based on combination with other vision-based tasks

In these methods, the geometric relationship between a task coupled with MDE is explored to enhance the loss function of the network. A combination of image segmentation and MDE are very popular in this aspect, and a cross-consistency term is introduced to help reduce the error metrics. Yin *et al.* [67] proposed a joint framework for estimating depth, ego-motion and optical flow.

2.2.3 Semi-supervised Monocular Depth Estimation techniques

Semi-supervised methods provide two advantages as compared to unsupervised techniques; an increase in accuracy and recovery of scale information from the semi-supervised signals. Also, the dependence on dense and expensive ground truth data is reduced as compared to supervised methods.

Crux of Semi-supervised model

A disparity map is estimated between the left and right images in a basic semi-supervised monocular depth estimation model. This disparity image is used to estimate the right image, given the left image by inverse warping. This reconstruction loss between the estimated right image and the original right image is used as a supervisory signal to constrain the training process. The process has been described, as shown in figure 2.1. The equation is defined as:

$$L_{recons} = \sum_{p} \|I_l(p) - I_r(p + Dis(p))\|^2 \tag{2.6}$$

where I_l is the left image, I_r is the right image and Dis(p) is the obtained disparity. This concept was first utilized by Garg *et al.* [10], where they also used a smoothness term in order to refine the coarse disparities and improve their continuity. Godard et al. [14] improved the above loss function by introducing a left-right consistency term. As the name suggests, the left and right disparity maps were made to be consistent with each other. Also, Garg's network was modified to predict the left and right disparity maps together, which was used to predict the right image from the left and vice-versa. A structural similarity index measurement (SSIM) [68] was introduced to strengthen the similarity between the real images and the synthesized images. Ramirez *et al.* [69] proposed

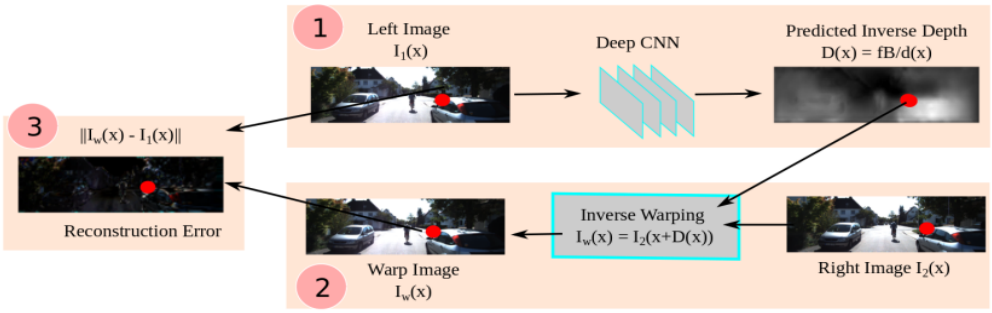

Figure 2.1: The crux network architecture of an unsupervised neural network introduced by Garg *et al.* as discussed in 2.2.3. Source: [10]

a framework for jointly estimating depth and semantic image segmentation. They designed an additional decoder stream to estimate semantic labels and trained the network in a supervised manner. Also, a cross-domain discontinuity term taking into account both the predicted semantic image and the predicted depth image, was applied to improve the smoothness of the predicted depth map. The additional cross-domain loss term showed a better performance than the previous smoothness loss terms used by Garg [10] and Godard [14].

Techniques based on stereo matching

Features from different views can also be combined to provide additional supervision. To this end, Tosi et al. [3] leveraged a stereo-matching strategy by combining features from different view-points. Their network framework contains three parts: a multi-scale feature extractor for extraction of high-level features, a disparity network for prediction of the disparity map, and a refinement network for refining the obtained disparities.

Techniques based on sparse ground truth availability

Depths obtained by lidar are sparse and are sometimes used as extra supervision in mono depth estimation. Kuznietsov *et al.* based the training of their network on the following

equation:

$$L_{recons} = \sum_{p \epsilon \Omega_{Z,l}} \|D_l(p) - G_l(p)\|_\delta + \sum_{p \epsilon \Omega_{Z,r}} \|D_r(p) - G_r(p)\|_\delta \tag{2.7}$$

where $\Omega_{Z,l}$ are the pixels where ground truth is available and $\|*\|_\delta$ is the berHu norm.
It has been proven how prior information can help in localizing information required in monocular depth estimation. Fei *et al.* [70] used the global orientation computed from inertial measurements as a priori information in order to constrain the normal vectors to the surfaces of objects. The prior information, in this case, improves the accuracy of the depth map significantly.

2.3 Role of GANs in Monocular Depth Estimation

Monocular depth estimation can be summarized as an image generation task, and encoder-decoder networks are often used for image generation tasks. However, such configurations trained using an L_1 or an L_2 loss produce blurry images, as mentioned in [71], since the output of the network is penalized based on the input image alone. The feedback from the generated output images in subsequent iterations is not taken into account in such configurations. Here, enters GANs. As discussed in 1.4, Generative Adversarial networks introduced by Goodfellow *et al.* [72] has received much attention in the research community in solving various problems in Computer Vision. GANs are very useful where generating globally coherent images are important. To this end, adversarial learning was first made use of in monocular depth estimation by Jung *et al.* [73]. Their network consisted of a Global Net and a Refinement Net to estimate the global and local 3D structures from a single image. To introduce adversarial training into the process, a discriminator was used to distinguish between the real and the predicted maps. The confrontation between the generator and the discriminator networks facilitates the training of the framework and helps maintain the global coherence in the predicted images. This confrontation can be best described by the following equation:

$$\min_G \max_D E_{x \sim P_{gt}}[logD(x)] + E_{\hat{x} \sim P_G}[log(1 - D(\hat{x}))] \tag{2.8}$$

where x is the real depth map, and $\hat{x}$ is the generated one. Apart from normal GANs used, conditional GANs have also been used to estimate depth from RGB videos, as evident in

[74]. Since the ground truth depth maps are not available in an unsupervised framework, the synthesized images and the real images are fed to the discriminator to trigger the adversarial training. [1] used GANs for monocular visual odometry and depth estimation. Zhao *et al.* [75] used cycle GAN [76] for the transformation between the synthetic and real domains to increase the size of the data set, and proposed an epipolar geometry-aware symmetric domain adaptation network (GASDA) to use the synthetic data effectively. The learning of their network was thus from two sources- first, from the epipolar geometry of the real domain and second, from the ground truth labels in the synthetic domain, thus achieving competitive results.

Inspired by the compelling use of GANs in solving various problems of stereo vision in general and MDE, the work presented in the thesis takes Godard's [14] model and extends the training in an adversarial fashion using ground truth depth images as extra supervision. As suggested by Isola *et al.* [44], in his Pix2Pix paper, the performance of a deep network increases significantly when a high-level adversarial loss is combined with a low-level reconstruction loss such as L_1. This is because an adversarial network is capable of updating both high and low-level details in an image generation task. Usage of GANs in works like novel view-point estimation by Huang *et al.* [77], predicting future frames in a video by Yin *et al.* [78], etc has been beneficial. The key idea in a good generator-discriminator network is making them equally competitive. As the usage of left-right consistency loss forces the generator to output better depth maps, having a depth map supervision in addition to the synthesized image ones can also make it a better discriminator. The subsequent sections talk about the architecture in detail.

Chapter 3

Method Overview

Estimating depth from a single image is essentially an image reconstruction problem first introduced by Garg *et al.* [10], where a convolutional neural network takes in a left image of a stereo pair and generates a disparity map which is used to construct the corresponding right image of the pair. The difference between the predicted image and the original rectified image serves as the main driving force for training the network.

In this thesis, Godard's [14] model for monocular depth estimation is used as the baseline to formulate the adversarial framework to devise supervised monocular depth estimation framework. The crux of the idea lies in formulating a novel loss function based on a min-max game played between two equal competitors. This confrontation is achieved by using a generator to generate depth maps and a discriminator to criticize it, thus leading to improvement. As described in the previous chapter, the key to a good adversarial network formulation lies in designing comparable discriminators and generators. A combination of an excellent generator and a weak discriminator would always generate images that lack appropriate feedback by the discriminator leading to poor improvement in the generated images over iterations. Similarly, a poor generator and a strong discriminator would always reject the generated images even though they appear close to being real. In light of the above observations, additional depth supervision could be added to both the discriminator and the generator with their relative weights, adjusted to obtain better reconstructed images and corresponding depth maps. The subsequent sections of this chapter discuss the adversarial formulation in detail and how reconstruction losses are used to combine with

adversarial losses and improvements over previous works.

3.1 Generator Model

The input to the generator network is a left image of a stereo pair that outputs two disparity maps(left-to-right and right-to-left). This will be referred to as d^r and d^l respectively. The left-to-right disparity map and the left image is used to generate the right image through inverse warping. Let the left and right reconstructed images be called $\tilde{}I^l$ and $\tilde{}I^r$ respectively. A good generator should predict d^r and d^l such that the reconstructed images, $\tilde{}I^l$ and $\tilde{}I^r$ are close to the real images of the stereo pair. Minimal distortions in the generated disparity maps will retain the shapes and structures of the reconstructed images post warping.

The training loss is governed by four individual losses, each measuring and minimizing a different aspect of the reconstruction process. In addition to the losses used by Godard, a depth supervision loss is added to the generator network to improve the training process. All the losses are calculated in four image scales starting at a size of 256×512 and resizing by a factor of 2 for the consecutive scales. Since the depth values captured by the LiDAR is sparse, the depth loss is calculated only at those points where the ground truth is available. The combined loss L_s is defined at each output scale s such that the total loss is defined as $L = \sum_{s=1}^{4} L_s$. L_s is composed of four terms as described under :

- **Appearance Loss or Image Loss** : The input image I_{ij}^l and the reconstructed image $\tilde{}I_{ij}^l$ are compared by using a combination of L1 and SSIM loss which is described as under:

$$L_{ap} = \frac{1}{N} \sum_{i,j} \alpha \frac{1 - SSIM(I_{ij}^l, \tilde{I_{ij}^l})}{2} + (1 - \alpha) \left\| I_{ij}^l - \tilde{I_{ij}^l} \right\| \tag{3.1}$$

  The value of α here is chosen as 0.85.

