



UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Depth estimation on synthesized stereo image-pairs using a generative adversarial network

Sverre Boer
MSc. Thesis
July, 2021

Supervisors:

Prof. Poel, M.
MSc. Conde Moreno, L.
MSc. Niesink, B.

Info Support B.V.
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

This research presents a novel method for depth estimation on synthesized stereo image-pairs. The goal of this research is to explore the possibilities of generative adversarial networks and improve the quality of existing depth estimation networks. This is done by synthesizing a stereo image-pair from a single-view image and using this stereo image-pair image, the depth is estimated. For both actions, i.e. the synthesis and the depth estimation, a generative adversarial network is trained.

The method is mainly based on building a cycle consistency generative adversarial network and finding the most optimal network architecture and training methods, for the synthesis network and for the depth estimation network. In the conducted experiments; the influence of the identity loss function is measured, as well as various network architectural changes in both the generator- and discriminator model and the discriminator's ability to learn is restricted. We extracted the four most promising model configurations and trained a full-scale models. The dataset that was used to train our models, contained ground-truth depth maps that have been estimated by other depth estimation networks. Those have been evaluated using the FID score, RMSE metric and visual inspection.

The main findings were that the stereo image-pair synthesis network performed better than expected, because it was able to quite successfully transform the single-view image's perspective. An improvement to this network would be to improve the quality of the synthesized image. The depth estimation network was able achieve fairly okay results. The per-pixel quality of the depth estimation can be improve quite a lot. Nonetheless, interesting to see was that our model outperformed the ground-truth depth maps that were estimated by state-of-the-art depth estimation networks: where the ground truth depth map was wrong, our depth prediction was more correct.

Contents

Abstract	iii
1 Introduction	1
1.1 Motivation	1
1.2 Problem statement	3
1.3 Context	4
1.4 Scope	4
1.4.1 Requirements	5
1.4.2 Challenges and limitations	6
1.5 Research questions	6
2 Background information	9
2.1 Depth estimation using epipolar geometry	9
2.1.1 Two camera setup	9
2.1.2 Epipolar rectification	10
2.1.3 Depth from triangulation	11
2.2 Depth estimation using stereo view	12
2.2.1 Cost aggregation based on rectangular windows	12
2.2.2 Cost aggregation based on unconstrained windows	13
2.3 Matching cost	14
2.3.1 Semi-global matching	14
2.3.2 Local guided aggregation	15
2.3.3 Disparity refinement	16
2.4 Depth estimation using monocular depth estimation networks	16
2.4.1 Learning techniques	17
2.4.2 Traditional depth estimation methods	18
2.4.3 Common neural network architectures	18
3 Literature review	21
3.1 Datasets	21
3.1.1 Domain-specific	21

3.1.2	Multi-domain	22
3.2	Stereo matching	22
3.2.1	Common stereo matching structure	23
3.3	Comparison of neural networks	24
3.3.1	Evaluation metrics	25
3.3.2	Unsupervised learning	25
3.3.3	Non-end-to-end networks	27
3.3.4	End-to-end networks	31
3.4	Generative adversarial network	36
3.4.1	Noise-to-image translation based GANs	37
3.4.2	Image-to-image translation based GANs	41
3.5	Stereo image-pair synthesis	43
3.5.1	Image synthesis related to depth estimation	43
3.5.2	Image synthesis unrelated to disparity estimation	44
3.6	Discussion	45
3.7	Conclusion	48
4	Methodology	51
4.1	Research goal	51
4.2	High-level outline	51
4.3	Dataset	52
4.4	CycleGAN	53
4.4.1	Loss functions	54
4.4.2	Objective function	57
4.4.3	Model definition	58
4.4.4	Training details	60
4.5	Framework	61
4.6	Evaluation metrics	61
4.6.1	FID	62
4.6.2	RMSE	62
5	Experiments	63
5.1	Goal	63
5.2	Training parameters	63
5.3	Experiment 1: the influence of the identity loss function	64
5.3.1	Setup	64
5.3.2	Results	64
5.4	Experiment 2: the influence of network architecture	67
5.4.1	Setup	67

5.4.2	Results	68
5.5	Experiment 3: the influence of restricting the discriminator’s ability to learn	70
5.5.1	Setup	70
6	Results and discussion	73
6.1	Synthesizing stereo image-pairs	73
6.1.1	Model configuration and evaluation	73
6.1.2	Visual inspection of the synthesized views	74
6.1.3	Evaluation of the synthesized stereo-image pairs	76
6.1.4	Evaluation of rotation- and translation of objects in the scene .	78
6.2	Selecting the best model for depth estimation	78
6.3	Training the models	79
6.3.1	Tracking the adversarial loss	79
6.3.2	Tracking the FID score	81
6.4	Testing the models	83
6.4.1	Visual inspection of the estimated depth maps per model	83
6.5	Testing the final model on synthesized stereo image-pairs	86
7	Discussion	89
7.1	Discussion of the literature review	89
7.1.1	Concrete results	90
7.2	Discussion of the research results	90
8	Conclusions and recommendations	95
8.1	Conclusions	95
8.2	Recommendations and future work	96
	References	99

Introduction

This chapter will address the motivation and aims behind this dissertation, along with the formulation of the context, scope and research questions.

1.1 Motivation

Perceiving depth plays an important role in the perception of the spatial surroundings and the environment's three-dimensional structure. Humans and other mammals can do this naturally very well, due to their stereoscopic way of perceiving the world around them [1]. Their brain is exceptionally good at perceiving depth from the two single-view images from both eyes. Within the field of computer vision, similar techniques are applied to perceive depth; high-end technology is used to gather stereoscopic images, which can be used to estimate the depth of a certain point to the camera [2]. An example of a depth map can be seen in Figure 1.1.

Depth estimation refers to the extraction of three-dimensional information of a scene using two-dimensional information captured by a camera [3]. Before the emergence of advanced software-based solutions, depth estimation was done using sensors, such as: Time-of-Flight (ToF) or laser-based scanners (Li-DAR) [4]. Such solutions are called active methods, whereas software-based solutions are called passive methods. Active methods are more expensive to produce and deploy compared to passive methods, but the results active methods produce are more accurate and reliable [5]. For this reason, developing more accurate passive depth estimation methods is a popular and widely studied issue.

Traditional passive depth estimation methods rely on calculating the offset between an object visible in the left- and right view of a stereo image-pair, which is known as disparity estimation. The technological advancements of the past decade are marked by the emergence of neural networks, the first major step towards developing arti-



Figure 1.1: An example of a gray-scale depth map

ficial intelligence and human-like learning. Since the breakthrough performances of convolutional neural networks (CNNs), depth estimation transformed into a regression problem, that uses end-to-end trained deep networks. Eigen et al., pioneers in the field of depth estimation, developed the first CNN that was used for depth estimation. To improve their results, they connected it to a refinement network to reconstruct the fine details of an image [6]. Their method is called a monocular depth estimation network, which has been the standard ever since. Most, if not all, state-of-the-art depth estimation methods incorporate a neural network in their work and combine it with graphical models [7].

Generative adversarial networks (GANs), developed by Goodfellow et al. [8], have been gaining a lot of popularity over the last years due to their promising research results [9], [10], [11]. Recently, their possibilities are also explored within the field of depth estimation. They are highly applicable to complex learning tasks, such as, but not limited to: image-to-image translation [10], [12], [13], image style transfer [14] and generating high quality synthetic data, such as faces [15], [16].

The adversarial learning technique of deep neural networks is built upon a game theory where adversaries play a zero-sum game, a game where both players try to beat their opponent. A generative adversarial network consist of two neural networks: a generator and a discriminator model. The discriminator model learns to determine whether a sample is from the model distribution or the data distribution. The gener-

ative model learns to produce fake data that resembles the data distribution, in order to fool the discriminator in thinking that fake data belongs to the data distribution [8].

These techniques are rapidly gaining popularity in state-of-the-art applications and (consumer) products. Examples of recent applications that require accurate depth estimations are: autonomous vehicles, robotic navigation [17], augmented reality [18], [19] and mixed reality [20]. Neural networks and the underlying techniques of computer vision are becoming increasingly more important for depth estimation.

1.2 Problem statement

Worldwide, closed-circuit television cameras (CCTV) have been extensively implemented over the past decades, especially in public areas [21]. The video footage provided by these systems is generally used for real-time inspection, like monitoring public safety and traffic situations [22] or subsequently for analysis of the data [23]. Due to new technological possibilities, manual analysis and inspection of such data is gradually replaced by automated software. To do so, these systems need to be able to interpret the spatial surroundings and environment's three-dimensional structure. Depth information is crucial for building this understanding, but traditional depth estimation methods either rely on stereoscopic data or use expensive hardware to measure the actual distance.

This raises the problem that standard CCTV systems provide two-dimensional, single-view data that does not contain direct data about the scene depth. Demonstrating the need for software that is able to interpret scene depth from a single-view image, as solving this issue by implementing the required hardware everywhere is highly impractical and extremely expensive [24], [25], [26]. Within the field of computer vision, depth estimation models that are based on single-view images, is an ill-posed problem [27], [28]. Therefore state-of-the-art research explores various software-based solutions, in order to be able to successfully use the single-view images provided by these CCTV systems for depth estimation.

Although research and the application of deep neural networks and adversarial learning has shown remarkable progress, these developments are not sufficiently- accurate and reliable to be applied in real world applications. A clear example that describes the importance of accurate depth predictions are autonomous vehicles, because those have to rely on precise depth information in order to avoid colliding with their surroundings or worse. Another example is the purpose of the overarching project of this research, which is to perform automatic 2D-to-3D conversion. Thus, neural-network-

based and other depth estimation techniques have become essential in many modern day technologies and applications, but are not yet as good as these technologies and applications require them to be.

1.3 Context

Info Support is a company that is developing highly advanced technical software solutions for their clients. One of their clients, Paaspop, wants to perform crowd analysis to extract useful information from the flow of people at their festival terrain, in order to improve their infrastructure and logistics. They want to analyse the video recordings of their surveillance cameras, but in order to respect the new General Data Protection Regulation (GDPR) privacy regulations, this footage needs to be anonymized [29].

Ultimately, this project must deliver an end-to-end solution to reconstruct the camera recordings into an anonymized- and 3D representation of the data. Prior work on this project has been done by Info Support [21], which formed the foundation of this project. Based on the recommendations of their prior work, this end-to-end solution has been divided into six sub-components, where each sub-component is responsible for one part of the solution. Those six sub-components are as follows:

1. Object detection
2. Object tracking
3. **Depth estimation**
4. Camera self-calibration
5. 3D reconstruction
6. 3D animation

In the recommendations of this prior work, one of the main issues was that the state-of-the-art depth prediction module did not perform well enough and therefore, required further research. Hence, demonstrating the need to perform further research and describing the context of this research.

1.4 Scope

Considering all of these aspects, this research aims to explore the various neural network architectures and their applicability for single-view depth estimation. In the current model of Info Support [21] they make use of an externally developed, pre-trained depth

estimation model and it can be improved in terms of: accuracy, reliability and the ability to generalize better. One of the requirements of Info Support for this research is to explore the possibilities of GANs for depth estimation.



Figure 1.2: Flowchart of the core principles

This research also aims to design a solution that is able to convert any non-synthetic single-view image collection into an effective stereo training dataset, which in turn will be used to train a depth estimation network using an adversarial learning technique. The core principles of this process are visualized in a flowchart, as illustrated in Figure 1.2 and Figure 1.3.

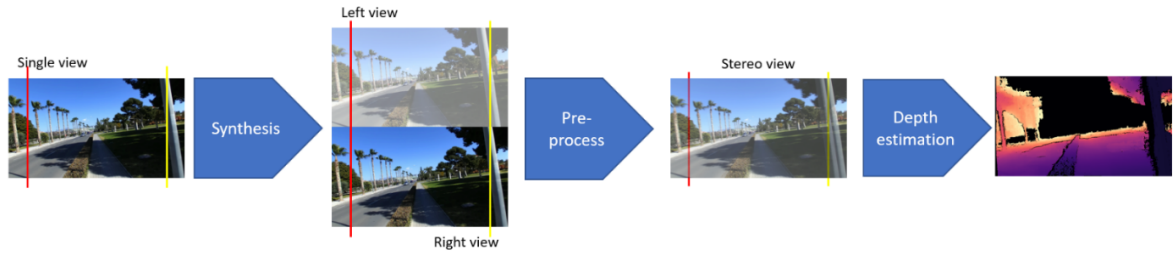


Figure 1.3: Flowchart of the core principles, visualized with images

1.4.1 Requirements

The implementation of the proposed method in Chapter 4 must fulfil the following criteria:

1. It must be able to synthesize a plausible stereo image-pair from a non-synthetic single-view image.
2. It must be able to be trained on stereo image-pairs, using adversarial learning.
3. It must be able to generate a plausible depth map for an unseen non-synthetic single-view image.
4. It should be able to generalize well on unseen data.

Under the observation that traditional depth estimation techniques require stereo image-pairs to make a depth estimation, the proposed method must be able to synthesize a plausible stereo image-pair. In turn, this synthesized stereo data will be used to train a neural network that perform depth estimation. The resulting network should be able to process an unseen non-synthetic single-view image, synthesize a stereo image and generate its corresponding depth map. According to recent research, GANs have shown very promising results [30], [31], [32], [33], [34], [35] and are therefore a key aspect in this research.

1.4.2 Challenges and limitations

Due to the nature of this research, i.e. synthesizing data instead and the interdependency of components within the proposed method, it is expected to have certain weaknesses:

1. Synthesizing a stereo image-pair will not produce data that is as accurate as the ground truth data captured by hardware that is specifically designed for that purpose. A slight error in this synthesized data will accumulate and propagate throughout the other components that rely on this data.
2. Existing state-of-the-art monocular depth estimation networks still have difficulties with making accurate estimations in crowded scenes, which have a lot of small details or contain ill-posed regions. This will affect the accuracy of the estimated depth map.
3. Neural networks are difficult to train and to fine-tune, because there is relatively little knowledge about how a neural network architecture will perform on a particular dataset in advance. Additionally, generative adversarial networks are inherently unstable and are therefore even harder to stabilize. In order to improve the chances of a satisfactory performance of the proposed method, all of the aspects have to be taken into account.

1.5 Research questions

Based upon the motivation, problem statement, context and scope, the following research question has been formulated and subsequently, five sub-questions;

RQ: *“How can the performance of current depth estimation networks be improved by training a generative adversarial network for depth estimation on stereo image-pairs, synthesized from a single-view images?”*

In order to be able to evaluate the performance of the proposed network, it is important to a priori gain a deeper understanding of the quality of the synthesized stereo data. The quality of the synthesized view of a stereo image-pair will be evaluated by comparing the similarity of the synthesized view to the original view. This can both be done at a per-pixel level or at a larger scale, by comparing the synthesized dataset to the ground truth dataset. Either way, the first sub-question is posed as follows:

SQ1: *"How similar are the synthesized views of the synthesized stereo image-pair to their corresponding ground truth view?"*

After performing some post-processing steps on the synthesized views of the new stereo image-pair, a depth estimation will be made upon the resulting stereo image. Essentially, there will be two networks that each learn a different task, but the method to evaluate their generated output can be the same. The estimated depth map can be evaluated in a similar way as the synthesized views are evaluated. Thus, the second sub-question is posed as follows:

SQ2: *"How similar are the generated depth maps estimated on (non-)synthesized stereo image-pairs to their corresponding the ground truth depth map"*

Generative adversarial networks (GANs) are known for their ability to generate new data with a similar distribution as the training set. The generator is trained indirectly, because it must learn to fool the discriminator, rather than minimizing the distance to a specific image.

In doing so, it could potentially learn to overcome the flaws that exist in the training dataset. Given this assumption, it might be possible to train a GAN on a dataset that contain (some) depth maps that are not entirely correct. This would allow for a much wider application in the future, because it would allow networks to be trained on data that doesn't need to be captured with expensive hardware, but existing depth estimation networks (software) instead. Therefore, the third sub-question is posed as follows:

SQ3: *"To what extent does the performance of the depth estimation depend on the level of plausibility of the 'ground truth' depth?"*

The third sub-question is measured through visual inspection, as absolute or numerical evaluation methods most likely can't measure or detect something like: a cloud in the sky that is estimated to be 10 meters away in the ground truth dataset, whereas

our model predicts it to be too far away to measure. Differences like that are easily spotted by the human eye, but are hard to detect for a machine.

In line with the third sub-question, another point of interest is evaluating how well the network can predict depth of ill-posed regions and around edges. The context of this research suggests that this is an important aspect, since ultimately: it should be able to predict depth at areas like a festival terrain - where there are a lot of moving small moving objects or people. Hence, the fourth sub-question is posed as follows:

SQ4: *“How well can the network accurately predict the depth of ill-posed regions and around edges?”*

At last, given the context of the research, the scenes at which the network has to be operational can differ quite a lot. Therefore, one of its requirements is that it should be able to generalize well on unseen data and so, the fifth and last sub-question is posed as:

SQ5: *“How reliable is the network and to what extent can the network generalize on unseen data?”*

Background information

In this chapter additional background information is provided, that is deemed necessary for understanding the concepts that are introduced in the remainder of this dissertation.