- **Disparity Smoothness Loss** : This loss forces the disparities to be smooth. Hence, in areas of high image gradients, where depth discontinuities often happen, this loss smoothens the effect.

$$L_{ds} = \frac{1}{N} \sum_{i,j} |\partial_x d_{ij}^l| e^{-\|\partial_x I_{ij}^l\|} + |\partial_y d_{ij}^l| e^{-\|\partial_y I_{ij}^l\|} \tag{3.2}$$

The weight associated with this loss is 0.1.

- **Left-Right Consistency Loss** : This loss makes the left-to-right (or right-to-left) disparity map output by the network to be equal to the projected right-view (left-view) disparity map. it can be defined as under:

$$L_{lr} = \frac{1}{N} \sum_{i,j} |d^l_{i,j} - d^r_{i,j+d^l_{i,j}}| \tag{3.3}$$

  The weight associated with this loss is 1.0.

- **Supervised or Lidar Loss** : This loss term measures the difference between the scaled disparity predicted by the network and the ground truth depth(converted to disparity) for the points where the ground truth is available(input depth map is sparse). this can be explained as under:

$$L_{sup} = \frac{1}{N} \sum_{i,j \epsilon \Omega} |d^l_{i,j} - Z| \tag{3.4}$$

  where Ω refers to the set of points where ground truth is available and Z is the ground truth depth which has been converted to disparity using the formula mentioned in 4.1. A weight of 0.25 has been associated with this loss.

The generator outputs a scaled disparity map at each scale which is gradually up-sampled by a factor of 2. Each of the above mentioned reconstruction losses are computed at every scale and summed up using $\sum_{s=1}^{4} L_s$ where,

$$L_s = \alpha_{ap} L_{ap} + \alpha_{ds} L_{ds} + \alpha_{lr} L_{lr} + \alpha_{sup} L_{sup} \tag{3.5}$$

where α_{ap} is the weight associated with the appearance loss,α_{ds} is the weight associated with disparity smoothness loss, α_{lr} is the weight for the left-right consistency loss, and α_{sup} is the supervised lidar loss's weight.

We use the first three weights associated with the losses following Godard *et al.* The weight associated with the supervised LiDAR loss has been obtained through hyperparameter tuning.

3.2 Discriminator Model

As mentioned by Godard [14], the error metrics for evaluation of the predicted disparities improve by 3% on addition of the left-right consistency loss to the pixel-wise reconstruction losses. Owing to the addition of this loss and the supervised LiDAR loss, the performance of the generator would increase significantly. As such, the discriminator which is added to distinguish between the real and fake images, needs to be made competitive to provide efficient feedback to the generator in order to generate depth maps that are both realistic and globally coherent. To this end, the present work improves over existing works by adding a combination of lidar depth supervision and image reconstruction loss - a two-step discriminator framework to compete well with the generator. Thus, the generator, in addition to the per-pixel losses, consists of two more adversarial losses, viz, the GAN loss via image reconstruction, and the GAN loss via the depth supervision. The relative weights of the two adversarial losses have been tuned to provide effective feedback to the generator.

3.3 Adversarial Extension of Base Model

For the adversarial training of the network, the baseline model is extended by a two-stage discriminator network, the first of which distinguishes between the real and the generated right images. The second one distinguished between the generated and the real disparity maps. The generator is thus trained using a combination of the four per-pixel losses as mentioned in section 3.1, the adversarial GAN loss from the first discriminator(stereo image reconstruction) and the adversarial GAN loss from the second discriminator(depth image reconstruction) with their relative weights being adjusted by hyper-parameter tuning. The overall network architecture can be seen in figure 3.1. To study the effectiveness of different GAN variants, the GAN loss, which is added to the image reconstruction loss, L_s described in section 3.5 is varied according to the GAN variant being used. These are described as under:

- Vanilla GAN
- LS(Least Squares) GAN

- Wasserstein GAN with gradient penalty

The final loss which is added to train the generator is:

$$L = L_s + \alpha_{gan-recons} L_{GAN-recons} + \alpha_{gan-depth} L_{GAN-depth} \tag{3.6}$$

The values of $\alpha_{gan-recons}$ and $\alpha_{gan-depth}$ are crucial to making the discriminator effective for the training of the generator. Too strict a regime would reject even a good depth image generated by the generator, which can impact the training process. These values are carefully chosen through hyperparameter tuning.

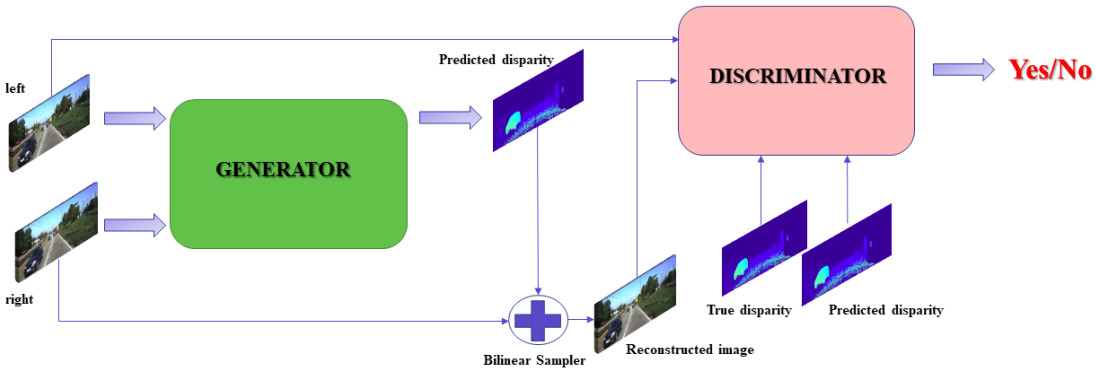

Figure 3.1: The network architecture of the proposed method. As discussed in section 3.2, the discriminator takes as input both the reconstructed pair (real and fake) and the depth pair of images. The input to the generator is the stereo pair of images.

Chapter 4

Experiment and Results

4.1 Implementation Details

In this thesis, we have extended Godard's work [14] to accommodate adversarial losses to improve the training process. For the generator, since it is an encoder-decoder structure, a VGG30 [28] architecture has been used as the encoder. The network outputs scaled (per unit width of the image) disparities at intermediate layers of the decoder while upsampling from the bottle-neck layer. For the discriminator, three different variants, viz Wasserstein-GAN (with gradient penalty) [79], Least-squares GAN [80] and Vanilla GAN have been used to experiment with the relative advantages of one network type over the other. For the Vanilla and LS variants, a 5-layer PatchGAN [44] has been used, while for the Wasserstein GAN, a 3-layer fully connected network has been used following Groenendijk *et al.* [2]. PatchGAN, first introduced by Zhu *et al.* in their work on Pix2Pix [81], penalizes the fake input to the discriminator in patches as opposed to classifying the whole image as real or fake. As shown by Isola *et al.*, using a PatchGAN has led to the better reconstruction of images as opposed to using ImageGAN, where the receptive field extends to the whole image. Since the task of the generator is more involved than that of the discriminator, a more complex network has been used for building the generator network. The number of parameters in the generator is 39 million, as opposed to 8 million in the discriminator.

4.2 Training, Hyperparameter Analysis, and System details

For the training, as suggested by [72], an Adam Optimizer[33] with a learning rate of 0.0001 has been used while for the discriminator, a stochastic gradient descent based optimizer is used. All the models described in this work have been trained for 50 epochs to enable a fair comparison. Following the work of [82], the learning rate has been reduced by half of its initial value after 30 epochs for every ten epochs for the remainder of the training.

Many tricks and tips, as suggested by [72], are applied to enable a stable training of the GAN. Some of these are label smoothing, which assigns a value of 0.9 and 0 to real and fake images as opposed to hard labels of 1 and 0. This has a positive impact on the training process as it allows the gradients to flow better. Also, in the generator, a random horizontal flip is applied to the input RGB and depth images. This significantly helps in improving the randomness and achieves lower error metrics. Following Godard's paper, color augmentation is also done on the input RGB images on the go. Based on the output of a random number generator, gamma, brightness and color shifts in the ranges of [0.8,1.2], [0.5,2.0] and [0.8,1.2], respectively, are done for each color channel separately.

The training has been done on the KITTI dataset [83], which consists of 29000 stereo pairs of an image of a car driving in various environments from highways to rural roads, etc. The typical size of an image is 1242×375. However, for training purposes, it has been resized to 256×512. For the depth maps, the official KITTI annotated depth maps(downloaded from [1]) are used as opposed to using raw LiDAR images which have been known to have inherent issues [56]. For choosing the hyperparameters, the following values are set, $\alpha_{ap} = 1.0, \alpha_{ds} = 0.1, \alpha_{lr} = 1.0.$, following the work of [14]. For choosing the weight of the supervised LiDAR loss, hyperparameter tuning was done, and a value of 0.25 yielded the best results. The weight of the adversarial losses was chosen as 0.5 for both the adversarial losses obtained through hyper-parameter tuning.

The training was done on a K40 GPU cluster provided by the Minnesota Supercomputing Institute. 2 GPUs were used for every training experiment. The source code is available at [2] .

[1] `http://www.cvlibs.net/download.php?file=data_depth_annotated.zip`

[2] `https://github.com/mitra052/SUP_MonoGAN`

4.3 Evaluation Criteria

During the evaluation, the disparities are converted to depth maps using the formula

$$depth = (focal\ length\ *\ Baseline)/disparity \tag{4.1}$$

As mentioned earlier, the predicted depth values are capped to a maximum depth of 80metres. As by Garg *et al.* [10], vertical centre crop of images are used. For quantitative evaluation, Absolute relative Distance(ARD), Squared Relative Distance(SRD), Root Mean Squared Error(RMSE), and log Root Mean Squared Error(log RMSE are used). For these metrics, lower values indicate better results. Also, accuracy within threshold t, where t $\epsilon[1.25, 1.25^2, 1.25^3]$ is used as another set of metrics. For these metrics, higher values indicate better results.

Following Godard, a post-processing step is applied to obtain smoother disparities and reduce the effect of occlusions, which create disparity ramps along the sides of the image. The input images are horizontally flipped so that the disparity ramps are now on the right side of the image. The final disparity map is obtained by combining the first 5% from the flipped disparity, the last 5% from the original disparity image, and the central part using a mean of the original and the flipped disparities. The final post-processing step doubles the inference time. However, it helps reduce the visual artifacts present in the predicted disparity map.

4.4 Datasets used

For testing and inference generation, following datasets have been used:

4.4.1 KITTI: Kitti split

The Kitti split consists of 200 test images. These are high-quality disparity images provided as part of the official training set, which covers 28 scenes. These disparity maps are much better in quality than the reprojected Velodyne laser depth values. Also, the maximum depth in the KITTI dataset is of the order of 80m. The predictions of all the networks are capped to this value. As observed in table 4.1, there is a significant improvement in the RMSE values on the addition of adversarial training to our baseline model. The

improvement in the addition of the second discriminator can also be observed in this case. It is to be noted that no batch normalization or residual learning has been used, demonstrating the sole effect of the adversarial training on a supervised monocular depth estimation model. In figure 4.1, we have shown qualitatively, how we obtain more continuous and crisp boundaries due to the usage of GANs.

Method	**ARD**	**SRD**	**RMSE**	**log RMSE**	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline	0.1252	1.4486	6.295	0.220	0.839	0.934	0.973
Proposed-1	**0.1225**	**1.3454**	**6.076**	**0.214**	**0.841**	**0.938**	**0.975**
Proposed-2	0.1233	1.3946	6.202	0.219	0.837	0.936	0.974
Baseline+pp	0.1205	1.2996	6.057	0.214	0.840	0.938	0.976
Proposed-1+pp	**0.1190**	**1.2277**	**5.860**	**0.208**	**0.843**	**0.941**	**0.978**
Proposed-2+pp	0.1188	1.2246	5.971	0.212	0.838	0.938	0.977

Table 4.1: **Baseline improvements on Kitti split.** For ARD, SRD and RMSE and log RMSE, lower values are better. For the rest, higher values are better. Proposed-1 refers to the usage of 1 discriminator and Proposed-2 refers to the usage of 2 discriminators. pp implies results with post-processing as discussed in 4.3. As we can observe, that without the post-processing, the adversarial training effect is more pronounced.

4.4.2 Kitti: Eigen split

The Eigen split is most popularly used by the research community for the sake of fairness of comparison with existing work. The split consists of 697 images covering a total of 29 scenes. As opposed to the Kitti split as discussed in 4.4.1, these ground truth values are not part of the official KITTI training data. Following the community, we generate the ground truth by reprojecting the 3D points as viewed from the LiDAR onto the left RGB camera. For fairness of comparison with existing methods, we use the crop suggested by [12] for evaluating all our models. The quantitative results and improvements over our baseline model can be found in table 4.2. It shows improvements in the obtained RMSE values compared to baseline with the maximum depths being capped to 80metres and 50metres. Results have also been shown with and without the post-processing step (marked as pp), as discussed in section 4.3. It is interesting to note that while the error metrics improve on the addition of a single discriminator, the values have increased slightly beyond the baseline on the introduction of the second discriminator. More details on it have been

explained in section 4.6. The qualitative results on different GAN variants on the Eigen split can also be viewed in 4.5, 4.6 and 4.7.

Method	**cap**	**ARD**	**SRD**	**RMSE**	**log RMSE**	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline	80m	0.1194	1.1048	5.392	0.224	0.843	0.932	0.968
Proposed-1	80m	**0.1167**	**1.0583**	**5.291**	**0.2222**	**0.8498**	**0.9333**	**0.9675**
Proposed-2	80m	0.1199	1.1245	5.452	0.229	0.840	0.929	0.966
Baseline + pp	80m	0.1150	1.0095	5.230	0.217	0.846	0.936	0.970
Proposed-1 + pp	80m	**0.1134**	**0.974**	**5.127**	**0.215**	**0.851**	**0.937**	**0.970**
Proposed-2 + pp	80m	0.1156	1.0232	5.287	0.221	0.843	0.933	0.969
Baseline	50m	0.1122	0.7974	4.047	0.209	0.856	0.941	0.972
Proposed-1	50m	**0.1070**	**0.7622**	**3.943**	**0.208**	**0.863**	**0.942**	**0.972**
Proposed-2	50m	0.1126	0.8048	4.090	0.2141	0.854	0.938	0.970
Baseline + pp	50m	0.1083	0.7333	3.932	0.203	0.860	0.944	0.974
Proposed-1 + pp	50m	**0.1069**	**0.7060**	**3.835**	**0.202**	**0.864**	**0.945**	**0.975**
Proposed-2 + pp	50m	0.1088	0.739	3.974	0.207	0.856	0.941	0.973

Table 4.2: **Baseline improvements on Eigen split of Kitti dataset.** For ARD, SRD and RMSE and log RMSE, lower values are better. For the rest, higher values are better. Proposed-1 refers to the usage of 1 discriminator. Proposed-2 refers to the usage of two discriminators. pp implies post-processing applied on the predicted disparity images as discussed in section 4.3.

4.4.3 CityScapes

To test the generalization properties of the network to other datasets, CityScapes [11] has been used. It consists of 22973 training images, 500 validation images, and 1525 test images. As noted by previous researchers during CityScapes evaluation, some images contain artifacts at the top and bottom of the images. Also, the cameras capture some part of the car on which the sensor is mounted. Since depth at such close distances can be difficult to be estimated with high accuracy, the top 50 and bottom 224 rows of the images are cropped. The quantitative results of the improvements in CityScapes data can be found in table 4.3. Similar to the trends in Kitti and Eigen split, this dataset also shows improvements with the addition of the adversarial training containing one discriminator only.

Method	ARD	SRD	RMSE	log RMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline	0.3660	9.1115	20.406	0.545	0.309	0.614	0.783
Proposed-1	**0.3646**	**8.9514**	**20.145**	**0.539**	**0.298**	**0.622**	**0.791**
Proposed-2	0.3677	8.9943	20.327	0.550	0.292	0.609	0.780
Baseline + pp	0.3344	7.3664	19.923	0.525	0.306	0.617	0.789
Proposed-1 + pp	**0.3332**	**7.2146**	**19.631**	**0.518**	**0.295**	**0.625**	**0.797**
Proposed-2 + pp	0.3374	7.3054	19.860	0.531	0.288	0.611	0.786

Table 4.3: **Baseline improvements in Cityscapes.** For ARD, SRD and RMSE and log RMSE, lower values are better. For the rest higher values are better. Proposed-1 refers to the usage of 1 discriminator. Proposed-2 refers to the usage of two discriminators. pp implies post-processing applied on the predicted disparity images as discussed in section 4.3.

4.4.4 Make3D

To further evaluate our model's generalization capability, we test the inference performance of our model (trained on KITTI) on the Make3D dataset [84]. Make3D dataset consists of 134 mono image sequences, which are different from those of Kitti and CityScapes, consisting of stereo image pairs. The improvements can be found in the table 4.4. Even though the calibration parameters of KITTI are very different from Make3D and the images' content are different, we have observed reasonable improvements in the error metrics on the introduction of adversarial training.

Method	ARD	SRD	RMSE	log RMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline	0.7505	16.2641	14.038	0.744	0.192	0.398	0.607
Proposed	**0.7559**	**16.5522**	**13.974**	**0.747**	**0.192**	**0.397**	**0.606**

Table 4.4: **Baseline improvements on Make3D dataset.** For ARD, SRD and RMSE and log RMSE, lower values are better. For the rest higher values are better.

4.5 Generalization Capability

Overall we have observed that our model trained on KITTI generalizes on the CityScapes and Make3D dataset. However, this improvement is only due to the introduction of one discriminator. The second discriminator produces error values that increase beyond the

baseline. This trend has been observed in all the datasets, and it can be improved by effectively choosing the weights of the associated loss components, which has been a challenge in our case, as explained in section 4.6.

4.6 Ablation study

For this work, we extend Godard's model in a supervised adversarial fashion. For future reference, we define our baseline model (as is described in all quantitative evaluation tables) as Godard's model plus the additional depth supervision loss. We perform an ablation study to study the effect of adding the adversarial training on our baseline model. The results can be found in table 4.5.

Method	**ARD**	**SRD**	**RMSE**	**log RMSE**	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Ours (Baseline)	**0.1194**	**1.1048**	**5.392**	**0.224**	**0.843**	**0.932**	**0.968**
Ours(Proposed-1)	**0.1167**	**1.0583**	**5.291**	**0.2222**	**0.8498**	**0.9333**	**0.9675**
Ours (Proposed-2)	0.1199	1.1245	5.452	0.229	0.840	0.929	0.966

Table 4.5: **Ablation study of adversarial training used.** The evaluation has been done on the Eigen split of KITTI dataset.

From the table, it can be observed that there is a drop in the RMSE values on the addition of the adversarial training to our baseline model. However, with the addition of the second discriminator, the error values have increased slightly. This can be improved with more careful consideration of the relative weights of the adversarial losses while training the network and by altering the relative rates with which the generator and the discriminator learn, which, as discussed in section 3 is crucial for efficient training and feedback provided to the generator. Optimizing the two mentioned parameters to produce an additive effect of the two discriminators for error reduction has been difficult in our case. However, as mentioned before, we have found that the addition of a discriminator network with one loss only, has helped reduce the error of MDE.