2.1 Depth estimation using epipolar geometry

Depth estimation or depth prediction refers to the techniques and algorithms that are used to obtain the spatial structure and 3D surroundings of an environment [36]. Meaning the technique used to calculate the distance of each point in the scene to the observer. Humans have learned to estimate depth naturally using both their eyes. Although their brain performs the computational part for depth perception, this can be done with two cameras as well. Using these two viewpoints, its corresponding depth map can be calculated using epipolar geometry, which is the geometry for stereo vision.

So, depth can be inferred from a pair of two-dimensional images. Bleyer [37] and Revuelta [36] describe in their research how a scene point can be reconstructed using a two camera setup, this will be explained in the remainder of Section 2.1 .

2.1.1 Two camera setup

Figure 2.1 illustrates a stereo vision setup, in which both cameras are correctly calibrated [38]. Both cameras, denoted as C and C_r , capture the same scene point P . The projection of point P is denoted as p_l and p_r on their corresponding image plane L and R . The projections of P onto both images planes are given by the intersection of the two lines $\overrightarrow{C_l P}$ $\overrightarrow{C_r P}$ with the corresponding image planes. As a consequence of this projection of P onto both image planes L and R , the z-coordinate of point P is lost in each image and cannot be recovered if there is only one camera available.

Loss of that specific z-coordinate is the main reason monocular depth estimation is not possible using only geometry. Scene point P lays at the intersection of the

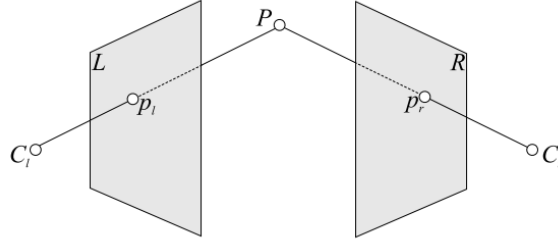


Figure 2.1: Two camera setup

rays $\overrightarrow{(C_l P_l)}$ and $\overrightarrow{(C_r P_r)}$, and can be reconstructed given that p_l and p_r are known. Unfortunately, these points are unknown a priori and this leads to a difficult problem in stereo vision reconstruction. Namely, the correspondence problem: given projection p_l of P onto image plane L , and so raising the question on where the projection p_r of P lies on image plane R .

2.1.2 Epipolar rectification

To simplify the process of reconstructing P , epipolar lines between the cameras C_l , C_r and scene point P can be drawn, that form a plane. This is illustrated in Figure 2.2.

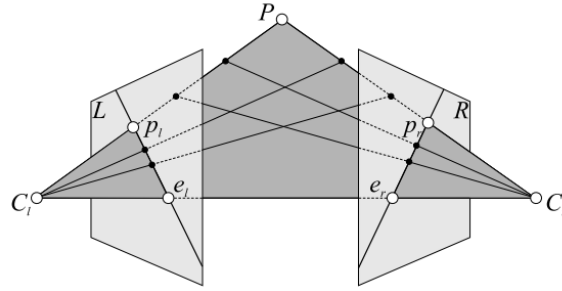


Figure 2.2: Epipolar geometry of a stereo vision system [37]

Any scene point projection to p_l lies on a line in its image plane L , perpendicular to the projection ray $\overrightarrow{C_l P_l}$. Consequently, all scene points must also exist on the line in the right view. This line is called the epipolar line of p_l and with respect to the images planes L and R , each epipolar line must cross its correspondence point e .

Epipolar rectification, as illustrated in Figure 2.3, reduces the complexity of solving the correspondence problem. Placing either image plane L (or R) parallel to scene point P , creates a configuration in which both image planes lie in the a single plane.

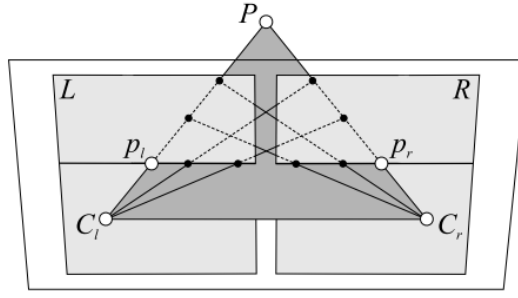


Figure 2.3: Epipolar lines after epipolar rectification [39]

After epipolar rectification, both epipolar lines move to infinity and align both with each other and with the line between p_l and p_r . Thus, the matching point of pixel in one image can be found on the same horizontal line in the right image. The horizontal offset is also known as disparity and can be calculated by the pixel difference between x_l and x_r .

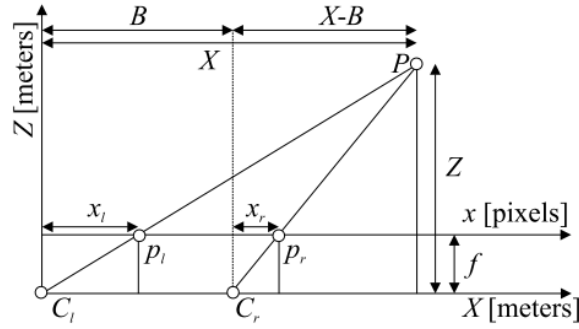


Figure 2.4: Depth reconstruction via triangulation

2.1.3 Depth from triangulation

As the correspondence problem is solved and the disparity is known, depth can be inferred through triangulation as shown in Figure 2.1.3. From similar triangles the equations $\frac{X}{Z} = \frac{x_l}{f}$ and $\frac{X-B}{Z} = \frac{x_r}{f}$ can be derived, from which consequently the equation for depth reconstruction can be derived, as described in Equation 2.1.

$$Z = \frac{B \times f}{x_l - x_r} = \frac{B \times f}{d} \quad (2.1)$$

2.2 Depth estimation using stereo view

In traditional depth estimation, using a two camera setup, the disparity and therefore the depth as well, can be calculated using triangulation and epipolar geometry. In real world applications it is often uncertain which pixel corresponds to which pixel in the other image of the stereo image-pair. Hence, it is important to compute a per-pixel similarity to determine which pixels show the same point in the 3D scene. These corresponding pixel-pairs are used to calculate the disparity. Calculating this per-pixel (patch) similarity is formulated as a multistage optimization problem, that includes the steps: (matching) cost aggregation, disparity optimization and some post processing steps [39], [40].

The purpose of cost aggregation is to find the best set of pixels on which to compute the matching cost for each patch of pixel under evaluation (i.e. the correspondence) [41]. Most traditional methods rely on fixed static support, which is typically a squared window or a single point. Cost aggregation using local algorithms based on a variable support (i.e. unconstrained shapes of pixels) yield a comparable result to global methods. These methods using variable support date back to the 70s to 90s [42], [43], only the last years these methods have found their ways into modern stereo networks. They have shown to be very effective in improving the performance of global algorithms, such as Belief Propagation (BP) [44], Dynamic Programming (DP) [45] and Scanline Optimization (SO) [46]. Hence, traditional cost aggregation methods often use rectangular shaped windows to calculate the per-pixel (patch) similarity. There are, however, alternative methods that aim to improve the accuracy and therefore, use unconstrained window sizes instead of a rectangular shaped window.

2.2.1 Cost aggregation based on rectangular windows

There are various categories of variable support methods that rely on a fixed set of rectangular window pair (i.e. pixel-patches), generally: varying the window size and/or offset, selecting more than one window and associating different weights to window points [41]. All of them rely on a fixed set of rectangular window pairs, $S(p, q)$, which is symmetrically defined on the stereo image-pair. The correspondence of a subset, (p, q) , is evaluated and used to determine a criterion $S_V(p, q)$, which varies at each correspondence under evaluation and since it does, it should adapt itself to the local characteristics of (p, q) . Thus, enabling better handling off depth borders and low-texture areas.

An algorithm for varying the window size and/or offset is proposed by D. Scharstein et al., which they called Shiftable Windows (SW) [4]. This algorithm is useful along

depth borders and aims at finding pixels that lay on the same depth plane, by minimizing the error function over $S(p, q)$. In their algorithm the set of windows is described by Equation 2.2.

$$S(p, q) = \{W_n(i, j, d) : i \in [x - n, x + n], j \in [y - n, y + n]\} \quad (2.2)$$

Another approach is to vary size of the window itself, which allows to deploy larger windows in low texture regions [47]. In their algorithm the set of windows is described by Equation 2.3.

$$S(p, q) = \{W_n(x, y, d) : n \in [N_{min}, N_{max}]\} \quad (2.3)$$

A more general approach is proposed by Veksler, his algorithm selects as support the window minimizing the cost over a set of windows [48]. In their algorithm the set of windows is described by Equation 2.4.

$$S(p, q) = \{W_n((x, y, d) \cup \{W_N(X \pm n, y \pm n, d)\} \quad (2.4)$$

Another method for cost aggregation based on rectangular windows is to select multiple windows, rather than one. In here, $S_V(p, q)$ is not a single window pair, but a subset of window pairs. Innocent et al. proposed a version of this method in which $S(p, q)$ is a subset of five squared windows [49]. In their algorithm the set of windows is described by Equation 2.5.

$$S(p, q) = W_N(x, y, d) \cup \{W_N(x \pm n, y \pm n, d)\} \quad (2.5)$$

2.2.2 Cost aggregation based on unconstrained windows

Cost aggregation based on unconstrained windows, builds upon the concept that $S_V(p, q)$ can be a subset of window pairs, rather than a single window pair, which allows supports to better adapt to local characteristics of each correspondence (p, q) . Boykov et al. [50] were the first to exploit this method, by classifying each correspondence as either plausible or implausible. Classification is based on the photo-metric relation between p_i and its correspondent I_q at the same disparity as (p, q) . For each pixel p , the best disparity is chosen from the largest set of connected plausible pixels, therefore allowing variable supports.

2.3 Matching cost

Matching cost is a measure that describes pixel dissimilarity for potentially corresponding image locations [51], most matching cost computations are done using: sum of absolute difference (SAD), sum of squared difference (SSD) and normalized cross-correlation (NCC). Jiao et al. proposed a stereo matching method that formulates a cost volume from a combined cost, where after performs cost-volume filtering to improve the accuracy of a disparity map [52]. More recently, Shaked and Wolf proposed two networks, a highway network that performs matching cost computations and, secondly, a global disparity network that predicts disparity confidence scores to further refine the disparity map [53].

Semi-global matching (SGM) is such an algorithm that estimates the disparity map for a rectified stereo image pair [51], [54]. The energy function E for solving SGM is described by Equation 2.6.

$$E(D) = \sum_{i=1} (C(x, d^x)) + \sum_{y \in N_X} P_1 T[|d^x - d^y| = 1] + \sum_{y \in N_X} P_2 T[|d^x - d^y| > 1] \quad (2.6)$$

Where $C(x, d^x)$ represents the matching cost of pixel $x = (u, v)$ of disparity d^x . Respectively, the first term of the sum represents the sum of matching costs of all pixels for the disparity map D . The second term penalizes pixels if they exist in a different surface with respect to its neighbouring pixels, whereas the third term penalizes pixels for discontinuities.

2.3.1 Semi-global matching

Traditionally, the SGM algorithm [51] repeatedly aggregates the matching cost in different directions. In a given image, the cost $C_r^A(p, d)$ of a location p at disparity d is recursively aggregated in the direction r , as shown in Equation 2.7.

$$C_r^A(p, d) = C(p, d) + \min \left\{ \begin{array}{l} C_r^A(p - r, d), \\ C_r^A(p - r, d - 1) + P_1, \\ C_r^A(p - r, d + 1) + P_1, \\ \min_i C_r^A(p - r, i) + P_2 \end{array} \right\} \quad (2.7)$$

This algorithm contains several issues that arise when it is used to train a deep end-to-end neural network. First of all, the SGM algorithm has many user-defined parameters, defined as (P_1, P_2) , which are difficult to tune and are therefore an unstable factor during the training of the neural network. Second, the cost aggregation and penalties in the SGM algorithm are fixed for all pixels, regions and images and thus, cannot adapt to different conditions. Third, a hard-minimum selection causes front-o-parallel surfaces in the depth predictions. These issues are solved by: (a) changing the user-defined parameters (P_1, P_2) to learnable weights (W_1, W_2, W_3, W_4) ; (b) changing the internal min to a max in order to maximize the probability at the ground truth labels and avoid negative values or zeros; and (c) take the weighted sum instead of the min to reduce the front-o-parallel surfaces in texture less regions. These adjustments are shown in Equation 2.8.

$$C_r^A(p, d) = C(p, d) + \sum \left\{ \begin{array}{l} w_0(p, r) \times C(p, d), \\ w_1(p, r) \times C_r^A(p - r, d), \\ w_2(p, r) \times C_r^A(p - r, d - 1), \\ w_3(p, r) \times C_r^A(p - r, d + 1), \\ w_4(p, r) \times \max_i C_r^A(p - r, i) \end{array} \right\}_{s.t.} \sum_{i=0,1,2,3,4} w_i(p, r) = 1 \quad (2.8)$$

The cost volume $C(p, d)$ with a size of $H \times W \times D_{max} \times F$ can be sliced in D_{max} slices at the third dimension for each candidate disparity d , where all of the slices repeat the aggregation step of Equation 2.8 with the shared weights ($w_{(0..4)}$). Instead of aggregating into sixteen directions, like the original SGM algorithm, it aggregates in four directions $r \in \{(0, 1), (0, -1), (1, 0), (-1, 0)\}$. The last aggregation step is obtained by selecting the maximum between the four directions, as shown in Equation 2.9.

$$C_r^A(p, d) = \max_r C_{rA}(p, d) \quad (2.9)$$

Selecting the maximum takes the best value for one direction, which makes sure that the aggregation is not distorted by other the other directions. The back propagation for w and $C(p, d)$ in the SGA layer can be done inversely as is shown in Equation.

2.3.2 Local guided aggregation

Thin structures and object edges will be refined using the local guided aggregation (LGA) layer. Usually, these finer details and edges are blurred, because stereo matching models apply down-sampling and up-sampling methods. The LGA layer learns to

refine the matching cost through several guided filters and aids in recovering these finer details. The local aggregation follows the cost filter definition and is shown in Equation 2.10.

$$C_r^A(p, d) = \sum \left\{ \begin{array}{l} \sum_{q \in N_p} \omega_0(p, q) \times C(q, d), \\ \sum_{q \in N_p} \omega_1(p, q) \times C(q, d-1), \\ \sum_{q \in N_p} \omega_2(p, q) \times C(q, d+1) \end{array} \right\} s.t. \sum_{q \in N_p} \omega_{0,1,2}(p, q) = 1 \quad (2.10)$$

Various slices of the cost volume (of totally D_{max} slices) share similar aggregation/filtering weights in the local guided aggregation (*LGA*). The traditional cost filter employs a $K \times K$ filter kernel to filter the cost volume in a $K \times K$ local region N_p . The *LGA* filter employs three $K \times K$ filters that are described as (ω_1 , ω_2 and ω_3) at each pixel location p for disparities d , $d-1$ and $d+1$ respectively. In short, it aggregates in a $K \times K \times 3$ weight matrix in a $K \times K$ local region for each pixel at location p .

2.3.3 Disparity refinement

In the majority of works aiming to improve the accuracy and performance of stereo networks much effort is put into optimizing the cost aggregation function, but far less in disparity refinement, nor cost measurement for that matter. Most traditional disparity refinement methods consist of three consecutive steps [53], [55], [56]: left-right consistency check for outlier pixel detection and interpolation tied to a confidence score, sub-pixel enhancement to enhance image resolution and median and bilateral filtering to smoothen the disparity, without blurring the edges. These disparity refinement steps are similar across different researches and well-documented by [52], [53], [55], [56].

2.4 Depth estimation using monocular depth estimation networks

In the following sections the rather traditional, state-of-the-art and novel methods for monocular depth estimation will be addressed. In Section 2.4.1 the learning technique that neural networks use will be addressed. Section 2.4.2 consists of a concise overview of the traditional methods, i.e. the conventional methods used in commercial applications, for depth estimation. At last, Section 2.4.3 presents the most popular monocular depth estimation networks, that will become a cheaper alternative to the conventional methods that involve expensive hardware.

2.4.1 Learning techniques

The development and deployment of deep learning models has taken a huge leap forward in the past decade and has proven to be a solution for many complex learning problems, amongst others; monocular depth estimation [3]. Researchers have developed numerous different approaches and models in an attempt to solve this problem, most of which rely on convolution neural networks. Although the issue is well-studied, it remains an ill-posed problem [27].