4.7 Comparison with state-of-the-art MDE techniques

As mentioned by Laina *et al.* [58], deep residual learning helps improve the accuracy of depth prediction from single images. In the thesis, our purpose is to show the usefulness of supervised adversarial training to solve the MDE problem without having to use the benefits of residual blocks in ResNet [29]. Hence, for fairness of comparison, only those methods where VGG-based architectures have been used, are selected for comparison with our method. Also, those techniques which use extra training data, for example, using Cityscapes for their training, has not been considered because it was observed that with more training data, the error metrics of all compared techniques decrease. While all of them are semi-supervised, meaning they either use some ground truth or both the left and right images during training, some use GANs. Our method outperforms the two mentioned GAN based methods, as shown in table 4.6, and outperforms one other method where ground truth depth supervision is used [85]. Another noteworthy point is that no Batch Normalization is used, which generally pairs up well as suggested by Groenendijk *et al.* [2] in their work. The decreased error metrics of Tosi *et al.* can be explained by their usage of dense semi-global matching based depth map as extra supervision to assist their training process while we use only a limited amount of ground truth depth data in the form of a sparse LiDAR depth map. With more depth information, our method is also expected to perform at par with theirs. It is to be noted that Pilzer *et al.*, which outperforms ours, use a student-teacher based architecture, a variant of adversarial training, to train their MDE model. Our increased error metrics as compared to theirs can be explained by the difficulty of training a GAN and finding the appropriate global minima. Mode-collapse is a very common problem faced while training GANs. Our method outperforms the error metrics obtained in [1], [85], and [2] due to the usage of supervised adversarial training of the MDE network.

4.8 Comparison with stereo

To compare the performance of monocular depth estimation techniques with their stereo counterparts, we have compared our method with two stereo depth estimation techniques (methods which use both left and right images during inference generation) evaluated on

Method	Year	ARD	SRD	RMSE	log RMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Almalioglu *et al.* [1]	2018	0.150	1.414	5.448	0.216	0.808	0.939	0.975
He *et al.* [85]	2018	0.123	1.3164	6.169	0.216	0.841	0.937	0.974
Chen *et al.* [86]	2019	0.118	0.905	5.096	0.211	0.839	0.945	0.977
Tosi *et al.* [3]	2019	0.111	0.867	4.714	0.199	0.864	0.954	0.979
Pilzer *et al.* [87]	2019	0.098	0.831	4.656	0.202	0.882	0.948	0.973
Groenendijk *et al.* [2]	2020	0.142	1.200	5.694	0.239	0.809	0.927	0.967
Ours	2020	**0.113**	**0.974**	**5.127**	**0.215**	**0.851**	**0.937**	**0.970**

Table 4.6: **Comparison with SOTA techniques**. It can be seen that our method outperforms the two GAN-based MDE technique suggested by [1] and [2], while [3] performs the best among the compared techniques. For detailed discussion, see text in section 4.7.

the Eigen split of Kitti dataset. As observed in table 4.7, the difference between stereo-based methods and ours is of the order of 1.117 meters. However, with the usage of ResNet as the encoder in the generator, Batch Normalization and using more data to train the network can bring this number down significantly to reduce the gap between the stereo and mono methods.

Method	ARD	SRD	RMSE	log RMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Godard *et al.* [14]	0. 068	0.835	4.392	0.146	0.942	0.978	0.989
Feng *et al.* [88]	0.065	0.673	4.003	0.136	0.944	0.979	0.991
Ours	**0.113**	**0.974**	**5.127**	**0.215**	**0.851**	**0.937**	**0.970**

Table 4.7: **Comparison with Stereo Depth Estimation techniques**. As discussed in section 4.8, the order of the maximum difference between the RMSE values of mono and stereo methods is 1.12 meters.

4.9 Discussion of results using GANs

The training of all the models in the experiments has been done using the KITTI dataset. In all the experiments, the baseline model is the model consisting of the four reconstruction losses mentioned in 3.1, without the use of any adversarial training. Initially, the effect of using three variants of GANs viz. Vanilla GANs, Least Squares(LS) GANs, and Wasserstein GANs on two types of loss components have been studied. These loss component types are having the L_1 loss combined with the supervised lidar loss and the other variant being

the full loss, as described in section 3.1. It has been observed that Wasserstein GAN yields the best results when using only L_1 loss combined with the supervised lidar loss. However, the performance of WGAN decreases when all the loss components are used. These results are consistent with the discussion of [2]. In general, WGAN produces extremely blurry disparities, which can be found in figure 4.1. It can be observed that Wasserstein GAN tends to over-smooth the obtained disparities. There are no sharp variations or transitions in the predicted disparity values. The other GAN variants, i.e., Vanilla GAN and LS-GAN, produce improvement in the predicted disparity values. The difference in the different error metrics compared to the baseline model is significant, with RMSE dropping from 6.06(baseline model) as opposed to 5.86 using Vanilla GAN. Without the post-processing step, the improvement of the supervised model with the adversarial addition is 3.5% in RMSE. These results are based on using only the adversarial loss owing to the generated images, as discussed in 3.2. The results are summarized in the table 4.1. Comparing the results with those of Aleotti *et al.* [56], we can observe that the decrease in RMSE is of the order of 2% as opposed to 3.5% in our case in the KITTI split. This improvement can be attributed to the usage of the adversarial losses related to image reconstruction. With a more careful usage of the hyperparameters used to relatively weigh the two loss components as described in 3.2 and 3.3, a further drop in RMSE values can be achieved. Our improvements in the Eigen split of the KITTI dataset are also more pronounced than those of Aleotti *et al.* [56] with the improvement percentage in RMSE being 0.8 % for their work and 1.90% in our case. Also, when the metric Absolute Relative Distance (ARD) metric is considered, they report no improvement w.r.t their model without the addition of adversarial training, while in our case, there is an improvement of 2.3%. It must be noted here that no batch normalization has been used in the training of any of the models. Hence, the improvements are solely obtained based on the introduction of the adversarial losses. The discussions of [2] however, state that their improvements are due to batch normalization in their trained models, and the partial decrease in error can be attributed to model initialization and batch normalization rather than the usage of adversarial losses. Based on the above observations and comparisons, it can be said that adversarial training favors a network more, which has depth supervision added to it. Furthermore, upon detailed analysis, we notice that the predicted disparities show some

trends. These are described as under and be found in the figure 4.1:

- It is observed that the disparities obtained through training using GANs(Vanilla GANs) produce disparities that are more continuous compared to scenarios where GANs are not used. The reconstruction seems to retain the shape and the orientation of structures present in the original input image to a better extent than not using adversarial losses.

- Another interesting observation is that the boundaries in the predicted disparity values are more distinct upon using GANs.

- It has also been observed that the usage of GANs makes it better to perceive different objects placed at different depths. These differences are not so pronounced in the baseline model, and boundaries of objects placed at different depths seem to be overlapping. This can partially explain that GANs take into account the global perspective instead of just penalizing the losses on a local patch. The differences in the object boundaries and disparities at different depths have been better taken care of due to the introduction of global adversarial losses.

- Yet another interesting observation was that the disparity layers are more in the predicted disparity images when GANs are used. This can be seen in the legend at the side of every image that where there are more variations in small range of disparity values, meaning that the disparity layers are more well-separated and distinct. This is an indication that only local but also global patches are considered by GANs. To the best of our knowledge, this observation has not been reported elsewhere about the benefits of using GANs in monocular depth estimation.

- Another consistent information is that Vanilla GANs perform the best in predicting the depth followed by LS-GAN. This has proved to be true for all the datasets where our method has been evaluated.

The above mentioned observations are in line with the decreased error metrics.

4.10 Speed in various platforms

For the model training, two NVidia Tesla K40m GPUs provided by the Minnesota Supercomputing Institute [3] , have been used. Each of the K40m GPU has 11 GB of RAM and 2880 CUDA cores. The inference speed using one of the GPUs is 76ms per image or approximately 13fps. However, this inference time can be reduced by removing the post-processing step, which would result in slightly spiky disparity values.