Currently, the majority of this research focuses on developing monocular depth estimation networks, which mostly rely on convolutional neural networks [57], [58]. Such neural networks are trained to learn to map an RGB image to its corresponding depth map. Their learning method can be categorized into the following methods:

1. Supervised learning, is a method that requires a very large amount of single images and their corresponding depth map for training.
2. Semi-supervised learning, is a method that requires a small amount of labelled data and a large amount of unlabelled data for training.
3. Self-supervised learning, is a method that requires a small amount of unlabelled data only.
4. Unsupervised learning, is a method that requires no labelled data at all.

All methods have their advantages and their disadvantages. Supervised learning techniques require enormous amounts of images with their corresponding high-quality depth map, which can be difficult to collect and hard to generalize for all use cases. Semi-supervised learning techniques are unable to correct their own bias and require external domain information. Self-supervised learning techniques suffer from generalization problems. Unsupervised learning provides no control over what it will learn and mainly focuses on clustering data, dimension reduction and finding undetected patterns.

For the last years, the learning techniques that are applied are mostly semi-supervised learning and self-supervised learning or a combination of both. There are a few researches out there that include either supervised or unsupervised learning. Nonetheless, the majority of recent researches have shown promising results with mainly the use of self-supervised learning techniques.

2.4.2 Traditional depth estimation methods

Passive depth estimation methods process the optical features that are captured in an image, from which depth information can be extracted using computational image processing. These methods can be categorized into two primary approaches: (1) multi-view depth estimation, like depth from a stereo camera setup and (2) monocular depth estimation [3]. Multi-view depth estimation requires high computational power and consumes a lot of energy. Monocular depth estimation methods require lower computational power and have less energy consumption. Despite that multi-view depth estimation is inherently more accurate than monocular depth estimation, monocular based depth estimation methods are a more economical and practical solution. Thus, research shifted its focus to developing better monocular depth estimation network.

Previous approaches of such methods relied mostly on: (1) operating on hand-crafted features, (2) based on probabilistic graphical or (3) adopting deep networks models [59]. Take for example Delage et al., who proposed a dynamic Bayesian framework that was intended to extract 3D information from indoor scenes [60]. Or take Saxena et al., who introduced a discriminatively trained multi-scale Markov Random Field that optimizes the fusion of local and global features [68]. Years later depth estimation was approached as a discrete continuous conditional random field problem [69].

2.4.3 Common neural network architectures

With the emergence of CNNs and following their breakthrough performances, depth estimation transformed into a regression problem, using end-to-end trained deep networks, with some recent efforts being made in combining these networks with various graphical models [7]. Eigen et al. were pioneers in the field of depth estimation, because they were the first to develop a CNN for monocular depth estimation, which was connected to a refinement network to reconstruct the fine details of an image [6].

At that time, Wang et al. [61] introduced a hierarchical convolutional neural network (CNN) that makes use of conditional random fields (CRF), or in short a CNN-CRF, which is a network that predicts depth and performs semantic segmentation from the same features. Shortly after, Liu et al. presented a CNN-CRF network which showed the huge potential of a regression term that predicts depth for a given pixel, using convolutional layers [62]. Xu et al. [59] built an encoder-decoder using a continuous CRF framework to learn multi-scale representations by recovering depth maps. Later on they added attention modules to act as a bottleneck in their encoder-decoder network.

Through the last years, many new architectures have been developed and researchers found certain architecture structures that performed well, while being adopted by other researchers, they became a standard. Examples of recent works that have incorporated those well-known architectures in their research are; Herman et al., who used a network based on the U-Net architecture [14], Godard, who incorporated a residual network (ResNet) [63] and Shu et al. that used a pre-trained VGG16 network in their research to perform image classification [64].

Although new architectures are continuously being developed, there are some network architectures that, of which some are older than others, are widely accepted and applied:

1. VGG [65]
2. Inception [66]
3. ResNet [67]
4. Xception [68]
5. ResNeXt [69]
6. DenseNet [70]

All of the above network architectures are an extension on the core structure of a convolutional neural network, which consists of an input layer, hidden layers and an output layer. In the middle layers, i.e. the hidden layers, the inputs and outputs are masked by the activation function and final convolution - hence the name. Typically, these hidden layers perform a multiplication on its input and the activation function that is commonly used is ReLU. Those layers are followed by other convolution layers, such as; a pooling layer, a fully connected layer and a normalization layer.

Up until today, the core architecture of a convolutional neural network is still used as the backbone of many neural network applications that perform monocular depth estimation network. Over time, the performance- and complexity of these networks grew and despite the fact that depth estimation models are equipped with new methods, it remains a solid architecture for these applications

Literature review

In this chapter the literature review is presented; it is divided into multiple sections, each addressing a different component of the proposed depth estimation solution that is presented in this dissertation.

3.1 Datasets

There is a scarce availability of datasets that closely match the expected scenario, i.e. crowded groups of people. A more detailed explanation of the context is given in Section 1.3. Most of the available datasets contain many images that: are taken while driving around in a vehicle; are captured at random public places; are synthetic non-realistic images; are images from indoor scenes or are images of random everyday scenes or objects. They may also vary from which perspective the images are captured. In Table 3.1 are some snapshots from various datasets.

3.1.1 Domain-specific

There is currently a variety of real-world stereo datasets available. Popular datasets like CityScapes [71], DrivingStereo [72], KITTI [73] and Middlebury [74] provide stereo image-pairs with the ground truth depth- or disparity maps. One of the limitations of all of the datasets is that they either contain a limited number of images, or domain-specific images. The CityScapes, DrivingStereo and KITTI datasets have been constructed mainly for self-driving vehicle use-cases, whereas the Middlebury dataset contains a small number of scenes that are all captured in a laboratory setting.

There are also synthetic datasets like MVS-SYNTH [75], which extract their images from realistic video games, such as GTA V. The advantage of those datasets over the other driving-oriented datasets is that they can contain larger number of samples. Despite their size, training a model on synthetic data can cause issues related to the










Holopix50k	UASOL	MVS-SYNTH
		
DrivingStereo	DIML-CVLAB	KITTI
		
CityScapes	Middlebury	DIODE
		

Table 3.1: Snapshots of various datasets for depth- or disparity estimation

domain difference, when the model is tested on real-world data.

3.1.2 Multi-domain

More recent datasets such as DIML/CVLAB RGB-D [76], DIODE [77], Holopix50k [?] and UASOL [78] provide a wider variety in indoor- and outdoor scenes, and stereo image-pairs that are taken from another point of view. Another difference that can be observed between the more recent datasets and the older datasets, is that the images in the more recent datasets contain fewer humans or human activity.

3.2 Stereo matching

Stereo matching, also known as disparity estimation, is the process of finding pixels in a stereoscopic image-pair that correspond to the same three-dimensional point in the scene and the computation of the horizontal distance in centimetres between these pixels, i.e. the disparity. This disparity is used to estimate the distance, whereas in monocular depth estimation the distance is directly estimated. The major difference between these methods is the number of views that are available.

Estimating depth from a single image is from a geometrical point of view impossible, it requires an stereo image-pair of which the per-pixel depth can be inferred using stereo matching. This is the per-pixel horizontal displacement in centimetres, i.e. disparity, between each corresponding pixel in both images [79]. Typically, this is framed as a matching problem, where current state-of-the-art performance is achieved by deep stereo networks [80], [81], [82], [83].

Currently, a fundamental issue for training deep stereo networks is a lack of sufficient, usable stereo training data. Good, usable and large datasets are hard to acquire, because the hardware required for gathering stereo images is expensive and it is rarely used in real-world applications [58], [84], [85]. The available datasets either contain a low quantity of images or a low variety of different images (scenes) within the datasets. Having a low variety of different scenes, has the effect that a trained model is not very well at handling unseen, different images. . The available datasets either contain a low quantity of images or a low variety of different images (scenes) within the datasets. Having a low variety of different scenes, has the effect that a trained model is not very well at handling unseen, different images.

Most state-of-the-art stereo networks are trained on large datasets of synthetic stereo data [83]. Mayer et al. created such a dataset, which is currently, one of the standard datasets for stereo disparity estimation and optical flow estimation [86]. Assuming the availability of sufficient amounts of stereo training data, deep stereo networks for depth estimation seem to be a very promising alternative to monocular networks. Pretraining a network on large amounts of noisily labelled data improves its performance on image classification [87], [88], [89], [90].

3.2.1 Common stereo matching structure

Traditional stereo matching consists of (some of) the following steps: cost aggregation, matching cost, disparity optimization/refinement, possibly some post-processing steps as well [41], [4]. Today, deep neural network architectures are used to compute similarity scores for clusters of pixels, with cost aggregation and disparity- computation or refinement methods. [86]. Common matching cost computations, i.e. the loss function computations, are done using amongst others: sum of absolute difference (SAD), sum of squared difference (SSD) and normalized cross-correlation (NCC).

Stereo matching networks that are able to achieve state-of-the-art accuracy, are limited by their matching- and cost aggregation function, which often leads to wrong predictions around the object edges, occluded regions and large- or texture less ar-

eas. Some methods aim to improve the matching- and cost aggregation functions of stereo networks [54], [91], [92]. Seki et al. used neural networks to predict the penalty-parameters, as shown in Equation 7, whereas Yang proposed to aggregate the cost using a minimum spanning tree. Traditional stereo networks [51], [81], [93] add additional local and (semi-)global constraints by penalizing changes of neighbouring disparities, in order to improve smoothness. Other state-of-the-art stereo networks treat disparity estimation as a regression problem, these models define their loss function directly on true disparities and their estimates [86].

Over time different approaches have been developed and improved, some consider disparity estimation as a regression problem, whereas others approach it as a multi-class classification issue. Some years ago, Eigen et al. [94] proposed a two-parts multi-scale deep network to estimate disparity. One network estimates the disparity on a global level and the other locally refines the estimations. Kendall et al. [95] proposed a novel deep learning architecture that tackles stereo depth estimation as a regression problem. Their model predicts a disparity map using three-dimensional convolutions with a disparity cost volume, that represent geometric features. Zhou et al. [96] proposed a network that consists of two CNN's that are able to predict a disparity map and the camera position, including directional information. To train their models, they used video material as input and the network selected a single frame as the target image on its own. Luo et al. [97] considered depth estimation as a multi-class classification problem. This proved to be a much more efficient alternative to the, at that time, state-of-the-art Siamese networks. Whereas those networks performed the necessary computations on an image-pair in about one minute, their multi-class classification network did so in under a second.

3.3 Comparison of neural networks

Although the common structure of a stereo matching network architecture is somewhat predetermined, one can make a clear distinction in their different forms [98]. These types of networks can be categorized into three main categories: (1) unsupervised stereo matching networks, (2) non-end-to-end stereo matching network and (3) end-to-end stereo matching networks. In the following sections each network category will be reviewed, discussed and current state-of-the-art networks will be compared, followed some concluding words. An overview of these network categories is provided in Table 3.2.

Framework	Methods	Advantages	Disadvantages
Unsupervised	Left-right consistency check	Little ground truth data	Poor performance
Non-end-to-end	MC-CNN, content-CNN	Simple; decent performance	High computational load; lack of context; pre-processing
End-to-end	PSMNet, GC-NET	Disparity image quality; easy to design	Very high computational load; long training time; ground truth data

Table 3.2: Overview of the three main categories of types of CNNs

3.3.1 Evaluation metrics

In the following sections each stereo matching framework is accompanied by a comparison table to measure the performance of the networks. Since the unsupervised networks are relatively old and differ from the newer frameworks, they are measured against the KITTI 2012 stereo dataset. For this comparison the following evaluation metrics are used; Absolute relative difference (Abs. Rel.); Square relative error (Sq. Rel.); Root mean square error (RMSE); The log of the root mean square error (RMSE log). For all these evaluation metrics holds, that the lower they are, the better the score. Additionally, the error is measured (δ) in percentages, which provides a comprehensive comparison among the methods: $\delta < 1.25\%$ represents the number of pixels that satisfy $\delta < 1.25$ and is calculated by taking the maximum of the predicted disparities and the ground truth disparities. For the error holds, higher values are better.

The remainder of the frameworks are newer and are compared against the KITTI 2015 stereo dataset. In those comparisons the percentage of erroneous pixels and average end-point errors are reported, for both non-occluded pixels and all pixels. The percentage of disparity outliers ($D1$) is calculated for the foreground and the background, hence the names $D1 - bg$ and $D1 - fg$. For this evaluation metric holds that a lower value is a better score.

3.3.2 Unsupervised learning

Most unsupervised stereo matching networks rely on an approach that minimizes the error between the warped frame and the target frame, which is learned by a CNN in an unsupervised way. Over the last few years, several methods have been proposed that are based on spatial transformation and view synthesis.

Flynn et al. [99] proposed a novel image synthesis network that generates a new view by selecting pixels from neighbouring images in image sequences, which they called DeepStereo. Xie et al. [100] also addressed the issue of view synthesis and proposed

a method that generates the right view of an input left image, i.e. the source image. They generated a binocular image-pair by minimizing a pixel-wise reconstruction loss. Hence, their method produces a per-pixel disparity distribution for every pixel and from that, the most likely disparity is selected to generate the pixel in the right view. Both of these synthesis networks created a foundation for unsupervised stereo matching networks. Luo et al. [97] used these works to reformulate the issue of monocular depth estimation into two subproblems; view synthesis and standard stereo matching. The main structure of their proposed network is based on Deep3D and DispNet [86], where Deep3D synthesizes a stereo image-pair and DispNet predicts the disparity using the stereo image-pair.

The first unsupervised network for single-view depth estimation that relies on an image reconstruction loss is proposed by Garg et al. [101]. Their network generates the inverse warped image of the target image, using the predicted depth to reconstruct the source image. This method proved itself to have a great performance compared to, at that time, state-of-the-art supervised networks. However, this monocular method is inaccurate when it comes down to reconstructing finer details. Godard et al. [102] extended this image reconstruction loss by adding bi-linear sampling in order to synthesize images. This feature was adopted by Ren et al. [103] and their proposed method resulted in a fully differentiable training loss, making it a solid foundation for end-to-end networks. These works together showed that image synthesis using a reconstruction loss on its own produced depth images of poor quality. This problem was addressed by proposing a network architecture with a novel training loss, which enforces left-right depth consistency. This consistency constraint greatly benefits the performance, even outperforming state-of-the-art supervised methods trained on ground truth data. These works showed the maturity of unsupervised stereo matching approaches that rely on minimizing the photo-metric warping error.

Some other unsupervised methods rely on estimating the optical flow using pose information. Zhou et al. [96] proposed an unsupervised method for both monocular depth and camera pose prediction. Their learning approach is based on view synthesis as the supervisory signal and it predicts monocular depth and ego-motion, i.e. the motion of something in 3D space. Unfortunately, their network performance was poor and not closely comparable to traditional stereo matching methods. Lastly, Yin et al. [104] proposed an unsupervised learning framework for monocular depth estimation, based on optical flow and, again, ego-motion. Their method is fairly unconventional, because they fed a monocular depth estimation network with stereo pair images and did not perform well either.

Previously discussed methods have tested against the KITTI 2012 stereo benchmark and their performances can be found in Table 3.3. These network performances are in line with the results of their own research. The stereo matching approaches that are based on minimizing the photo-metric warping error and use a left-right consistency constraint perform much better than all other approaches.

Methods	<i>Lower is better</i>				<i>Higher is better</i>			Runtime (s)
	Abs. rel	Sq. rel	RMSE	log(RMSE)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	
Luo et al.	0,094	0,626	4,252	0,180	0,891	0,965	0,984	-
Garg et al.	0,169	1,080	5,104	0,270	0,750	0,904	0,962	-
Godard et al.	0,068	0,835	4,392	0,150	0,942	0,978	0,989	0,035
Zhou et al.	0,208	1,768	6,856	0,280	0,678	0,885	0,957	-
Yin et al.	0,155	1,296	5,857	0,230	0,793	0,931	0,973	0,015

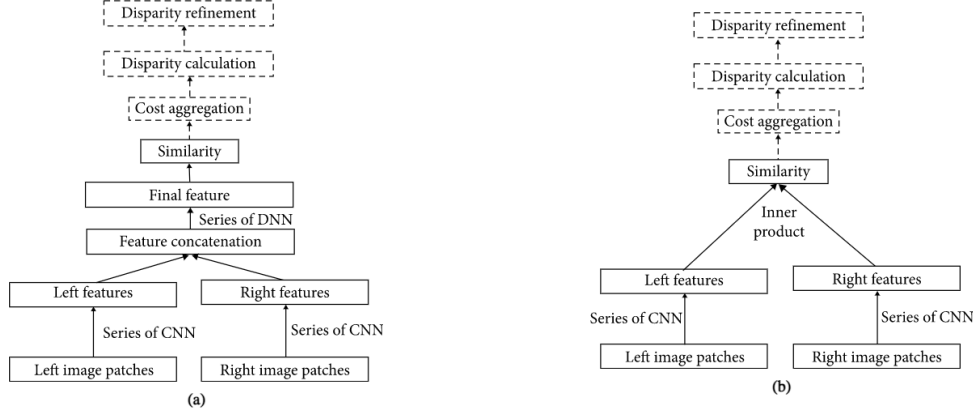
Table 3.3: Comparison of unsupervised stereo matching networks on the KITTI 2012 stereo benchmark

3.3.3 Non-end-to-end networks

Convolutional neural networks as a replacement for the legacy stereo matching pipeline components, were first introduced by Zbontar et al. [105]. They proposed a method, called MC-CNN, which performs matching cost computations using a neural network and refines its results using cross-based cost aggregation and semi-global matching. Using a deep Siamese network structure, consisting of several CNN and DNN layers, the similarity between two image patches of 9×9 pixels is measured. This similarity measure, i.e. the matching cost, is then refined using cross-based cost aggregation and semi-global matching. Lastly, they added a left-right consistency constraint to eliminate errors in the occluded areas. At that time, their method outperformed the existing state-of-the-art methods on the KITTI stereo dataset. This also showed that feature extraction is performed much more precisely by a CNN, than it is when the extracted features are handcrafted. Inspired by the success of MC-CNN many other top ranked methods [106], [107] adopted this method, either to compute matching cost or perform feature extraction.

Not long after this breakthrough, Yusof et al. [108] explored and proposed various different neural network models that make use of the similarity function, a function to measure the similarity between two images. By utilizing the CNN output features, the similarity function computes the similarity between both given image patches. Their goal was to find more challenging applications for this similarity function, using different types of neural networks. The conclusion of their exploratory research seems

obvious: the performance improves when (1) the model complexity increases and (2) the size of the training data increases.



(a) The basic Siamese network structure, which estimates the similarity between two image patches (b) The accelerated Siamese network structure, which employs a dot layer

Figure 3.1: Two Siamese network structures

Building upon the method to exploit a Siamese network structure, these new methods [107], [108] achieved state-of-the-art performances, but suffered from one major issue: time consumption. Unfortunately this issue could not be resolved due to the nature of their network architecture. As described by Luo et al. [97], their Siamese architecture is concatenated by a few fully connected layers (DNN) in order to compute the final score, which is illustrated in Figure 3.1-A. To illustrate this issue, assume an image size of $M \times N$ pixels, maximum disparity D and the inference time of the Siamese network T , i.e. the duration in seconds to make a prediction; the duration of the cost calculation step is described as $M \times N \times (D + 1) \times T$. Therefore, as the inference time T increases, the greater the computation time. Take for example MC-CNN [109], it took the network 67 seconds to process one stereo image-pair from the KITTI dataset.

To solve this problem, Chen et al. [110] proposed an alteration that fused multi-scale features in the matching cost calculations. They directly computed the similarity in Euclidean space by taking the dot product of the extracted feature vector, given as output from the CNN, which is illustrated in Figure 3.1-B. The accelerated Siamese network structure, which employs a dot layer. Directly computing the similarity vector from the CNN output, rather than concatenating the features and computing the similarity off that, decreased the inference time of the networks a hundredfold. Luo et al. [97] added an inner-product layer, specially to compute the similarity vector and also proposed a multi-label classification model over all possible disparities. This inner-product layer reduces the required computational power, while also enhancing

the matching performance by learning a probability distribution over all disparity values using a smooth target distribution.

In all of these approaches CNNs are deployed that have to learn to extract features from the given input images. After the cost volume is obtained from these extracted features, post-processing functions are deployed to further refine the results, some of which; cross-based cost aggregation, semi-global matching, left-right consistency checks, sub-pixel enhancement and (bilateral) filtering. The performances of these CNN-based methods on the KITTI stereo 2015 benchmark is shown in Table 3.4. An important note is that the OCV-SGBM method is included as a baseline. It has been provided by the OpenCV community and adopts handcrafted features, whereas the other methods adopt CNN-based features. Interesting to see is that all methods using CNN-based features are much more accurate in terms of their predictions, but require more computational resources and are therefore, slower.

	<i>Lower is better</i>								
	> 2 pixels (%)		> 3 pixels (%)		> 4 pixels (%)		> 5 pixels (%)		
Methods	Non-occ	All	Non-occ	All	Non-occ	All	Non-occ	All	Runtime (s)
Deep Embed	5, 05	6, 47	3, 10	4, 24	2, 32	3, 25	1, 92	2, 68	3, 00
MC-CNN	3, 90	5, 45	2, 43	3, 63	1, 90	2, 85	1, 64	2, 39	67, 0
Content-CNN	4, 98	6, 51	3, 07	4, 29	2, 39	3, 36	2, 03	2, 82	0, 70
OCV-SGBM	9, 47	10, 86	-	-	-	-	-	-	1, 10

Table 3.4: Comparison of CNN based, non-end-to-end stereo matching networks for cost calculation on the KITTI stereo 2015 benchmark

Many different researches focus on developing new and more complex networks to solve the pixel-patch matching issue, because the simple convolutional layers are limited to generate detailed representations. Yusof et al. [108] have already proven with their research that more complex networks potentially produce better results, hence enforcing the probability of these new researches producing new, better network designs. An example of such a new network design method of Park et al. [111], they proposed a method that tackles the pixel-patch matching problem. In their network they included a per-pixel pyramid pooling layer, that is able to cover a large area without losing resolution or fine details in the new image representation. Shaked et al. [53] approaches matching cost computations in a new manner, they designed a network architecture that is capable calculating each possible per-pixel disparity based on a multilevel weighted residual shortcut. A similarity between these methods is that all of them focus on the calculation of the cost and all of them achieve state-of-the-art performance in comparison to traditional algorithms.

Another approach to pixel-patch matching is based on the observation that disparity images are generally piecewise smooth, these networks include smoothness constraints in their learning process. Seki et al. [54] build their network around that observation. Their framework, SGM-Net, predicts SGM penalties for regularization. After grey scaling the input image, it separates the image in pixel-patches of 5×5 pixels, normalization its position as input and connects the prediction of the SGM penalties to it. This novel loss function of path and neighbour cost was introduced to enabled the usage of poorly annotated disparity maps, similar to real-world Li-DAR sensor data. Their network achieved state-of-the-art performance of the KITTI benchmark, but the entire process is time consuming and relatively complicated.

Knobelreiter et al. [112] proposed a hybrid model that consists of a CNN and CRF model, that optimizes the energy function. Through the use of unary-CNN and pairwise-CNN models it extracts detailed features and afterwards, calculates the unary cost and binary cost of the CRF. Together with this network, they proposed a theoretical profound method, based on the structure output support vector machine (SVM) and it is used to trained the end-to-end CNN-CRF model on a large-scale dataset. It is a similar method to traditional global methods based on graph cuts (GCs) and propagation. However, in contrast to traditional global methods, the features are unknown and the disparity is not calculated by iteratively minimizing the energy function, which is composed of extracted features and the disparity. In this method the feature can be calculated by the SVM and afterwards, function as a label to train the CNN-CRF network.

Other approaches focus on post-processing the disparity map. Gidaris et al. [113] replaced handcrafted disparity refinement functions with a three-staged network. This network is able to detect, replace and refine erroneous disparity predictions. When incorrect labels are detected, it replaces them with a renewed and refined label. Using this three-staged structure in their network, they achieved state-of-the-art results on the KITTI stereo 2015 benchmark. Regardless of its success, this network structure was considered inefficient in terms of computational resource usage, because the network computes disparities and later discards erroneous ones, only to renew them. If this is done correctly right away, it saved computational resources. Güney et al. [114] proposed a network that aims to reduce the amount of incorrect disparities predicted. Disparities of texture less and reflective surfaces are difficult to recover using traditional local regularization techniques. Their method was based on the observation that, generally, objects exhibit a regular structure or shape and are not arbitrarily shaped and changed their network accordingly. To be concise, they incorporated an arbitrary form of object detection in their network and achieved state-of-the-art performance using this method.

However, this method requires more computational resources and is therefore, slower. An overview of these networks is given in Table 3.5.

	<i>Lower is better</i>						
	All pixels			Non-occluded pixels			
Methods	D_1 -bg (%)	D_1 -fg (%)	D_1 -all (%)	D_1 -bg (%)	D_1 -fg (%)	D_1 -all (%)	Runtime (s)
Displets	3,00	5,56	3,43	2,73	4,95	3,09	265
SGM-Net	2,66	8,64	3,66	2,23	7,44	3,09	67
DRR	2,58	6,04	3,16	2,34	4,87	2,76	0,4
CNN + CRF	-	-	5,50	-	-	4,84	1,3

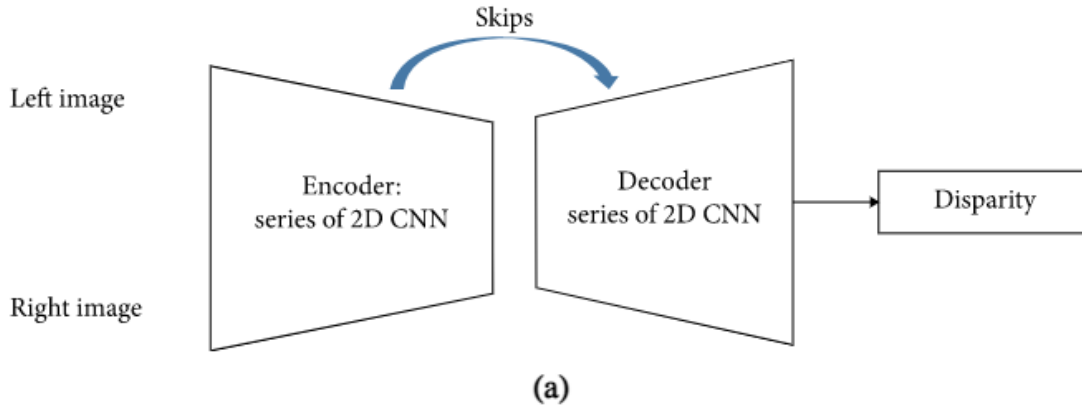
Table 3.5: Comparison of CNN-based, non-end-to-end stereo matching networks for cost calculation and post-processing on the KITTI stereo 2012 benchmark

All these non-end-to-end methods rely on some handcrafted regularization functions and some post-processing functions, in order to achieve state-of-the-art results, while they need increasingly more computational resources as they get better performance-wise. As demonstrated, all methods achieved great performance, but suffer from high runtimes as the network grows more complex. The DRR [113] network achieved the fastest runtime, because the network was designed to process the entire image at once, instead of performing calculations per-pixel.

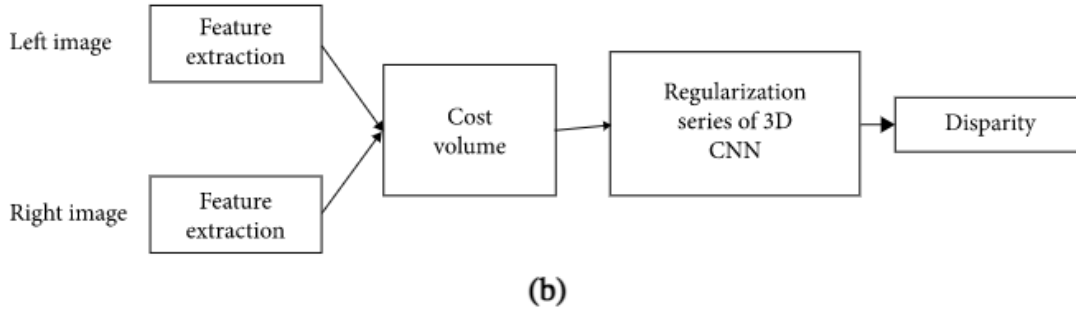
3.3.4 End-to-end networks

At last, the end-to-end networks, are all networks that seamlessly integrate all steps in the stereo matching pipeline for joint optimization [86]. These steps are; (matching) cost aggregation, disparity optimization and some post-processing steps [51], [93] and are explained in Section 2.4.3. Since the success of Mayer et al. [86], end-to-end stereo matching networks have become a popular solution for stereo matching algorithms and since then, many algorithms based on this have been proposed. Currently, the most popular end-to-end networks consist of 2D encoder-decoder structures, combined with refinement functions and regularization modules composed of 3D convolution layers. Figure 3.2 illustrates the structure of such a 2D encoder-decoder, which consists of multiple stacked 2D CNN networks with skip connections, i.e. a residual network.

Dosovitskiy et al. [115] were the first to develop such an end-to-end stereo matching network. They proposed a network to solve the optical flow estimation problem: FlowNet, a network that consists of CNNs following a basic 2D encoder-decoder structure. As a consequence of their success, many variations on this network emerged [86], [116]. Precise per-pixel localization is required for disparity estimation, but it also depends on finding the correspondences between two input images. Stereo matching and optical flow estimation have one critical difference: their search space. Stereo matching



(a) A 2D encoder-decoder structure with skip connections.



(b) Regularization module using 3D convolutional layers

Figure 3.2: Two popular end-to-end network structures

has a 1D search space, whereas the search space of flow estimation is 2D. Technically speaking, the search space of an optical flow estimation network can be downgraded in order to convert it to a stereo matching network [115], [117].

Inspired by the success of FlowNet, Mayer et al. [86] proposed DispNet, which is a network combined from an optical flow estimation network and a stereo matching network. The network uses a 1D correlation layer along the disparity line to calculate the cost and it uses an encoder-decoder structure with skip connections for disparity regression. Since this successful method was proposed, it became a popular end-to-end solution for disparity estimation, because it made the whole process much easier. One does only have to design a network that takes an image pair as input and the network directly predicts the disparity. These types of networks are much faster compared to non-end-to-end networks, because they directly consider the entire image, rather than processing it at a per-pixel or pixel-patch level. As illustrated in Table 3.6, DispNet is significantly faster in terms of runtime, but it still has difficulties finding the correct

correspondence at ill-posed areas. Using this successful method from DispNet, many different works emerged that focused on solving this problem.

	Lower is better								
	> 2 pixels (%)		> 3 pixels (%)		> 4 pixels (%)		> 5 pixels (%)		
Methods	Non-occ	All	Non-occ	All	Non-occ	All	Non-occ	All	Runtime (s)
PSMNet	2,44	3,01	1,49	1,89	1,12	1,42	0,90	1,14	0,41
SegStereo	2,66	3,19	1,68	2,03	1,25	1,52	1,00	1,21	0,60
iResNet	2,69	3,34	1,71	2,16	1,30	1,63	1,06	1,32	0,12
GC-Net	2,71	3,46	1,77	2,30	1,36	1,77	1,12	1,46	0,90
PDSNet	3,82	4,64	1,92	2,53	1,38	1,85	1,12	1,51	0,50
L-ResMatch	3,64	5,06	2,27	3,40	1,76	2,67	1,50	2,26	48,0
DispNet	7,38	8,11	4,11	4,65	2,77	3,20	2,05	2,39	0,06
EdgeStereo	2,32	2,88	1,46	1,83	1,07	1,34	0,83	1,04	0,32
GwcNet-gc	2,16	2,71	1,32	1,70	-	-	0,80	1,03	0,32

Table 3.6: Comparison of end-to-end stereo matching networks on the KITTI stereo 2012 benchmark

One of those works was proposed by Pang et al. [118], their method consisted of a two-stage architecture which they called cascade residual learning (CRL). The first stage makes initial predictions and the second stage continues with refining those predictions by generating residual signals across various scales. Both stages have a structure that is originating from DispNet [86], but the final disparity prediction is done by the summation of the output of both stages. They discover that the more complex a network structure is, the better its representation capability will be, which enforces the conclusions drawn by Yusof et al. [108], about the correlation between network complexity and performance. Although, by increasing the network complexity, the network speed will be reduced, as can be seen in Table 3.7. Their network, CRL, is about eight times slower than DispNet.

Another extension to DispNet was developed by Liang et al. [119], they proposed iResNet, a new, two-staged sub-network for disparity refinement. The two stages are combined for joint learning and is based on feature constancy. Notable is that this method handles all four steps in stereo matching and it is easy to optimize the network, due to the adoption of feature correlation and error reconstruction. It achieved state-of-the-art accuracy, whilst still remaining almost as fast as DispNet. Interesting to see is that this network slightly defies what Yusof et al. [108] concluded, because iResNet is less complex than CRL, but performs better and is faster. This can be explained by the fact that both of these networks aren't really one network, because they consist of two sub-networks that pass information on to each other.