[3] https://www.msi.umn.edu/content/gpu-cluster

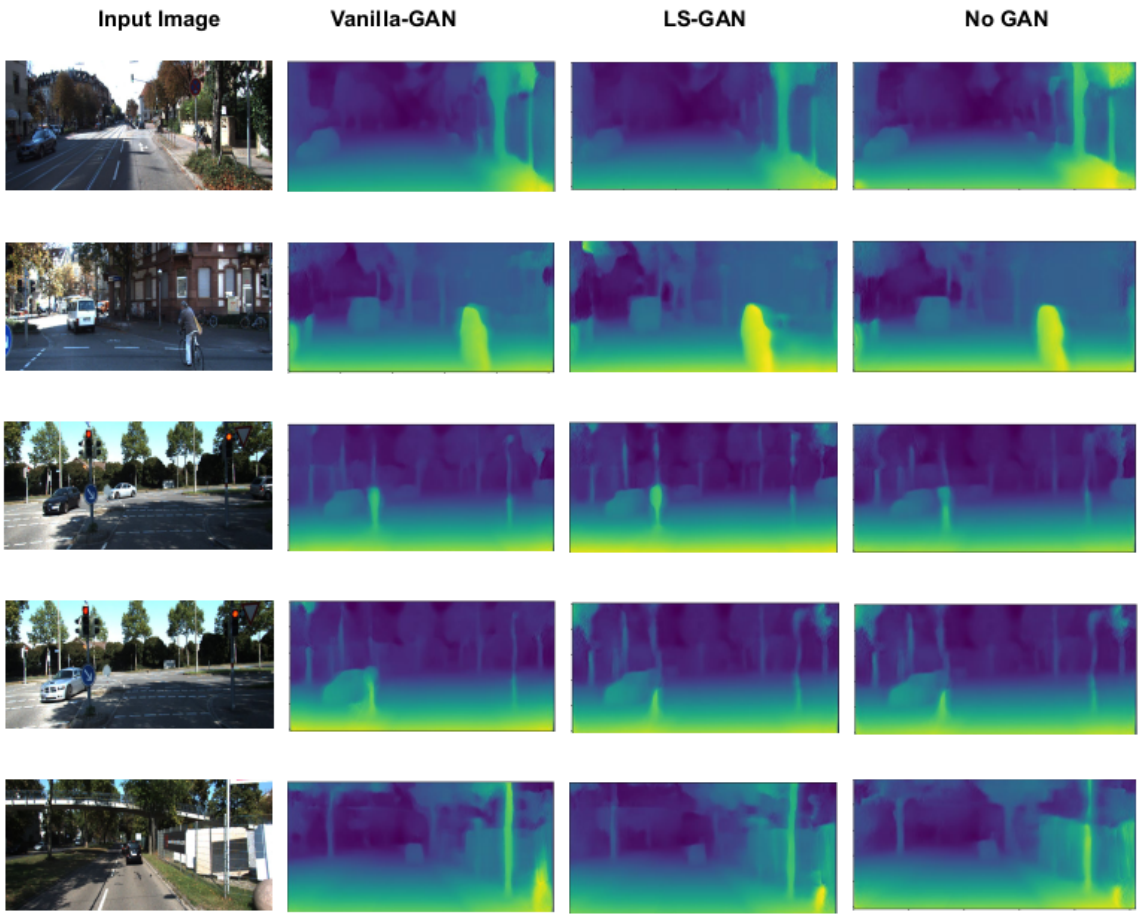

Figure 4.1: **Comparison of predicted disparities with different GAN variants.** As explained in section, 4.9, it can be seen in the bottom-most image, the pole is better distinguished than the back wall due to usage of Vanilla GAN. The same is observed in the second last image. Also, one can notice fewer overlaps in the structures in the second row of images from the top. Please note that the images have been resized for the purpose of viewing.

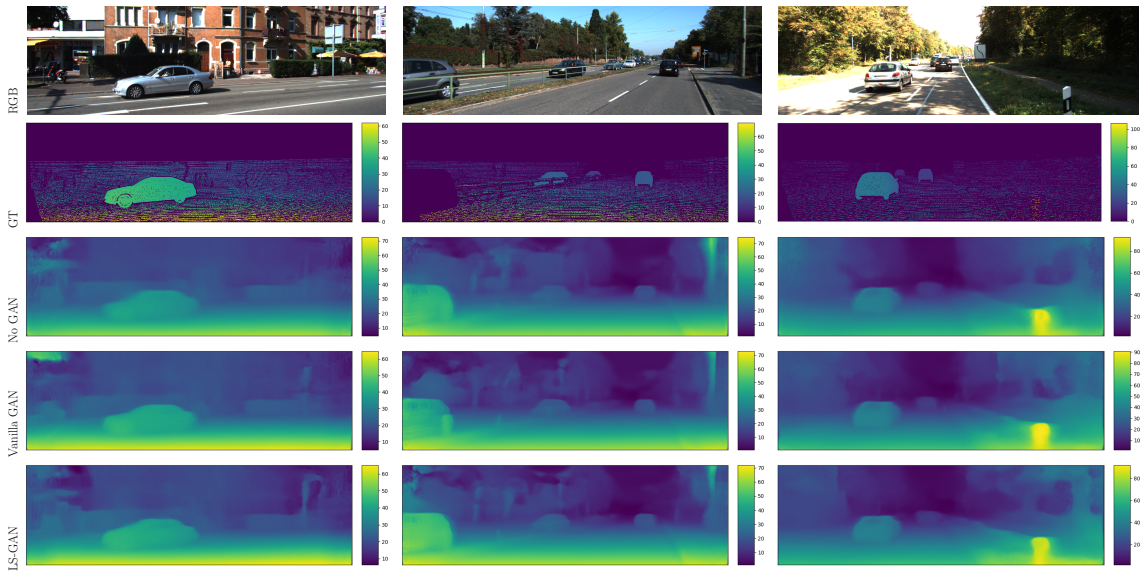

Figure 4.2: **Evaluation of different variants of GANs on the KITTI dataset.** Here, as shown in the legend along the side, higher intensity values represent distances which are closer to the camera. We can clearly see how Vanilla GANs produce more distinct disparity layers. This effect is more pronounced at lesser distances. For more discussion see text at 4.9.

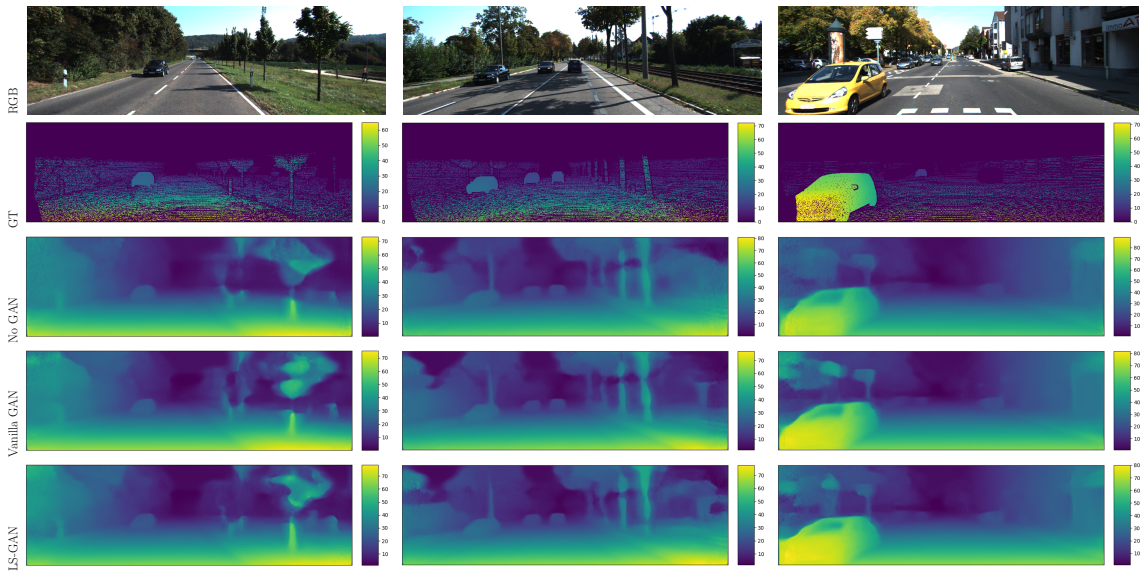

Figure 4.3: **Evaluation of different variants of GANs on the KITTI dataset.** Here, as shown in the legend along the side, higher intensity values represent distances which are closer to the camera. We can clearly see how Vanilla GANs produce more distinct disparity layers. This effect is more pronounced at lesser distances. For more discussion see text at 4.9.

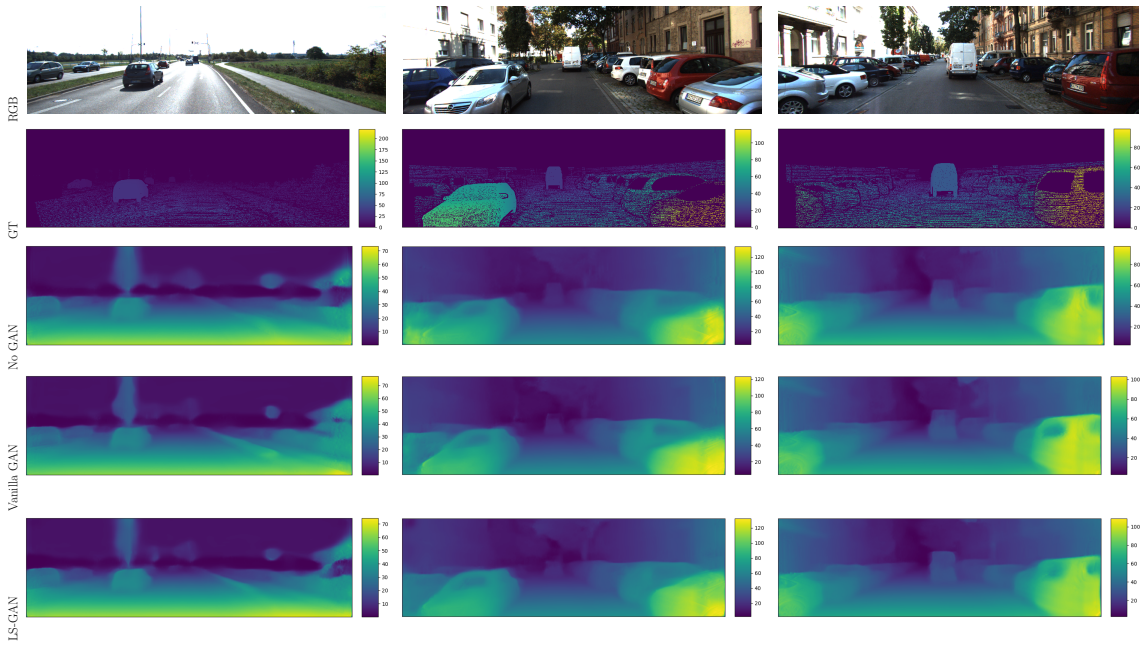

Figure 4.4: **Evaluation of different variants of GANs on the KITTI dataset.** Here, as shown in the legend along the side, higher intensity values represent distances which are closer to the camera. We can clearly see how Vanilla GANs produce more distinct disparity layers. This effect is more pronounced at lesser distances. For more discussion see text at 4.9.