	<i>Lower is better</i>						
	All pixels			Non-occluded pixels			
Methods	D_1 -bg (%)	D_1 -fg (%)	D_1 -all (%)	D_1 -bg (%)	D_1 -fg (%)	D_1 -all (%)	Runtime (s)
PSMNet	1, 86	4, 62	2, 32	1, 71	4, 31	2, 14	0, 41
SegStereo	1, 88	4, 07	2, 25	1, 76	3, 70	2, 08	0, 60
iResNet	2, 25	3, 40	2, 44	2, 07	2, 76	2, 19	0, 12
GC-Net	2, 21	6, 16	2, 87	2, 02	5, 58	2, 61	0, 90
PDSNet	2, 29	4, 05	2, 58	2, 09	3, 68	2, 36	0, 50
L-ResMatch	2, 72	6, 95	3, 42	2, 35	5, 74	2, 91	48, 0
EdgeStereo	1, 84	3, 30	2, 08	1, 69	2, 94	1, 89	0, 32
CRL	2, 48	3, 59	2, 67	2, 32	3, 12	2, 45	0, 47
LRCR	2, 55	5, 42	3, 03	2, 23	4, 19	2, 55	49, 2
DispNet	4, 32	4, 41	4, 34	4, 11	3, 72	4, 05	0, 06
GwcNet-gc	1, 74	3, 93	2, 11	1, 61	3, 49	1, 92	0, 32
SCV-Net	2, 22	4, 53	2, 61	2, 04	4, 28	2, 41	0, 36

Table 3.7: Comparison of end-to-end stereo matching networks on the KITTI stereo 2015 benchmark

Other methods attempt to solve the issues of DispNet by integrating additional information to the ill-posed regions. Yang et al. [120] proposed SegStereo, which integrates semantic features from segmentation into the image and they were the first to introduce semantic soft-max loss. The integration of additional semantic cues greatly benefits the disparity prediction and network performance. Xiao et al. [88], [121] proposed EdgeStereo, which is a combination of a disparity estimation network and an edge subnetwork, which is used to refine the disparity around object edges through regularization. Both SegStereo and EdgeStereo achieved state-of-the-art performance on the KITTI stereo benchmarks.

A fundamentally different method to DispNet and its variations, is to incorporate a regularization module based on 3D convolutions and is illustrated in Figure 8. Kendall et al. [122] were the first to use this in their proposed method, GC-Net, a network that constructs a 4D cost volume with concatenated features and uses a 3D convolution network to predict the disparity. These concatenated features are from the image-pair and are aligned along the disparity dimension. This is what greatly improves the network performance, hence achieving state-of-the-art performance. Chang et al. [123] adopted this network structure and proposed a pyramid stereo matching network (PSMNet), which essentially exploits global context information and consists of spatial pyramid pooling layers and stacked 3D CNN modules. These pooling layers extract multi-scale representations and the 3D CNN modules regularize the 4D cost volume to predict the disparity. PSMNet and its variations, have been one of the best stereo matching algorithms ever since.

Although networks like DispNet, PSMNet and all of their variations have demonstrated to perform very well in stereo matching, major issues are that they require large amounts of computational resources and are relatively slow. In an attempt to solve this problem and inspired by GC-Net, Lu et al. [124] proposed a sparse cost volume net (SVC-Net). This network generates a cost volume from features of an image pair; the batch size and disparity dimensions are merged to make a 4D cost volume. Designing a network like this greatly reduces the computational resources required, without making sacrifices performance-wise. To further reduce the memory usage, Tulyakov et al. [125] introduced the concept of a bottleneck matching module, which compresses the image descriptors into smaller matching representations.

So at this point, there are basically two conventional structures for stereo matching networks, but there are some methods that differ from this design. Yu et al. [126] proposed a method to perform cost aggregation on a learning-based method in order to improve the generation and selection of cost aggregation proposal from cost volumes. Slossberg et al. [127] provided a priori knowledge of inter-pixel interactions to regularize the cost volume, by added a densely connected conditional random field (CRF). Jie et al. [128] proposed a left-right comparative recurrent (LRCR) model to perform left-right consistency check alongside a disparity estimation network, which consists of stacked convolutional LSTMs. Although their disparity predictions are very good, using the LSTM structure requires a lot of computational resources, causing the network to be very slow. Kim et al. [20] designed a deep network architecture that estimates a stereo confidence. Lastly, Poggi et al. [129] attempted to improve the reliability of the disparity prediction, by introducing a confidence measurement network.

To summarize, current state-of-the-art networks mainly consist of either one of the two traditional architectures, that are illustrated in Figure 3.2, where (a) is the 2D encoder-decoder structure and (b) is the 3D regularization structure. These end-to-end stereo matching networks require a lot of computational resources, especially networks with the regularization structure. Although there are methods, like group-wise pixel correlation [80] and sparse cost volume techniques [124] to reduce their computational needs, they still require a lot of it. Moreover, these end-to-end stereo matching network require ground truth depth data for training, which can be very challenging and expensive to obtain.

3.4 Generative adversarial network

Since Goodfellow et al. [8] proposed generative adversarial networks it has been a much researched topic. Applying this gamification theory [130] to train neural networks has proven its near limitless potential, because both network become iteratively better over time. Only recently, these networks have made their breakthrough within the field of depth estimation; some of those adversarial trained neural networks outperform the state-of-the-art end-to-end networks based on 2D encoder-decoder structures or 3D regularization techniques. Currently, many works [24], [61], [131], [117], [123], [120], [124], [121], [118] heavily focus on developing monocular depth estimation networks or optical flow estimation networks, because those networks can be trained without the need for stereoscopic data. Other researches [13], [14], [132], [133] proposed an adversarial neural network trained on stereo image-pairs, of which most of these networks are trained using synthetic data.

Generative adversarial networks follow a strategy that is based upon a game theory approach, where two adversaries play a zero-sum game, i.e. two neural networks that represent the generator and discriminator. The goal of both players, i.e. the neural networks, is to beat their opponent. The generator model learns the distribution of the provided dataset, in order to generate fake data that fools the discriminator into classifying it as real data [8].

At the time of writing, no researches have been found that, using adversarial learning, train an end-to-end stereo matching network based on real-world stereo image-pairs. Although it can't be said with full certainty, most researchers mention a lack of sufficient real-world stereoscopic training data and therefore, aim to solve the depth estimation problem through monocular depth estimation. This unravels a gap in current research to depth estimation, or disparity estimation to be precise, through stereo matching. One research worth mentioning is that of Thakur et al. [134], they proposed a novel adversarial network that estimates scene flow, which is the simultaneous estimation of optical flow and disparity. However, their work still suffers from domain-shifting when it would be used in real-world applications, because their network is trained on artificial data and 3D models from the FlyingThings3D, Monkaa and Driving dataset [86]. Due to the scarcity of depth estimation networks based on generative adversarial networks that are trained on stereo data, monocular depth estimation networks that make use of adversarial learning are also considered for this literature review, because these works remain relevant to the topic.

Currently, there are two groups of generative adversarial networks; (1) the ones that

perform **noise-to-image translation** and (2) ones that perform **image-to-image translation** [3]. These GANs differ in the type of input that is being fed to the generator. The generator of noise-to-image GANs takes in random noise and from that, learns to generate the appropriate output, whereas the generator of image-to-image GANs takes in any input image and learns to transform, i.e. generate, that into the appropriate output. There are no significant differences between the discriminator in both network, since it only has to evaluate the authenticity of the generated- and real image.

3.4.1 Noise-to-image translation based GANs

The following section will dive deeper into the depth estimation networks that use noise-to-image translation as a basis in their depth estimation GAN. Many networks are tested on the KITTI dataset and evaluated using the same error- and accuracy metrics, their performances are given in Table 3.8. Some of these networks are not tested using the same evaluation metrics and are therefore left out of the comparison table, but make use of interesting techniques worth taking into consideration.

Methods	<i>Lower is better</i>				<i>Higher is better</i>		
	Abs. rel	Sq. rel	RMSE	log(RMSE)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Kumar et al.	0.120	0.747	4.756	0.187	0.849	0.955	0.848
Atapour Abarghoue et al.	0.423	9.343	9.002	0.122	-	-	-
Wu et al.	0.063	0.178	2.129	0.097	0.961	0.993	0.998
Tosi et al.	0.096	0.673	4.351	0.184	0.890	0.961	0.981
Chi et al.	-	-	-	-	0.935	0.982	0.998
Li et al.	0.136	1.064	5.176	0.289	0.830	0.942	0.976
Puscas et al.	0.141	1.283	5.677	0.237	0.815	0.930	0.968

Table 3.8: Comparison of GANs that perform monocular depth estimation

Kumar et al. [84] proposed a network to perform monocular reconstruction, i.e. a network that learns the camera pose and predicts a depth map accordingly. They extended current geometry-aware adversarial neural network architectures, that learn from photo consistency-based reconstruction loss functions over spatially and temporally adjacent images. Their proposed GAN learns from improved reconstruction models, that have flexible loss functions. Using generic semi-supervised and unsupervised datasets they made these loss functions less susceptible to adversarial examples. Their generator learns to synthesize neighbouring images to predict a depth map and their relative object pose, whilst the discriminator learns the distribution of monocular image in order to correctly distinguish the generator output from the real data. The

generator is assisted by a typical photo consistency based reconstruction loss function to enhance its training. Testing their method against state-of-the-art methods on the KITTI dataset showed that their method performed slightly better.

Atapour-Abarghouei et al. [14] proposed a network that attempts to overcome the issue that arise when using depth estimation networks on real-world data, whilst they have been trained on synthetic data. In their approach they make use of the advances made in image style transfer and adversarial learning, to predict a pixel-perfect depth map from a single real-world colour image, based on training over a large corpus of synthetic data. Experimental results indicate the effectiveness of their approach, but did not yet achieve state-of-the-art results.

Chen et al. [13] on the other hand, embrace the use of synthetic data, because it did prove itself very useful for training semantic segmentation models. In their work they proposed an approach to cross-domain semantic segmentation, using auxiliary geometric information. With this information, similar synthetic images and semantic labels can easily be generated using virtual 3D environments. They used this geometric information for two purposes; one is to reduce the issues in domain shifting, i.e. training a model on synthesized data and deploying it for real-world data. The other one is at the output level, to build a model that can perform semantic segmentation and depth estimation simultaneously. Their proposed network, Gio-ADA, is trained through adversarial learning on their synthetic data and validated by domain shifting from synthetic to synthetic and from synthetic to real-world data. Although their approach shows its effectiveness in reducing the issues that occur from domain shifting, their depth prediction does not achieve a state-of-the-art performance.

One after their last research, Atapour-Abarghouei et al. [135] proposed a depth filling network, which predicts missing pixels in depth maps based on the context and structure of the scene. Their model is based on a fully convolutional generative model and is trained using the available depth information for an RGB coloured image. As the ground truth data is not readily available, they use synthetic data instead with a separate model to predict where holes would appear in the depth map. The resulting synthetic data with holes in it, which makes it in a sense non-synthetic, is used to train the initial depth filling network.

Wu et al. [136] proposed the SC-GAN, a network monocular depth estimation network with end-to-end adversarial learning for image sequences, i.e. videos, without having to estimate the camera pose or change in camera pose over time. Their network exploits cross-frame relations, which is done by a spatial correspondence module that

makes use of Smolyak sparse grids for effective feature matching across neighbouring frames. Their generator learns to map the input frames to a estimated depth map, whilst their discriminator learns to distinguish this estimated depth map from the ground truth. Their experiments on the KITTI and CityScape datasets showed that their proposed network outperformed state-of-the-art methods on monocular video and was able to generate much more accurate depth maps.

Recent works proved that depth estimation can be done without the direct need for ground truth depth information. In this light, Tosi et al. [85] proposed a novel network, monoResMatch, which extracts features from a single view image to synthesize different views of that image, whereafter it performs stereo matching between these two cues. Although other researchers have been discussing this approach, monoResMatch is the first end-to-end network that performs stereo matching on extracted, synthesized data from single view images. Their experimental results with their network state-of-the-art performance for self-supervised monocular depth estimation networks.

Chi et al. [137] proposed a two-staged framework based on residual networks and conditional GANs to enhance the ResNet-based depth predictions. Their approach tackles the issue of high computational cost for accurate depth estimations. Their work mostly consists of improving the existing cGAN models to enhance the ResNet-based depth prediction. Their implementation of a two-staged deep regression model, however, deemed itself superior compared to the state-of-the-art depth estimation networks, when they tested it on publicly available datasets.

Li et al. [28] proposed a network that performs depth prediction for visual odometry from consecutive frames. They state that other self-supervised methods, which are based on the minimization of the photo-metric warping error or local structure from motion, are incapable of discriminating distortion in warped images, which renders the depth map vague. They tackle visual odometrical depth estimation as an image generation task through self-supervised, adversarial learning. Their generator learns to estimate the depth and poses to generate a warped target image, whilst the discriminator learns to evaluate the quality of the generated image with high-level structural perception, that should overcome the issue of pixel-wise loss in previous methods. Their experiments on the KITTI and CityScapes dataset achieved more accurate depth, while preserving finer details.

Puscas et al. [132] proposed a novel end-to-end unsupervised network for monocular depth estimation. Their network consisted of two GANs, which are coupled to a structured Conditional Random Field (CRF) model. Both GANs aim to generate distinct

and complementary disparity maps, but also aim at improving the quality of their generated disparity maps through adversarial learning. The purpose of the CRF coupling model is to fuse the generator and discriminator outputs from the dual GAN network, implicitly forcing mutual constraints on the two network branches and between the generator and discriminator. Adding the CRF coupling model facilitates network optimization to improve the generation of disparity maps. Experimenting prove that their network achieved superior performance in comparison to state-of-the-art methods.

Matias et al. [138] propose different approaches to deal with occlusion and ill-posed regions that occur while performing depth estimation. They state that these issues can be solved using the different scene perspective from stereo cameras, but this is often not possible. They propose a GAN for depth feature extraction, that estimates the depth within that extracted feature, in order to remove objects on disparity images. Their results show that incorporating depth features in the loss function and network architecture, greatly benefits the performance and precision of their network, making the output depth map closely resemble a ground-truth depth distribution.

Thakur et al. [134] proposed SceneFlowGAN, a novel network for scene flow estimation based on a two-ended loss function for both the generator and the discriminator. Scene flow estimation differs from optical flow estimation, because it estimates both the optical flow and disparity from input stereo images simultaneously. Thakur et al. were the pioneers, because their network was the first of its kind. Their network was based on a conditional adversarial network that requires stereo image-pair input.

Liu et al. [139] proposed a novel network for monocular depth estimation in endoscopy. Their network doesn't require a prior modelling of anatomy or shading and only requires little monocular endoscopic image sequences for supervised learning. It doesn't require manual labelling or patient computed tomography (CT) scans during the training and application phases. Cross-patient experiments using CT scans as ground-truth data showed that their method achieved a sub-millimetre mean residual error, which is especially important in endoscopy. In a comparison study, they showed that their proposed method outperforms recent state-of-the-art self-supervised methods for depth prediction from endoscopic data.

Khan et al. [3] performed a comprehensive review of existing deep learning-based monocular depth estimation methods. In their review they make a distinction in traditional methods, either: active or passive. Both of these methods rely on the assumption of having observations in the scene, that exist in either the space or time domain (e.g. stereo or multi-view or structure from motion) [4], [140]. Active methods base their

predictions on interactions made with objects and the environments, where passive methods extract information from optical features in the image for their predictions. Between passive methods another distinction can be made, those that infer depth from either a monocular view or a stereo view. They note that multi-view based methods have serious limitations, because they require a lot of computational resources and have a high energy consumption. An observation they made is that current research is mostly focused on developing deep-learning based monocular depth estimation networks, that require less computational resources and have a lower energy consumption. In their review they evaluate the performance of various supervised, semi-supervised and self-supervised methods using quantitative evaluation metrics, such as: absolute relative difference (Abs Rel), root mean square error (RMSE), root mean square error in log space (RMSE log) and square relative error (Sq Rel). Following from their results, they mentioned the networks that performed best at certain metrics and concluded that state-of-the-art supervised methods are generally low-cost networks that are fast, with sufficient accuracy, whereas self-supervised networks are high-cost networks that are slow, but achieve remarkable accuracy. Their recommendations for future work mostly regard reducing the cost of training these networks and improving the accuracy in general.

3.4.2 Image-to-image translation based GANs

As opposed to noise-to-image translation, where the goal is to learn to generate the distribution of the target domain, the goal of image-to-image translation is to learn the mapping between a source domain and a target domain. The development of GANs for cross-domain image-to-image translation has made a lot of progress the last few years. The application of image-to-image translation network is widely applicable and GANs performing this task have shown remarkable performance.

Isola et al. [10] developed Pix2Pix, which is a conditional generative adversarial network (cGAN) that formed the basis for many state-of-the-art image-to-image translation GANs. In their research they investigated how to use conditional adversarial networks as a general-purpose solution for image-to-image translation tasks. The network learns not only the mapping between the source distribution and the target distribution, but also learns from a loss function to improve this mapping. This generic method solved a lot of the image-to-image translation issues. Their generic approach made it possible to use the same method to problems that traditionally required very different loss- and mapping formulations.

Zhu et al. [141] built on top of the fundamentals of the cGAN and combines it with

a variational auto-encoder (VAE). In their work, called BicycleGAN, they enforce the connection between latent encoding and the network output in both directions. This allows for a multi-to-one mapping. Emami et al. [142] introduced SPA-GAN, which uses a novel spatial attention mechanism for image-to-image translation. This spatial attention mechanism computes to where the discriminator puts its attention and passes that information to the generator, so that the generator is able to focus and improve on the discriminative regions between the source- and target domains.

Li et al. [143] investigated the connections between the real- and latent space, in order to develop a model that is able to more accurately capture and produce the full distribution of the real dataset. In their model, CE-GAN, they have combined an auto-encoder with an existing GAN. The auto-encoder, encodes the ground truth data and generated data into a latent space, before it passed to the discriminator. This technique allowed the discriminator to give better feedback to the generator, which in turn forces the generator to produce images that better fit the full distribution of the dataset.

Zhu et al. [144] proposed CycleGAN, a network designed for image-to-image translation tasks without the need for paired images. They introduced two new cycle-consistency losses: the forward- and backward cycle-consistency loss. The network uses both losses to learn the mapping from source domain X to the target domain Y . They assume that there exists an underlying correlation between the source and target domain, so that they can combine their new cycle-consistency loss with the existing adversarial loss. The images from both domains however, still needs to be labelled. Yi et al. [104] built on top of the CycleGAN framework and proposed DualGAN, a similar network that can be trained using two sets of unlabeled images.

Liu et al. [145] investigated the image-to-image translation task from a probabilistic modelling perspective and poses that a key challenge is to learn a joint distribution of images from different domains. In their research they propose UNIT, an unsupervised GAN that is based on the assumption that images from different domains can be mapped to the same latent representation in a shared-latent space. Their model has some limitations: their training model is uni-modal to the Gaussian latent space assumption and therefore, causes training to be highly unstable. Huang et al. [70] built on top of the UNIT framework and proposed MUNIT, which is a multi-modal unsupervised image-to-image translation GAN that attempts to solve the issues of its predecessor.

3.5 Stereo image-pair synthesis

Training a stereo depth estimation network requires enormous amounts of stereo image-pairs and currently, there is insufficient real-world stereo data available to train them in a similar fashion as a monocular depth estimation network. Stereo networks have the potential to perform way better compared to monocular networks, given sufficient training data. Inspired by recent progress in monocular depth estimation, researches shifted its focus towards image synthesis in order to create stereo image-pairs. An increase in availability of training data should allow stereo matching networks to be trained on larger corpus of (synthetic) data and therefore, improve their performance in terms of accuracy and ability to generalize.

The following section will review various image synthesis methods, although their method differs, most of them [101], [63], [146], [142], [147], [148], [79], [149] are designed to create stereoscopic data used to train an optical flow estimation network or stereo matching network. Other researches tackle a different issue, such as the smoothing of video material [95], modelling of a 3D scene from video [150] or synthesizing a multi-view of an object or scene [151]. Needless to say, regardless of the purpose these networks were designed for, any stereo image-pair can be used to train a stereo matching network, hence demonstrating the relevance of these works.

3.5.1 Image synthesis related to depth estimation

Garg et al. [101] used a predicted depth map based on the left-eye view to compute the disparity map. After warping the right-eye view as the new left-eye view, they calculated the error from the two left-eye views, which was used to train their unsupervised CNN. Godard et al. [102] continued developing this into a more stable method. Instead of only warping one-eye view as the other, they warped both eyes simultaneously. Thus, the left-eye view becomes the new right-eye view and vice versa. The left-eye view was used to compute the disparity map and afterwards they warped the left- and right-eye views, respectively, to obtain new left- and right eye views. Essentially they simulated a rectified row of people looking at the same object and using the view of a single person's left eye they warped it to become the other people's eyes.

In contrast, Xie et al. [100] proposed a method that is inherently different from traditional methods, because their method directly predicts several disparity channels. It is trained by converting the conventional warping equation to a differentiable form. One could argue it inherently differs from traditional methods, but it still remains an conventional warping methods, which considers per-pixel disparity.

As most conventional image warping methods only consider the translation of the pixels to synthesize a new image-pair, Lo et al. [149] proposed a novel method to generate stereo image-pair from a single image, that besides pixel translation, also considers the rotation of individual objects by incorporating semantic segmentation in their model. They modified the existing architecture of the appearance flow network (AFN) to better suit their model needs and added a in-painting function, i.e. image refinement function to enhance the image quality and improve the reconstruction of it. Due to the inclusion of rotational information, the resulting network output seemed more stereoscopic and was able to achieve state-of-the-art results.

Recent research incorporated existing monocular depth estimation methods in order to synthesize stereo image-pairs. Watson et al. [79] proposed a novel method for generating accurate stereo image-pairs, which was a successful solution to the stereo image-pair synthesis problem. Their method is based on inverse triangulation, using existing monocular depth estimation. Using existing, pre-trained monocular networks they estimate a plausible depth map, transform it into a disparity map and perform disparity refinement methods to enhance the image quality and gradient. Their assumption is that it is unnecessary to have such a high reliance on ground truth depths or even corresponding stereo pairs to properly train a stereo depth estimation network. Based on their assumptions they developed an end-to-end pipeline that synthesizes an accurate stereo image-pair from a single image and using these stereo image-pairs they trained existing stereo matching networks, achieving state of the art accuracy.

3.5.2 Image synthesis unrelated to disparity estimation

Zhao et al. [151] proposed a method that generates different views of an object. The global appearance details, like colour and shape, are generated using variational inference and the completion of details is done by a GAN. Their method is based on the existing variational auto-encoder generative adversarial network (VAE-GAN) architecture, which they called Vari-GAN. Their network, however, produced low-quality results and was only applicable for one type of object.

Flynn et al. [99] proposed a method that synthesizes images for complementing frames. Given a sequence of multiple, consecutive images of a different view, their network predicts new views that do not appear in the given sequence. Their network was not given a large enough sequences and therefore performances was relatively poor, yet their network was able to smoothen the video some more. Another image sequence based method was proposed by Hedman et al. [150] proposed a method that recon-

structs an entire three-dimensional scene. Their model produces high-quality images, containing colour and depth information, albeit that their model requires its input to be a image sequence.

Kulkarni et al. [146] encoded geometric parameters, like pose and light, to the view of an object and trained their network with these geometric parameters. After decoding, a new view holding information about these parameters could be reproduced. Building upon the concept of encoding geometric parameters into a view, Tatarchenko et al. [142] added rotational information to the encoding process. Thereafter, this network was able to generate pixels of the new rotated view. Zhou et al. [98] approached this problem entirely different, he proposed an appearance flow network (AFN), which generates the appearance flow instead of pixels. By sampling the original view with the flow, they synthesized a new high-quality view. Unfortunately, this method has one major shortcoming. The view is sampled from the original view and therefore, influenced by the correlation between these two views: the larger the rotation angle will be, the fewer pixels the new view will share with the original view and inherently, be of low quality.

Based on these developments in AFNs [98] and GANs [8], Park et al. [148] proposed a new view synthesis network, in an attempt to overcome the shortcomings of the existing AFNs. They multiplied the output of an appearance flow network with a visibility map and concluded that this result could be used for image synthesis. Their findings were correct and using this new method, they were able to overcome the shortcomings of existing AFNs.

3.6 Discussion

In this literature review an overview of the different state-of-the-art stereo matching networks have been presented, which is presented in a structured manner over time. Starting with unsupervised stereo matching networks, followed by non-end-to-end stereo matching networks and lastly, end-to-end stereo matching networks. Afterwards methods based on adversarial learning are reviewed and this is topped off with a review on different image synthesis methods.

Stereo matching

The reviewed stereo matching networks are divided into three different categories (unsupervised, non-end-to-end and end-to-end), a major similarity between these network types is that all of them focus on developing and improving the core neural network architecture which performs the actual stereo matching. In general, each category

builds upon the work and progress of the previous one. After presenting the state-of-the-art stereo matching networks one category at a time, a new aspect comes into play: adversarial learning, which is a new technique to iteratively improve those neural networks through competition with another neural network. At last, different stereo image-pair synthesis methods are reviewed. Its contribution to this research is that different methods for stereo data synthesis are reviewed, which will allow us to explore different methods for the creation of stereo training data for a stereo matching network.

Unsupervised stereo matching networks were the first stereo matching networks that stepped away from traditional methods by incorporating a CNN in their architecture. Most of these methods rely on minimizing the photo-metric warping error between both images from the stereo image-pair, which is learned by a CNN in an unsupervised way. In general, these network generate predict a per-pixel disparity and perform some basic refinement operations, like incorporating a reconstruction loss to enhance the image quality. The networks that, essentially, set the standard for unsupervised networks are DeepStereo [100] and DispNet [86]. The advantage of these types of networks is that they do not require ground-truth data, but a major disadvantage is their poor performance.

Non-end-to-end stereo matching networks builds upon the flaws of unsupervised networks. Two general trends can be observed in these non-end-to-end networks; (1) these networks have to learn the similarity between pixel-patches, rather than learning a per-pixel disparity, which was done using a Siamese network structure and (2) using these Siamese network structure, these networks started to perform feature extraction from those pixel-patches. Initially, many of the extracted features were handcrafted, but over time these networks become more in control of which features had to be extracted. In general, these non-end-to-end networks became increasingly more complex and a lot of the manual work was taken over by these networks, but there were still some post-processing functions required to ensure proper image quality. This caused in increase in network performance, but on the contrary, it also increased the required computational resources to train these networks.

The end-to-end stereo matching networks mainly incorporate functionality that aims to reduce the manual labour or input, to create space for the network to make its own decisions. In end-to-end networks we can clearly observe two main network structures, which either exists of (1) a 2D encoder-decoder structure or (2) a 3D regularization structure. Both of these networks architectures require vast amounts of computational resources and take a long time to train, especially those based on a 3D regularization structure. Many different methods are proposed to reduce the com-

putational cost, like group-wise pixel correlation [80] and sparse cost volume techniques [124]. Despite this high computational cost, their performance is significantly better than the previously discussed network types. In contrast to the other network types, a user can simply provide a stereo image-pair as input and without the need of other actions, the network will output an accurate disparity map. This, coupled with their high performance, makes them the most promising type of network to use for stereo matching.

Generative adversarial network

Training a network against an adversary is a relatively new method and very few methods could be found that attempt to solve the issue of stereo matching through training a neural network using an adversary. Most of these works focus on solving the monocular depth estimation issue, rather than training a stereo matching network. Given the problem statement of these researches, the most probable cause that adversarial stereo matching is an ill-researched problem, is due to a serious lack of sufficient training data.

Although most techniques focus on monocular depth estimation, a notable method was proposed by Tosi et al. [85], they synthesized multiple views of a single image to perform stereo matching on this. It is arguably what type of network this exactly is, because it predicts disparity from synthesized images that originate from a single image, but regardless, their network performance was one of the best. The other technique that stands out from the others was proposed by Wu et al. [136], their network exploits cross-frame relations and essentially performs some form of stereo matching between adjacent frames in a video. Using this technique their network performed even better than to the one proposed by Tosi et al. One major drawback is that this technique relies on video material as input, whereas the other network can function make its prediction based on a single image. Another method worthwhile mentioning was proposed by Thakur et al. [134], an adversarial network that simultaneously estimates optical flow and disparity. It is one of the few works that has a high resemblance to this research, which is to train a GAN based on stereo image-pairs in order to predict the disparity. At last, a work published earlier this year, Khan et al. [3] performed an extensive review of state-of-the-art monocular depth estimation methods. This research provides a clear method to compare and evaluate different methods to one another, which – in short – boils down to; (a) comparing the degree of supervision; (b) comparing the accuracy and depth range; (c) comparing the computation time and memory footprint.

Stereo image-pair synthesis

At last, image synthesis, a method that allows for the creation of stereo training data. These synthesis methods are divided into two categories; (1) those that focus on the creation of stereo image-pairs explicitly for training a stereo matching and (2) those who don't and simply perform image synthesis, regardless of their purpose. Regardless of their research goal, an accurately synthesized stereo image-pair should serve as usable stereo training data. A general trend that can be observed is that many works focus on using and improving conventional warping methods, such as pixel transformation in the space domain. More recent methods rely on (1) creating different views of an object in the scene, (2) encoding geometric parameters into their data or (3) the output of an appearance flow network and visibility map. The performance of multi-view based methods is poor, whilst encoding geometric parameters significantly improved the performance, whereas the AFN based methods perform best. Unfortunately, these AFN based methods require video material as input. The most recent methods from earlier this year, proved itself very successful in synthesizing an accurate stereo image-pair; Lo et al. [149] encoded rotational information about an object into the image, which significantly increased the performance of conventional methods, whereas Watson et al. [79] proposed a novel method that synthesizes the other image using triangulation via an plausible depth map, estimated by an existing monocular depth estimation network.

3.7 Conclusion

When drawing conclusion it is important to take the context of this research and the research goal itself into consideration, because one method may not necessarily be better or worse than another, given that their context or purpose differs. The goal of this research is to synthesize stereo image-pairs from a single image and use that synthesized stereo image-pairs to perform stereo matching, i.e. disparity estimation, using a generative adversarial network. The context of this research is that the proposed network will, eventually, operate at a crowded festival terrain with a lot of people; the data comes from the surveillance cameras and its final purpose is to anonymously model the flow of people at this terrain, in order to be analysed afterwards and it is not for real-time surveillance. Furthermore, these surveillance cameras are prone to malfunction from time to time and video material can be incomplete. Hence, there is no direct restriction to the model, that requires it to operate at real-time speeds, but it demands a high accuracy, since there will be a lot of small details in the input image, given that festivals are often crowded. It is also important to avoid relying on video material, because the available data can be incomplete. Therefore, higher

computational cost methods, i.e. slower methods, that do not rely on image sequences should be considered. Despite their lack of speed, these network make up for it in terms of a higher accuracy.

Given the context, research goal and motivated by the literature research, an end-to-end solution fits best, and to achieve the highest accuracy possible it will make use of a deep convolution generative adversarial network architecture (DCGAN). Since it can effectively produce high-quality generator models, compared to other GAN models, such as: cGAN, InfoGAN, StackGAN and AC-GAN. In this architecture, deep convolutional neural networks are used both for the generator and the discriminator architectures. Using this method will result in a more stable training of the generator.

Stereo matching networks require a large amount of stereo training data and currently most, if not all, networks rely on (partially) synthetic training data. A major issue with this approach is that these networks suffer from domain shift, reducing their performance in terms of accuracy and ability to generalize well. To solve for this absence of sufficient real-world training data, a stereo image-pair synthesis method should be developed, which can – ideally – transform every real-world single view images into stereo image-pairs used for training the generator.

Methodology

In this chapter the methodological design of this research will be presented, which specifies the practical implementation of this dissertation.

4.1 Research goal

The main goal of this research is to develop a network that accurately estimates a per-pixel depth for a synthesized stereo image-pair, but while doing so to explore the potential of generative adversarial networks. The proposed model for this network consists of two main parts, which are essentially two separate neural networks. The first network will synthesize a stereo image-pair from that single-view image, after which the second network uses this synthesized stereo image-pair to estimate a depth map. This process is visualized in a flowchart, illustrated in figure 4.1.

In the remainder of this document, these two networks will be referenced as: the stereo synthesis network and the depth estimation network.

4.2 High-level outline

There are two tasks at hand, one is to develop a generative adversarial network that is able to estimate a depth map from a synthesized stereo image-pair and the other



Figure 4.1: Pipeline flowchart, where the yellow blocks represent a piece of data and the blue blocks represent the networks

is to synthesize that stereo image-pair from a single-view image. Both tasks are an image-to-image translation task, because: based on a given input image, a different output image must be generated.

The single-view image can be considered to be either the left-view or the right-view of a stereo image-pair. For simplicity, consider it to be the left-view image. The image translation task that can be performed, is that this left-view image can be translated into the right-view image (or vice versa). The result is a synthesized right-view image, which can be combined with the original left-view image into a stereo image-pair. These left- and right views can be combined into a single stereo image, by placing the images on top of each other and adjusting their transparency to 50%. Using the resulting stereo image, the second image-to-image translation task can be performed, which transforms the stereo image into its corresponding depth map.

One can find various networks in the literature, that can perform such image-to-image translation tasks. However, for this research CycleGAN [?] has been selected. Due to its capability to transform an image from one style domain to another, while keeping the structure of the image intact.

4.3 Dataset

The final dataset that is used to train the CycleGAN is a subset of the *UASOL* dataset [78]. This dataset contains a folder *TestSet* folder that holds 676 unique stereo image-pairs of 2208×1242 pixels in *.png* format, as well as their corresponding depth maps. Due to the heavy computational requirements training our network costs, this subset is used. This subset will be split into a 80% train-, 10% validation- and 10% test-set.