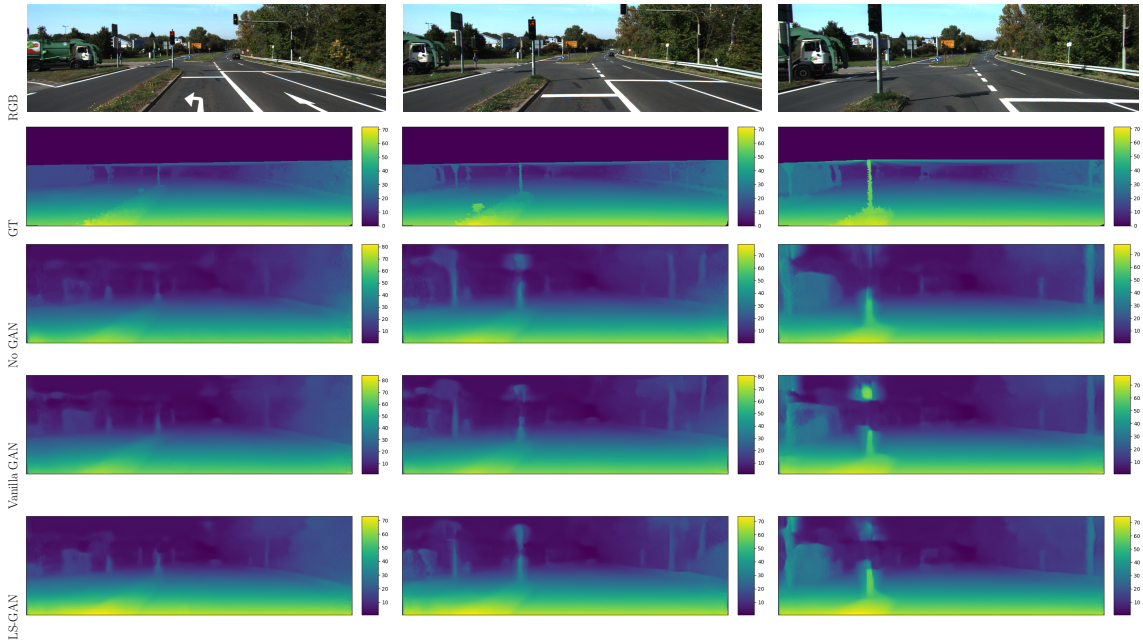

Figure 4.5: **Evaluation of different variants of GANs on the Eigen split of the KITTI dataset.** Here, as shown in the legend along the side, higher intensity values represent distances which are closer to the camera. The benefits of using GANs extends to this split as well and can be found in 4.2.

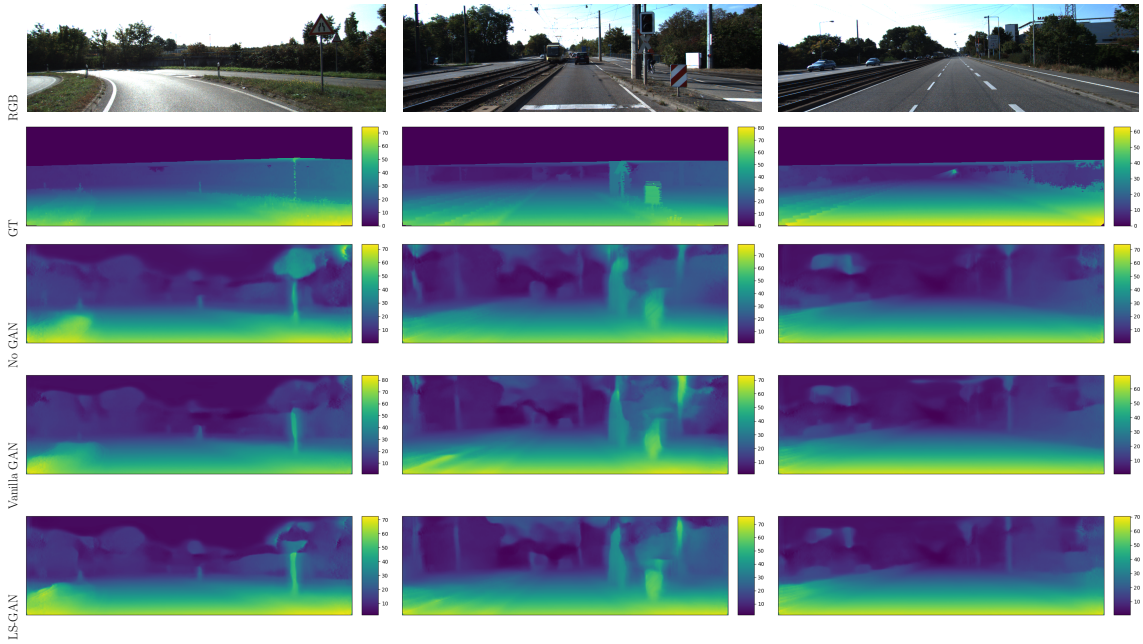

Figure 4.6: **Evaluation of different variants of GANs on the Eigen split of the KITTI dataset.** Here, as shown in the legend along the side, higher intensity values represent distances which are closer to the camera. The benefits of using GANs extends to this split as well and can be found in 4.2.

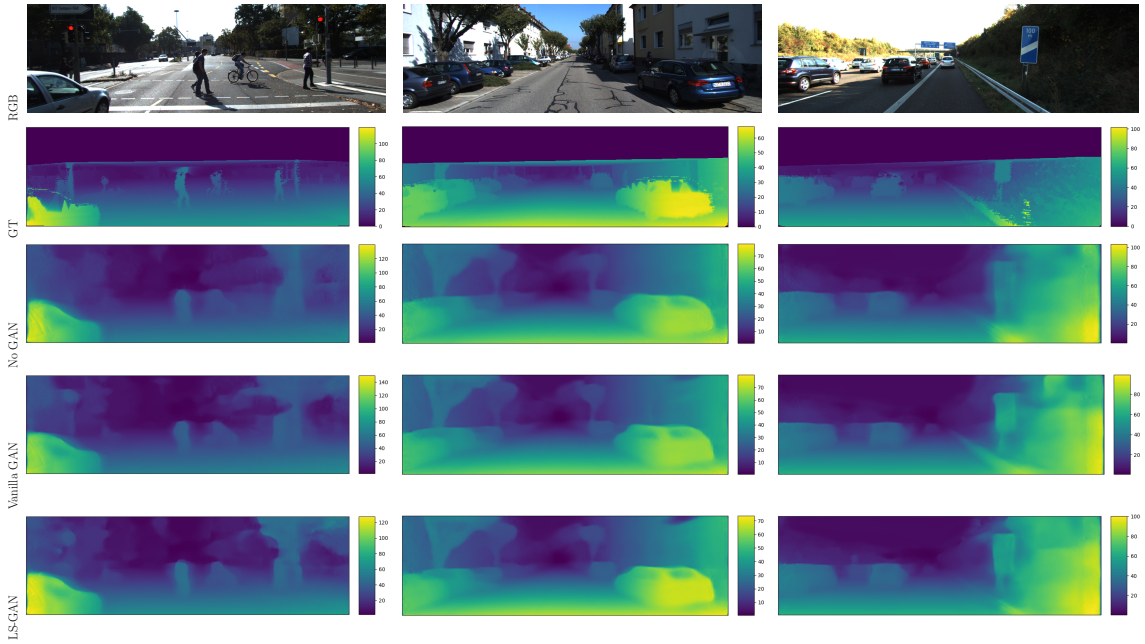

Figure 4.7: **Evaluation of different variants of GANs on the Eigen split of the KITTI dataset.** Here, as shown in the legend along the side, higher intensity values represent distances which are closer to the camera. The benefits of using GANs extends to this split as well and can be found in 4.2.