Figure 4.2: ZED 2 Stereo camera

The depth maps that correspond to the stereo image-pairs have been acquired with the Zed 2 stereo camera, Figure 4.2. It can measure the depth to up to 20 meters away and it is calibrated according the settings shown in Table 4.1. Additionally, every stereo image-pair comes with a second depth map, that is estimated using the existing

depth estimation method, monodepth2 [5].

Component	Value	
Baseline	12 cm	
Image resolution	2208 pixels (width)	1242 pixels (height)
Focal length	1200 pixels	
Pixel size	0.002 mm	
Field of view	57° V FOV	96° H FOV
Focus distance	28 cm	

Table 4.1: Zed 2 camera calibration

4.4 CycleGAN

The conceptual idea behind the CycleGAN model originates from the human ability to picture a scene in a different circumstance than its current, regardless of whether this circumstance has been observed before. To illustrate this, an example: a person is walking through a random street in a city during day time and has never seen that street during night time. This person is still able to picture how that street would look at night time, because it has learned the style and characteristics of world at night time through earlier experiences. The CycleGAN model works in a similar fashion, since it is able to learn the mapping between two domains, as long as the two domains have an underlying relationship.

Normal GANs are composed of two networks: a generator and a discriminator, but a CycleGAN [96] network is composed of two individual GANs and therefore, consists of two generators and two discriminators. The CycleGAN network is designed to perform unpaired image-to-image translation, where its task is to translate images from the source domain A to the target domain B . One GAN is responsible for the translation from A to B and the other is for the translation from B to A . Both the two discriminators and the two generators represent a function:

The generator G_A maps images from domain A to domain B . Then D_B receives both these generated fake images and the real images from domain B , and predicts whether these images are real or fake. Similarly, generator G_B maps images from domain B to domain A and D_A predicts whether these images are real or fake.

In these domains exist, respectively, the real images $a \in A$ and the real images $b \in B$. A generated image in domain A is described as $\hat{a} = G_{B \rightarrow A}(b)$ and a generated image

Discriminators	Generators
$D_A : A \rightarrow \mathbb{R}$	$G_A : B \rightarrow \mathbb{R}$
$D_B : B \rightarrow \mathbb{R}$	$G_B : B \rightarrow \mathbb{R}$

Table 4.2: The functions that represent the discriminator and generator models in their respective domain

in domain B is described as $\hat{b} = G_{A \rightarrow B}(a)$). The distribution over \hat{a} and \hat{b} matches the empirical distribution over respectively $(data(a))$ and $(data(b))$. The notations for the real and generated (fake) images per domain are also summarized in Table 4.3.

	Domain A	Domain B
Real images	$a \in A$	$b \in B$
Fake images	$\hat{a} = G_{B \rightarrow A}(b)$	$\hat{b} = G_{A \rightarrow B}(a)$

Table 4.3: The notations for the real- and generated (fake) images for the discriminator and generator models in both domains

4.4.1 Loss functions

Loss functions reflect the distance between the distribution of the generated data and the distribution of the real data. They are used to calculate the network error. The CycleGAN model makes use of three different loss functions: an adversarial loss, a cycle-consistency loss and an identity loss. The adversarial loss measures the mean squared error (MSE) between the fake- and real image, whereas the cycle-consistency loss and the identity loss both measure the mean absolute error (MAE) between the fake- and real image. The MSE loss function and MAE loss function are according Equations 4.1 and 4.2, respectively.

$$L = \frac{1}{n} \sum_{i=1}^n (y^i - \hat{y}^i)^2 \quad (4.1)$$

$$L = \frac{1}{n} \sum_{i=1}^n |y^i - \hat{y}^i| \quad (4.2)$$

At a high level, the generators always try to maximize the probability of the discriminator making a mistake, whereas the discriminators always try to minimize the probability of making a mistake.

Adversarial loss

The adversarial loss for $G(A \rightarrow B)$ and D_B is calculated according to Equation 4.3 and 4.4, respectively. The adversarial loss for translation $A \rightarrow B$ is calculated according Equation 4.5. Corresponding equations are applied for the translation from $B \rightarrow A$.

$$L_{G_{A \rightarrow B}}(G_{A \rightarrow B}, D_B, a) = E_{a \sim p_{data(a)}} \left[D_B \left(G_{A \rightarrow B}(a) - 1 \right)^2 \right] \quad (4.3)$$

$$L_{D_B}(D_B, a, b) = E_{a \sim p_{data(a)}} \left[D_B \left(G_{A \rightarrow B}(a)^2 \right) \right] + E_{b \sim p_{data(b)}} \left[(D_B(b) - 1)^2 \right] \quad (4.4)$$

$$L_{GANA \rightarrow B}(G_{A \rightarrow B}, D_B, a, b) = L_{D_B}(D_B, b) + \min_{G_{A \rightarrow B}} \max_{D_B} L_{G_{A \rightarrow B}}(G_{A \rightarrow B}, D_B, a) \quad (4.5)$$

Cycle-consistency loss

The cycle-consistency loss, at a high level, encourages the generators to avoid unnecessary changes. As a result, the generators produce images that share structural similarities with the input images. In some sense, being more cycle-consistent reduces the creativity of the generators. It allows the generators to learn to cycle between the domains, e.g. when the output of $G_{A \rightarrow B}(a)$ is used as input for $G_{B \rightarrow A}$ it should reproduce image a . The cycle-consistency loss is calculated according Equation 4.6.

$$L_{CC}(G_{A \rightarrow B}, G_{B \rightarrow A}) = E_{a \sim p_{data(a)}} [||G_{B \rightarrow A}(G_{A \rightarrow B}(a))||_l] + E_{b \sim p_{data(b)}} [||G_{A \rightarrow B}(G_{B \rightarrow A}(b))||_l] \quad (4.6)$$

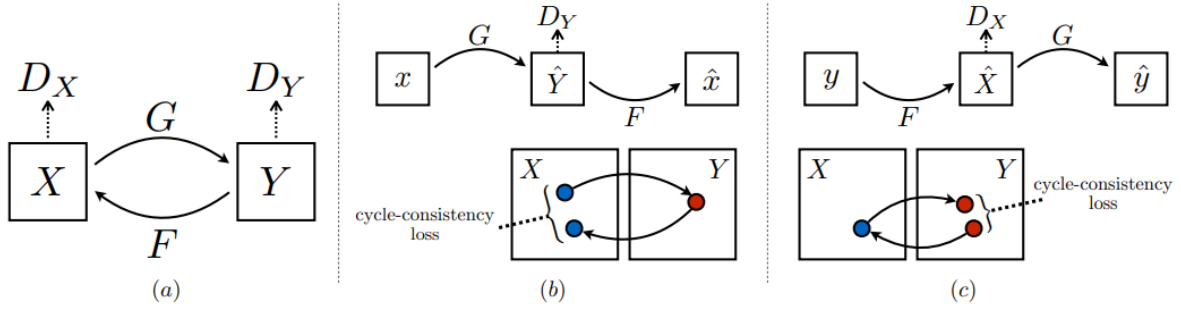


Figure 4.3: The model contains two functions that learn the mapping $G : X \rightarrow Y$ and $F : Y \rightarrow X$, and the associated discriminators D_Y and D_X . The discriminator D_Y encourages G to translate images from domain X into outputs indistinguishable from domain Y , similarly for D_X and F . This mapping is regularized using two cycle-consistency losses that capture the intuition that if one domain is translated to another and translate that output back to its original domain, that this result should be identical to the original input image. Hence, (b) forward cycle-consistency loss: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$, and (c) backward cycle-consistency loss: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$

Identity loss

The identity loss helps to preserve the colours of the input image and prevents reversed colours in the result. So as example, the network is given an image with a lot of blue colours and turns it into another style, but this causes the image to become a different colour. Including the identity loss ensures that this blue colour is preserved in the generated image. The influence of the identity loss is visualized in Figure 4.4 and its calculated according to Equation 4.7.

$$L_{ID}(G_{A \rightarrow B}, G_{B \rightarrow A}) = E_{a \sim p_{data(a)}} [||G_{B \rightarrow A}(G_{A \rightarrow B}(a)) - a||_l] + E_{b \sim p_{data(b)}} [||G_{A \rightarrow B}(G_{B \rightarrow A}(b)) - b||_l] \quad (4.7)$$

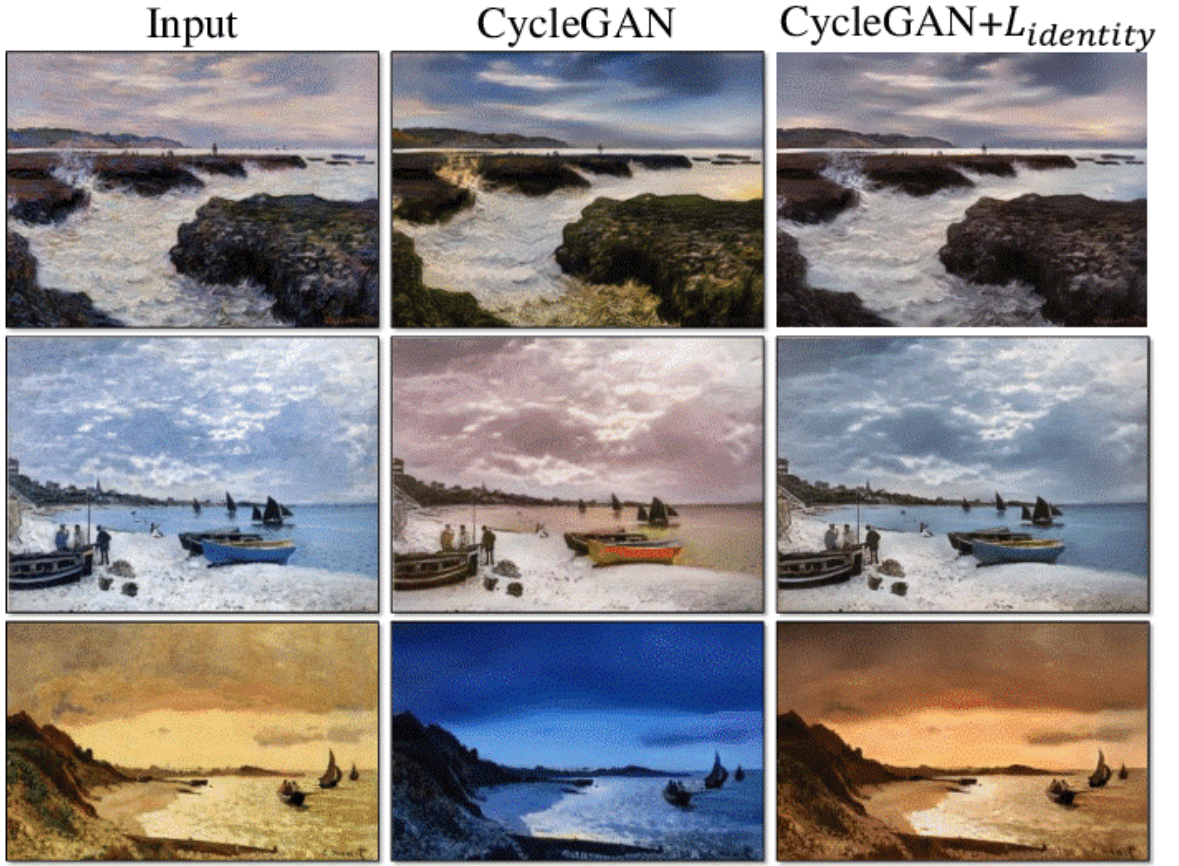


Figure 4.4: Illustration of the influence of the identity loss

4.4.2 Objective function

The full objective function of the CycleGAN is described in Equation 4.8. In here the hyper parameters λ_{CC} and λ_{ID} are introduced to control the impact of the cycle-consistency loss and identity loss, respectively. Ultimately, the model aims to solve Equation 4.9.

$$\begin{aligned}
 L(G_{A \rightarrow B}, G_{B \rightarrow A}, D_A, D_B) = & L_{GANA \rightarrow B}(G_{A \rightarrow B}, D_B, a, b) + \\
 & L_{GANB \rightarrow A}(G_{B \rightarrow A}, D_A, a, b) + \\
 & \lambda_{CC} L_{CC}(G_{A \rightarrow B}, G_{B \rightarrow A} + \\
 & \lambda_{ID} L_{ID}(G_{A \rightarrow B}, G_{B \rightarrow A}
 \end{aligned} \tag{4.8}$$

$$A^*, B^* = \arg \min_{G_{A \rightarrow B}, G_{B \rightarrow A}} \max_{D_A, D_B} L(G_{A \rightarrow B}, G_{B \rightarrow A}, D_A, D_B) \tag{4.9}$$

4.4.3 Model definition

A regular GAN is composed of two neural networks: a discriminator and a generator model. However, a CycleGAN is composed of two GANs, making it a total of two generators and two discriminators. In the following sections the network architecture of the generators and discriminators is addressed.

Generator architecture

The architecture of both generator models is displayed in Table X. The batch normalization layers and ReLU activation functions are excluded from this table, because those appear after every layer, i.e. row in the table. If the batch size is set to one (1), the batch normalization layer should be replaced with an instance normalization layer.

Function	Layer	Activation size
Encode	<i>Input image</i>	$3 \times 256 \times 256$
Encode	$64 \times 7 \times 7$ conv, 1 stride, 0 padding	$64 \times 256 \times 256$
Encode	$128 \times 3 \times 3$ conv, 2 stride, 1 padding	$128 \times 128 \times 128$
Encode	$256 \times 3 \times 3$ conv, 2 stride, 1 padding	$256 \times 64 \times 64$
Transform	9 Consecutive residual blocks	$256 \times 64 \times 64$
Decode	$128 \times 3 \times 3$ conv, 2 stride, 1 padding, 1 outer padding	$128 \times 128 \times 128$
Decode	$128 \times 3 \times 3$ conv, 2 stride, 1 padding, 1 outer padding	$64 \times 256 \times 256$
Decode	$3 \times 7 \times 7$ conv, 2 stride, 0 padding	$3 \times 256 \times 256$

Table 4.4: Generator architecture

The input of the generators is a RGB image (3 channels) with a size of 256×256 pixels. This image is down-sampled, transformed and up-sampled again into a RGB image of the same size. The result is the generated image. The kernel size of the first- and the last convolutional layer is 7×7 and the kernel size of the other (transposed) convolution layers is 3×3 . The residual blocks each contain two 3×3 convolutional layers with 128 filters on both layers. The architecture of an individual residual blocks is shown in Figure 4.5.



Figure 4.5: Residual block that is used by the generators

Discriminator architecture

The architecture of both discriminator models is displayed in Table X. The batch normalization layers and Leaky ReLU activation functions are excluded from this table, because those appear after every layer, i.e. row in the table. If the batch size is set to one (1), the batch normalization layer should be replaced with an instance normalization layer.

Layer	Activation size
<i>Input image</i>	$3 \times 256 \times 256$
$64 \times 4 \times 4$ conv, 2 stride, 1 padding	$64 \times 128 \times 128$
$128 \times 4 \times 4$ conv, 2 stride, 1 padding	$128 \times 64 \times 64$
$256 \times 4 \times 4$ conv, 2 stride, 1 padding	$256 \times 32 \times 32$
$512 \times 4 \times 4$ conv, 2 stride, 1 padding	$512 \times 31 \times 31$
$14 \times 4 \times 4$ conv, 2 stride, 1 padding	$1 \times 30 \times 30$

Table 4.5: Discriminator architecture

The input of the discriminators is a RGB image (3 channels) with a size of 256×256 pixels. The output of the discriminator is tensor of 30×30 with a depth of one (1). The kernel size of all convolutional layers is 4×4 .

Full model architecture

Figure 4.6 illustrates how the actual model architecture looks like. It captures the entire CycleGAN architecture and where every component fits into the model.