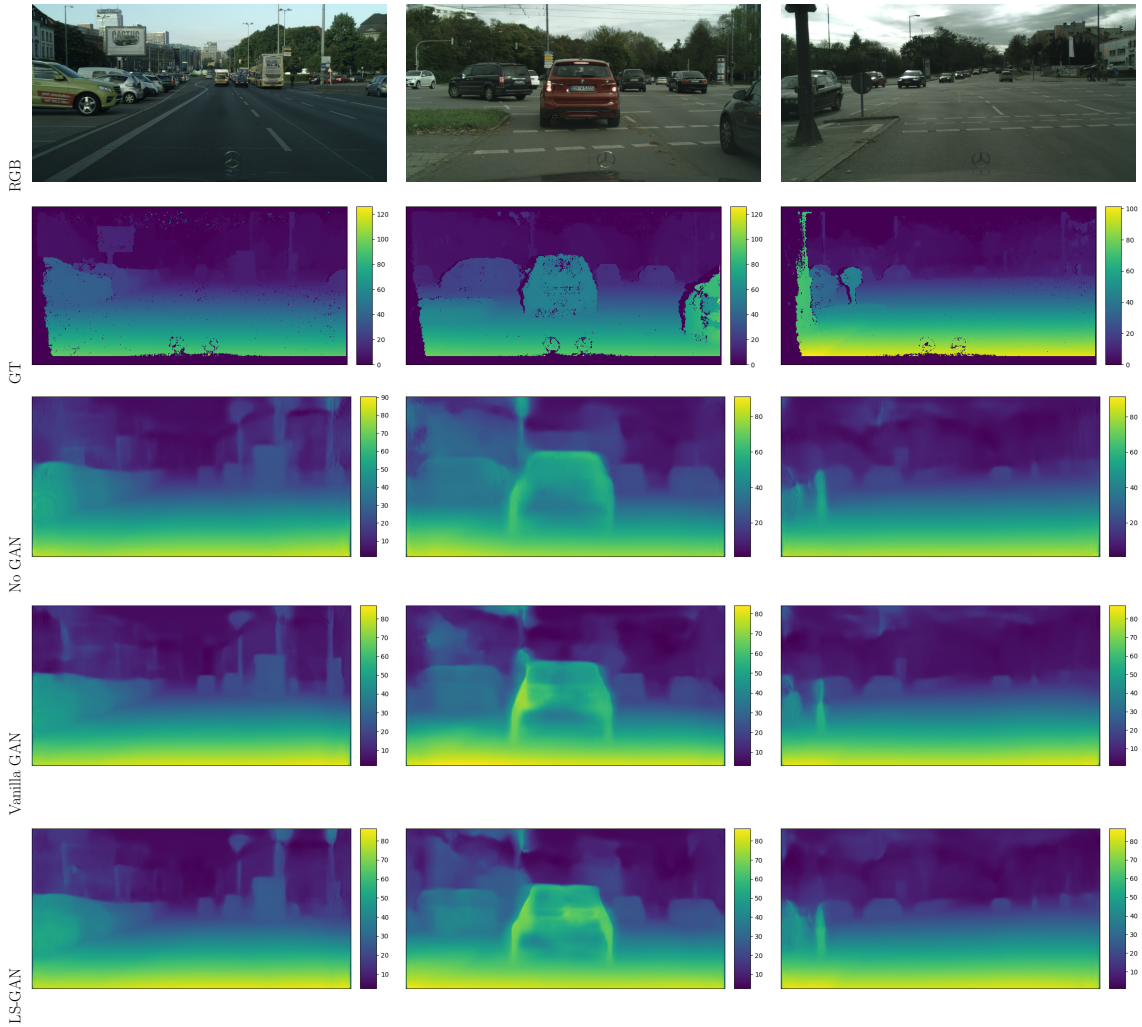

Figure 4.8: **Evaluation of different variants of GANs on the CityScapes [11] dataset.** Here, as shown in the legend along the side, higher intensity values represent distances which are closer to the camera. The benefits of using GANs extends to this split as well and can be found in 4.3.

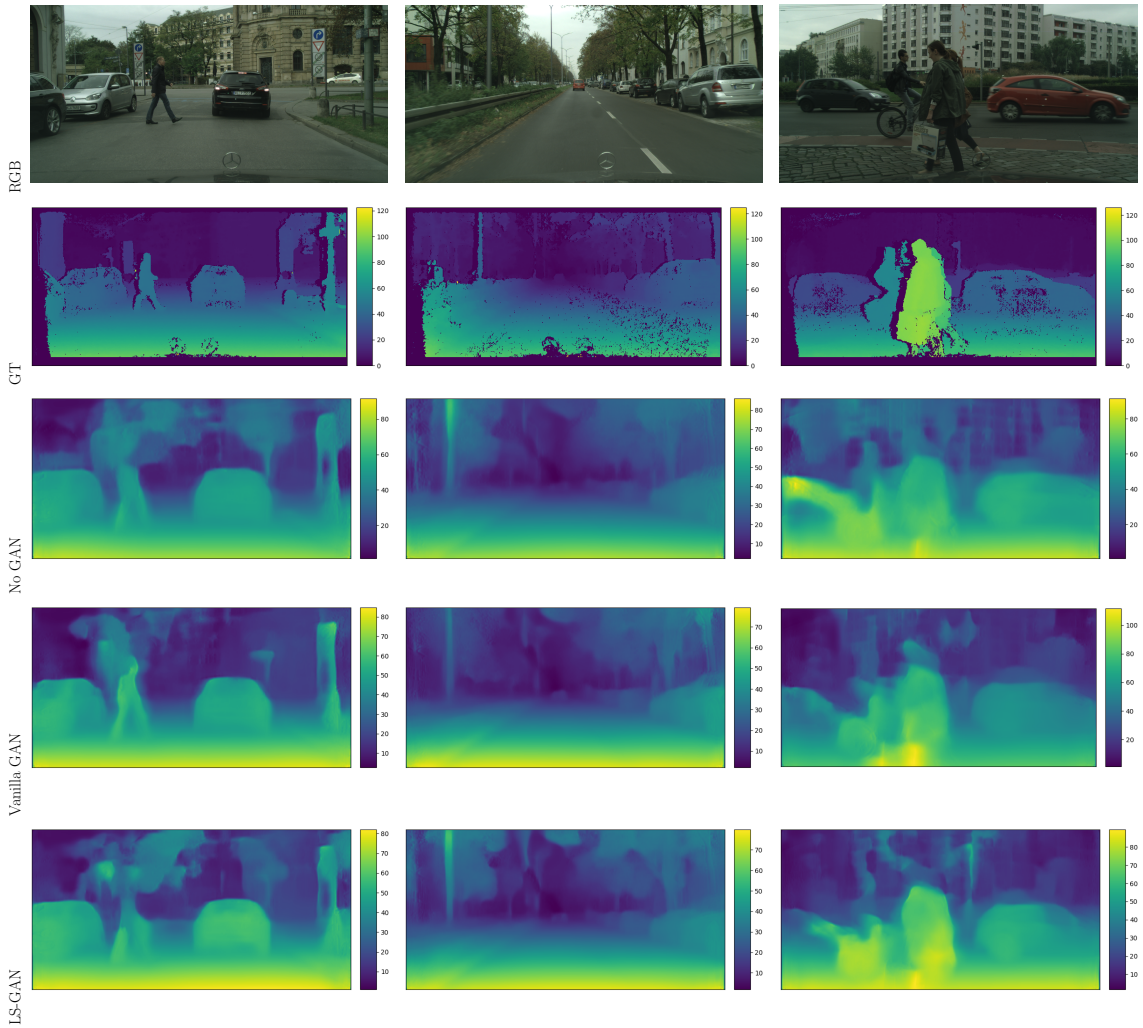

Figure 4.9: **Evaluation of different variants of GANs on the CityScapes [11] dataset.** Here, as shown in the legend along the side, higher intensity values represent distances which are closer to the camera. The benefits of using GANs extends to this split as well and can be found in 4.3.

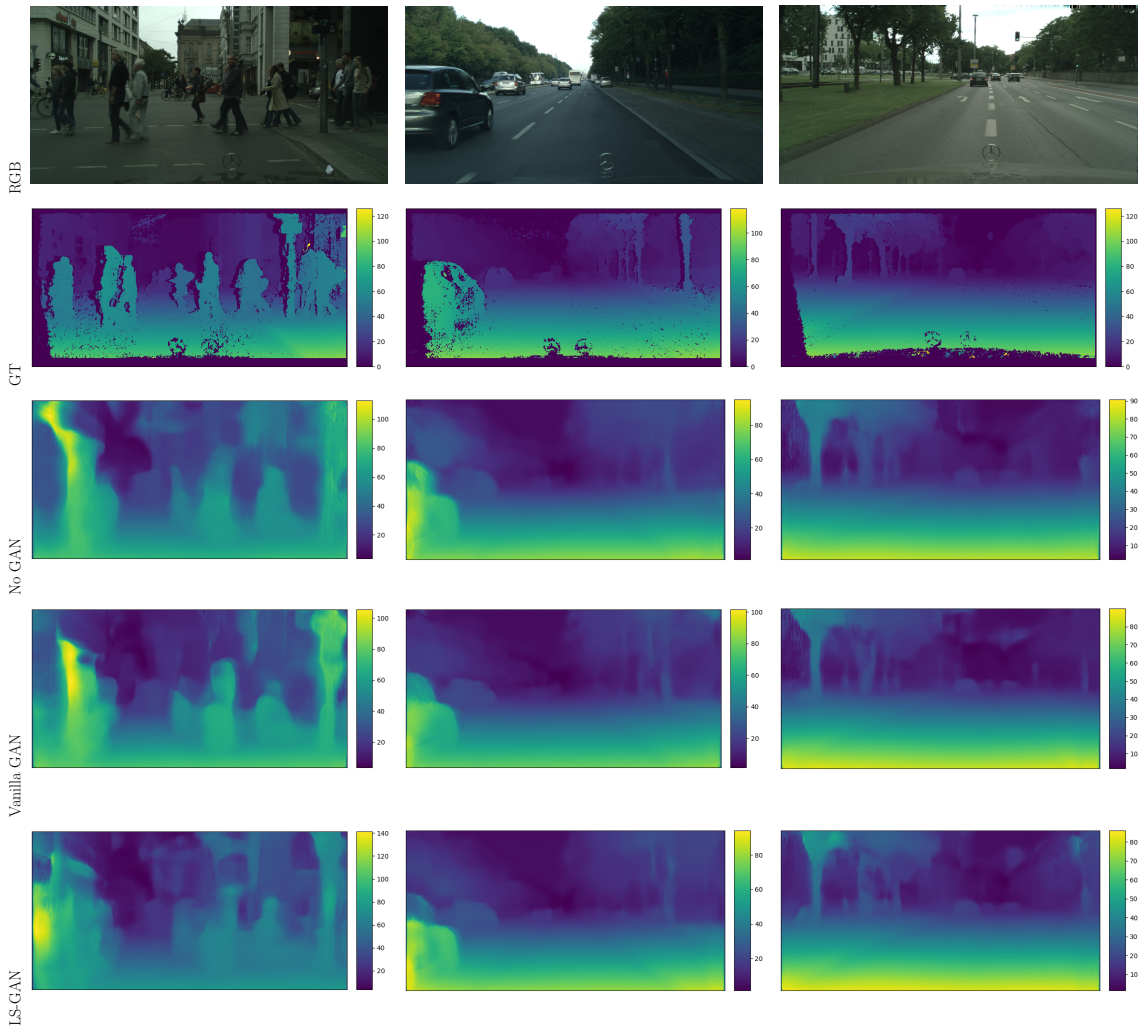

Figure 4.10: **Evaluation of different variants of GANs on the CityScapes [11] dataset.** Here, as shown in the legend along the side, higher intensity values represent distances which are closer to the camera. The benefits of using GANs extends to this split as well and can be found in 4.3.

Chapter 5

Conclusion

This thesis proposes the usage of adversarial training using a 2-step discriminator to deal with the problem of monocular depth estimation. While during training, the model requires an input of the stereo pair of images and a sparse lidar depth map; at inference, it can generate depth from a single image. The discriminator is trained to distinguish between the fake reconstructed images and the real images and also the fake and real disparity images. Through extensive experiments, it has been proved that, though not all, some adversarial training variants do help the cause of monocular depth estimation. A significant drop in the values of the RMSE can be noted using adversarial training.

We also notice that with the addition of the depth supervised discriminator, the error values increase partially as compared to our baseline. Combining several losses comes with its own set of problems. Carefully selecting hyperparameters with several loss terms poses a challenge in our case. This challenge increases the need for finding a more effective way of choosing hyperparameters for deep neural networks. Thus, we have been partially able to prove the fact that GANs do, in fact, help address the issue of capturing global scene context. With more careful consideration of the choice of hyperparameters, we can better address the capturing of the global scene context issue.

Though it is not clear as to why Wasserstein GAN tends to disrupt the obtained disparities and Vanilla and LS-GAN can improve it, it would be worthwhile to keep this a scope for future research. To the best of our knowledge, there has been no significant research in

finding out the effectiveness of one GAN variant over the other in monocular depth estimation. Also, usage of Conditional GANs, which would relate the input image to the depth, even more, to bring the global perspective into action, is an exciting direction of future research. Though this method of adding adversarial training has proved to be beneficial for Godard's Monodepth, it will be worthwhile to note its effect on other monocular depth estimation networks or depth estimation networks in general.

References

[1] Y. Almalioglu, M. R. U. Saputra, P. P. de Gusmao, A. Markham, and N. Trigoni, "Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5474–5480.

[2] R. Groenendijk, S. Karaoglu, T. Gevers, and T. Mensink, "On the benefit of adversarial training for monocular depth estimation," *Computer Vision and Image Understanding*, vol. 190, p. 102848, 2020.

[3] F. Tosi, F. Aleotti, M. Poggi, and S. Mattoccia, "Learning monocular depth estimation infusing traditional stereo knowledge," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9799–9809.

[4] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[5] *Photograph by Roland Denes in Unsplash*. [Online]. Available: https://unsplash.com/s/photos/depth-road?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

[6] "Layers of VGG network," https://towardsdatascience.com/distracted-driver-detection-using-deep-learning-e893715e02a4.

[7] "VGG architecture," https://towardsdatascience.com/deep-learning-for-medical-image-analysis-7989c8ed62a7.

[8] K. Zhou, X. Meng, and B. Cheng, "Review of stereo matching algorithms based on deep learning," *Computational Intelligence and Neuroscience*, vol. 2020, 2020.

[9] "Images generated by Style GAN," https://developers.google.com/machine-learning /gan.

[10] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *European Conference on Computer Vision*. Springer, 2016, pp. 740–756.

[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[12] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, pp. 2366–2374.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[14] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 270–279.

[15] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[16] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: theory and experiment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 9, pp. 920–932, 1994.

[17] M. J. Hannah, "Computer matching of areas in stereo images," STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, Tech. Rep., 1974.

[18] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *European conference on computer vision*. Springer, 1994, pp. 151–158.

[19] J. Sun, H.-Y. Shum, and N.-N. Zheng, "Stereo matching using belief propagation," in *European Conference on Computer Vision*. Springer, 2002, pp. 510–524.

[20] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," *International Journal of Computer Vision*, vol. 35, no. 3, pp. 269–293, 1999.

[21] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[22] J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *Journal of the american statistical association*, vol. 82, no. 397, pp. 76–89, 1987.

[23] D. Scharstein and R. Szeliski, "Stereo matching with nonlinear diffusion," *International journal of computer vision*, vol. 28, no. 2, pp. 155–174, 1998.

[24] A. Witkin, D. Terzopoulos, and M. Kass, "Signal matching through scale space," *International Journal of Computer Vision*, vol. 1, no. 2, pp. 133–144, 1987.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[26] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[27] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro, "Improving semantic segmentation via video propagation and label relaxation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8856–8865.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[31] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[32] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] S. Y. Loo, A. J. Amiri, S. Mashohor, S. H. Tang, and H. Zhang, "CNN-SVO: Improving the mapping in semi-direct visual odometry using single-image depth prediction," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5218–5223.

[35] "SVO: Fast semi-direct monocular visual odometry, author=Forster, Christian and Pizzoli, Matia and Scaramuzza, Davide," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 15–22.

[36] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[37] X. Yin, X. Wang, X. Du, and Q. Chen, "Scale recovery for monocular visual odometry using depth estimated with deep convolutional neural fields," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5870–5878.

[38] N. Yang, R. Wang, J. Stuckler, and D. Cremers, "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 817–833.

[39] "Collision avoidance in Audi," https://en.wikipedia.org/wiki/Collision_avoidance_system.

[40] *Autoliv*, https://www.autoliv.com/.

[41] "Seeing the road ahead: The importance of cameras to self-driving vehicles," https://loupventures.com/seeing-the-road-ahead-the-importance-of-cameras-to-self-driving-vehicles-2/.

[42] K.-T. Song and C.-J. Chen, "Autonomous and stable tracking of endoscope instrument tools with monocular camera," in *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2012, pp. 39–44.

[43] T. van Dijk and G. C. de Croon, "How do neural networks see depth in single images?" *arXiv preprint arXiv:1905.07005*, 2019.

[44] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[45] M. Ghafoorian, C. Nugteren, N. Baka, O. Booij, and M. Hofmann, "El-gan: Embedding loss driven generative adversarial networks for lane detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[46] Y. Galama and T. Mensink, "Iterative gans for rotating visual objects," 2018.

[47] A. Pilzer, D. Xu, M. Puscas, E. Ricci, and N. Sebe, "Unsupervised adversarial depth estimation using cycled generative networks," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 587–595.

[48] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6243–6252.

[49] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Advances in neural information processing systems*, 2006, pp. 1161–1168.

[50] B. Li, C. Shen, Y. Dai, A. Van Den Hengel, and M. He, "Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1119–1127.

[51] Q. Huang, M. Han, B. Wu, and S. Ioffe, "A hierarchical conditional random field model for labeling and segmenting images of street scenes," in *CVPR 2011*. IEEE, 2011, pp. 1953–1960.

[52] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2024–2039, 2015.

[53] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille, "Towards unified depth and semantic prediction from a single image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2800–2809.

[54] R. Wang, S. M. Pizer, and J.-M. Frahm, "Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5555–5564.

[55] P. Chakravarty, P. Narayanan, and T. Roussel, "Gen-slam: Generative modeling for monocular simultaneous localization and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 147–153.

[56] F. Aleotti, F. Tosi, M. Poggi, and S. Mattoccia, "Generative adversarial networks for unsupervised monocular depth prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[57] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658.

[58] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth international conference on 3D vision (3DV)*. IEEE, 2016, pp. 239–248.

[59] E. Shelhamer, J. T. Barron, and T. Darrell, "Scene intrinsics and depth from a single image," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 37–44.

[60] J. M. Facil, B. Ummenhofer, H. Zhou, L. Montesano, T. Brox, and J. Civera, "Cam-convs: camera-aware multi-scale convolutions for single-view depth," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 11 826–11 835.

[61] D. Wofk, F. Ma, T.-J. Yang, S. Karaman, and V. Sze, "Fastdepth: Fast monocular depth estimation on embedded systems," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6101–6108.

[62] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1529–1537.

[63] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1851–1858.

[64] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 3828–3838.

[65] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "Sfm-net: Learning of structure and motion from video," *arXiv preprint arXiv:1704.07804*, 2017.

[66] J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *Advances in neural information processing systems*, 2019, pp. 35–45.

[67] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1983–1992.

[68] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[69] P. Z. Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. Di Stefano, "Geometry meets semantics for semi-supervised monocular depth estimation," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 298–313.

[70] X. Fei, A. Wong, and S. Soatto, "Geo-supervised visual depth prediction," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1661–1668, 2019.

[71] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.

[72] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[73] H. Jung, Y. Kim, D. Min, C. Oh, and K. Sohn, "Depth prediction from a single image with conditional adversarial networks," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 1717–1721.

[74] K. Gwn Lore, K. Reddy, M. Giering, and E. A. Bernal, "Generative adversarial networks for depth map estimation from rgb video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1177–1185.

[75] S. Zhao, H. Fu, M. Gong, and D. Tao, "Geometry-aware symmetric domain adaptation for monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9788–9798.

[76] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[77] R. Huang, S. Zhang, T. Li, and R. He, "Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2439–2448.

[78] X. Yin, H. Wei, X. Wang, Q. Chen *et al.*, "Novel view synthesis for large-scale scene using adversarial loss," *arXiv preprint arXiv:1802.07064*, 2018.

[79] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, 2017, pp. 5767–5777.

[80] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.

[81] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[82] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[83] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[84] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 824–840, 2008.

[85] L. He, C. Chen, T. Zhang, H. Zhu, and S. Wan, "Wearable depth camera: Monocular depth estimation via sparse optimization under weak supervision," *IEEE Access*, vol. 6, pp. 41 337–41 345, 2018.

[86] Y. Chen, C. Schmid, and C. Sminchisescu, "Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7063–7072.

[87] A. Pilzer, S. Lathuiliere, N. Sebe, and E. Ricci, "Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9768–9777.

[88] T. Feng and D. Gu, "Sganvo: Unsupervised deep visual odometry and depth estimation with stacked generative adversarial networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4431–4437, 2019.

Appendix A

Glossary and Acronyms

Care has been taken in this thesis to minimize the use of jargon and acronyms.

A.1 Acronyms

Table A.1: Acronyms

Acronym	Meaning
CNN	Convolutional Neural Network
MDE	Monocular Depth Estimation
AV	Autonomous Vehicles
VAE	Variational Auto-encoder
GAN	Generative Adversarial Network
RNN	Recurrent Neural Network
CRF	Conditional Random Field