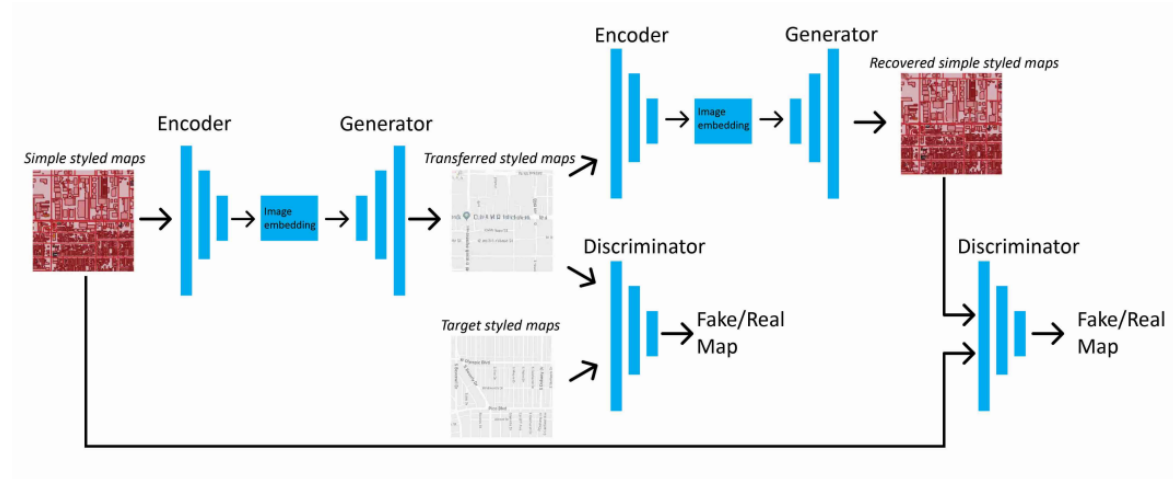


Figure 4.6: CycleGAN architecture [96]

4.4.4 Training details

The discriminators and generators are both optimized with the Adam optimizer, they all have the same learning rate ($2e^{-4}$), which linearly decays over a set amount of epochs to zero (0) and the batch size is set to one (1). If a batch size of one is used, the batch normalization layers in both the generators and the discriminator architectures should be replaced by instance normalization layers. Lastly, λ is set to 10 and a replay buffer [152] is used to reduce model oscillation and it can store the last 50 images.

The weights of the generators are updated by minimizing the sum of the adversarial loss, cycle-consistency loss and, if used, the identity loss, according to Equations 4.3, 4.6 and 4.7. The weights of the discriminators is updated by minimizing the adversarial loss, according to Equation 4.4. The entire CycleGAN model is tries to minimize the total loss, according to Equation 4.8 and ultimately tries to solve Equation 4.9.

In the remainder of this thesis there will trained both smaller models and larger models. In Chapter 5 smaller models will be trained to quickly compare many different model configurations, whereas in Chapter 6 four large models will be trained in order to fully test their potential. The small models will be trained according the described training parameters above, for 20 epochs and the learning rate will start to decay after 10 epochs. The large models will be trained for 200 epochs and the learning rate will start to decay after 100 epochs.

4.5 Framework

The main pipeline exists out of multiple processes that jointly cooperate and are called consecutively from the main script file. Individual processes, such as data loading, stereo image-pair synthesis, pre-processing and depth estimation exist in an own file. The deep learning models for stereo image-pair synthesis and depth estimation are trained individually and separately from the main script. The trained weights of the deep learning models are stored somewhere in the project folder and then called from the main script upon running it. The CycleGAN implementation has been coded from scratch according the method previously addressed. This model has been experimented with, to improve the final results. These experiments, tests and further fine-tuning techniques of the model will be addressed in chapter 5.

Package	Version
PyTorch	$\geq 1.7.0$
PyTorch_FID	$\geq 0.2.0$
Torchvision	$\geq 0.10.0$
Matplotlib	$\geq 3.4.2$
Scipy	$\geq 1.7.0$
Numpy	$\geq 1.21.0$
Tqdm	$\geq 4.61.1$
Labml-helpers	$\geq 0.4.77$

Table 4.6: Required python packages and their minimum version

The complete project has been developed in a fresh Anaconda environment running *Python* 3.8.5. The two main deep learning models have been trained using the *Pytorch* 1.7.1 library. To handle- and deal with the large matrices of data, *NumPy* 1.20 was chosen. Visualizing the progress during training and testing *matplotlib* 3.3.2 is used, since it enables the user to quickly make insightful graphs and plots. The necessary packages that have to be installed in the environment are listed in Table 4.6.

4.6 Evaluation metrics

In this section the evaluation measures that are used to assess the performance of the CycleGAN model is addressed.

4.6.1 FID

The main evaluation metric that is used to test the performance of the CycleGAN model is the Fréchet Inception Distance (FID) Score [153], which is the standard method to assess the quality of GANs anno 2021. It captures the similarity between two distribution and in our case, this means that the FID score measures the similarity between the data distribution of the generated depth maps and the data distribution of the real depth maps. The FID score is an improvement to the Inception Score (IS). The FID score uses the Inception Network (IN) to extract features from an intermediate layer [?]. Both data distributions are modelled using a multivariate Gaussian distribution, that is based on the mean and co-variance of that distribution. The FID between distribution x and g is calculated according to Equation 4.10.

$$FID(x, g) = ||\mu_x - \mu_g||_2^2 + Tr \left(\sum_x + \sum_g -2 \times \left(\sum_x \times \sum_g \right)^{\frac{1}{2}} \right) \quad (4.10)$$

In comparison to the IS and other evaluation metrics, the FID score is a more robust evaluation metric, since it has less interference of noise and is better at measuring the image diversity. Ideally, the values of the FID score range between $[0, \infty]$. A perfect similarity between two distribution has a FID score of zero (0) and the less similar the two distributions become, the higher the FID score becomes.

4.6.2 RMSE

The second evaluation metric that is used to evaluate the performance of the CycleGAN model is the Root Mean Square Error (RMSE) [154]. This evaluation metric compares, within the boundaries of image width m and image height n , each individual pixel I at coordinate x, y in the generated depth maps to their corresponding pixel \hat{I} at coordinate x, y in the real depth map. Unlike the FID score, the RMSE is calculated for every individual image in the dataset and then the average over all images is taken.

Similarly to the FID score, a perfect similarity between two images gives a RMSE of zero (0) and the less similar the two images become, the higher the RMSE value becomes. The RMSE value of an image is calculated according to Equation 4.11.

$$RMSE = \sqrt{\frac{\sum_{x=1}^m \sum_{y=1}^n (I_{x,y} - \hat{I}_{x,y})^2}{m \times n}} \quad (4.11)$$

Experiments

In this chapter the experiments of this research will be presented, which addresses why these experiments are performed, how these experiments are conducted and how the experimental results impact the remainder of this research.

5.1 Goal

The goal of the experiments is to find the most optimal network architectures for the two tasks at hand: stereo image-pair synthesis and depth estimation. Throughout multiple, small experiments various components and their influence is assessed. Components, such as; architectural changes, image pre-processing steps and training techniques. Then, the results of each experiment is evaluated using the same evaluation metrics, according to the provided methodology in Chapter 4.

After all the experiments are performed and their results are evaluated, they are compared in order to assess their influence on the results. Based on these experimental results and evaluation, the final two models are configured accordingly. These models will be trained on a larger scale. Subsequently, their results will be evaluated in a similar fashion as the experiments and their performance will be reviewed- and discussed in Chapter 6.

5.2 Training parameters

In every individual experiment, the same parameters are used for training the models. A manual seed of 999 is used to reproduce the initial random weights that are assigned to the models. The training data is being *shuffled*. The models are optimized using the *Adam* optimizer, at a learning rate of $2e^{-4}$ using a batch size of 1. The models are trained for 20 epochs and after 10 epochs the learning rate starts to decay linearly to zero over the remaining epochs.

5.3 Experiment 1: the influence of the identity loss function

Identity loss should help preserve the colours of the input image and prevent reversed colours in the result. Therefore it is expected that identity loss will not aid in generating a better depth map, since the generator has to distort the colours of the input image. Nonetheless, identity loss is expected to aid in generating realistic stereo image-pair images, either left or right. Preservation of the colours of the input image in the result is desired.

5.3.1 Setup

In this experiment the influence of the identity loss function (Section 4.4.1) is examined on stereo image-pair synthesis and depth estimation. For stereo image-pair synthesis this is done for RGB images only. For depth estimation this is examined for both gray-scaled images as well as RGB images.

The models in this experiment have been trained according the training parameters that are described in Section 5.2. For each model the FID score and the average RMSE has been calculated. This is done over the entire test set. The results of this experiments are shown in Table 5.1.

5.3.2 Results

Visual inspection of the outputs shown in Table 5.1, show us that indeed the identity loss negatively affects the generated depth maps, but also that the coloured, i.e. 3C (3 channels), depth maps generated have a greater resemblance with the ground truth image in comparison to their gray-scaled equivalent. In contrast, the stereo image-pair synthesis result become much better when the identity loss function is used. The image looks less pixelated, has brighter colours and has a greater resemblance with the ground truth image.

Figures 5.1 and 5.2 contains a plot that show the FID score and RMSE loss values for both models: (a) stereo-to-depth (S2D) and (b) left-to-right (L2R). It can be observed that for the grayscale S2D images (row 1, Table 5.1) the FID score of the outputs generated with identity loss have a lower FID score, but hardly look like the ground truth model. The coloured S2D images (row 2, Table 5.1) that have been generated using identity loss also have little resemblance with the ground truth image.

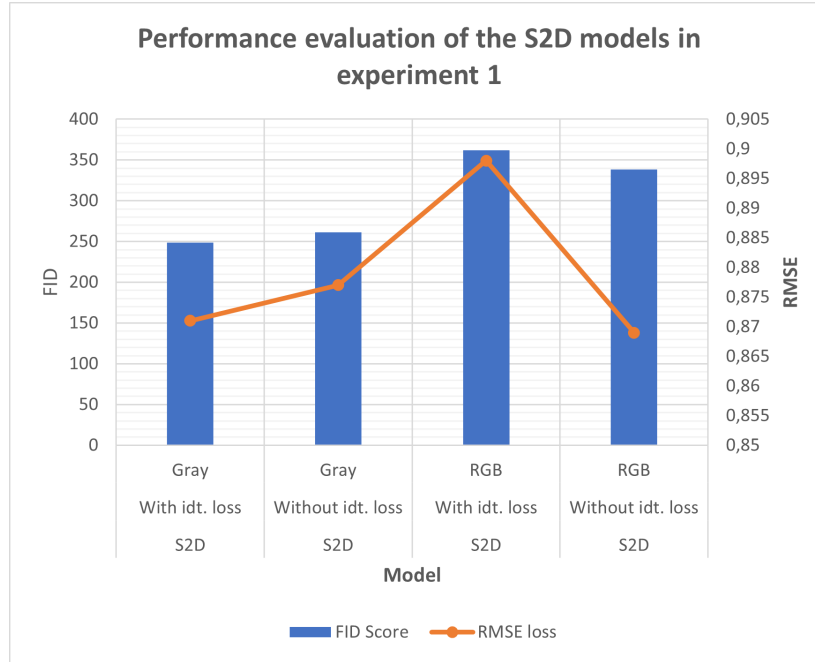


Figure 5.1: Performance evaluation of the stereo-to-depth (S2D) models in experiment 1 using the FID score and the RMSE loss

This is also confirmed by the FID score. Similarly to other models, the RMSE value doesn't provide much usable information to evaluate the difference between models.

Evaluation of the generated output images by the L2R models (row 3, Table 5.1) confirm the expectation that identity loss should be used when generated a right-view from a given left-view (or vice versa). The outputs generated by a model *without* identity loss have a significantly higher FID score in comparison the outputs generated by the *with* identity loss. Visual inspection also confirms this.

The results of this experiment clearly confirm that, in the case of synthesizing stereo-view images (L2R), identity loss is useful because we want to preserve the colours that already exist in the input image. In contrast, transforming a stereo-view image to a depth map (S2D) requires the input images to be totally transformed. Therefore, identity loss only counteracts the model in learning the desired translation.

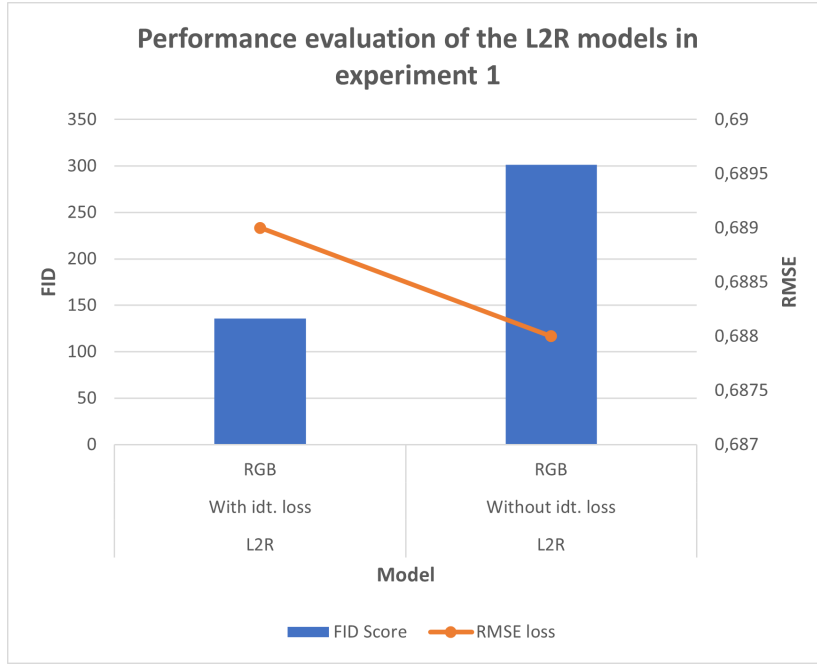


Figure 5.2: Performance evaluation of the left-to-right (L2R) models in experiment 1 using the FID score and the RMSE loss

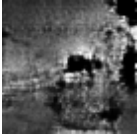
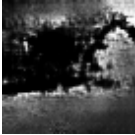

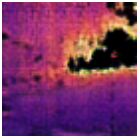
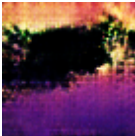




	With identity loss	Without identity loss	Ground truth
1C	 $FID \approx 248.8$ $RMSE \approx 0.871$	 $FID \approx 261.3$ $RMSE \approx 0.877$	 $FID = 0.000$ $RMSE = 0.000$
3C	 $FID \approx 362.1$ $RMSE \approx 0.898$	 $FID \approx 338.6$ $RMSE \approx 0.869$	 $FID = 0.000$ $RMSE = 0.000$
3C	 $FID \approx 135.8$ $RMSE \approx 0.689$	 $FID \approx 301.4$ $RMSE \approx 0.688$	 $FID = 0.000$ $RMSE = 0.000$

Table 5.1: Results of the experiment on the influence of identity loss for in the depth estimation model with grayscale (1C) and RGB (3C) images

5.4 Experiment 2: the influence of network architecture

In this experiment the influence of various architectural changes in both the generator architecture and the discriminator architecture are examined. Consult Section 4.4.3 for a more detailed explanation of the generator- and discriminator architecture. The architectural changes that are covered are the activation function and the upsampling method. In the original model they make use of the *ReLU* activation function and use *Transposed Convolutional* layers for upsampling. the *ReLU* activation function suffers from a problem, which is known as the "*dying ReLU problem*" [155]. This refers to the problem that neurons become inactive and start to output 0 (zero) only, irregardless of the input. The *Leaky ReLU* activation function tends to solve this issue and is therefore examined as a potential solution.

Transposed Convolutional layers, also known as *Deconvolutional* layers, are used in the original model architecture, but this upsampling method tends to suffer from a problem that is known as the checkerboard artifacts [156]. Due to this way of upsampling, lower quality images can be upsampled to higher quality images. However, if the kernel size is not divisible by the stride, an uneven overlap is created. Therefore, the kernel of the upsampling layer projects the new value onto the target pixel more often than on other pixels and in doing so, it creates a checkerboard artifact. This can be solved by replacing the transposed convolutional by a k-nearest neighbour upsampling layer, followed by a convolutional layer.

According to the authors of the CycleGAN paper, it is normal that the discriminator loss becomes very small, whilst the generator loss increases. Normally, this is considered to be a bad result, because this characterizes GAN instability and thus, the GAN can suffer from mode collapse or reaches a state of non-convergence.

5.4.1 Setup

In this experiment the influence of the *ReLU* or *Leaky ReLU* in combination with a *Transposed Convolutional* layers or a *K-nearest neighbours upsampling* layer is examined against a discriminator architecture with and without *dropout layer* with a dropout percentage of 30%. These experiments are performed for the depth estimation model using grayscaled (1C) images and coloured (3C) images. The models in this experiment have been trained according the training parameters that are described in Section 5.2. For each model the FID score and the average RMSE has been calculated.

5.4.2 Results

The results of this experiments are shown in Table 5.2. Initial inspection of the FID scores, that are also plotted in Figure 5.3, show that the original model based on Transposed Convolutions and the ReLU activation function has the lowest FID scores. This, however, is not that surprising given the working of the upsampling method. Since it projects new values onto one pixel at a time, whereas k-nearest neighbour based upsampling methods take the average of its neighbouring pixels around it. The difference between these two upsampling methods is greatest when grayscale images are used. This difference becomes smaller once coloured images are used.

Looking at the activation function, Leaky ReLU seems to perform better than the ReLU activation function for coloured images. Especially in the black areas. Adding dropout to the discriminator seems to slow down the learning speed. Take for instance the image in column 4, row 3: Visually, this looks like the model that performs best. However, when there's dropout added to the discriminator (column 4, row 4) the result (visually and FID score) becomes worse, while the generator model remains the same.

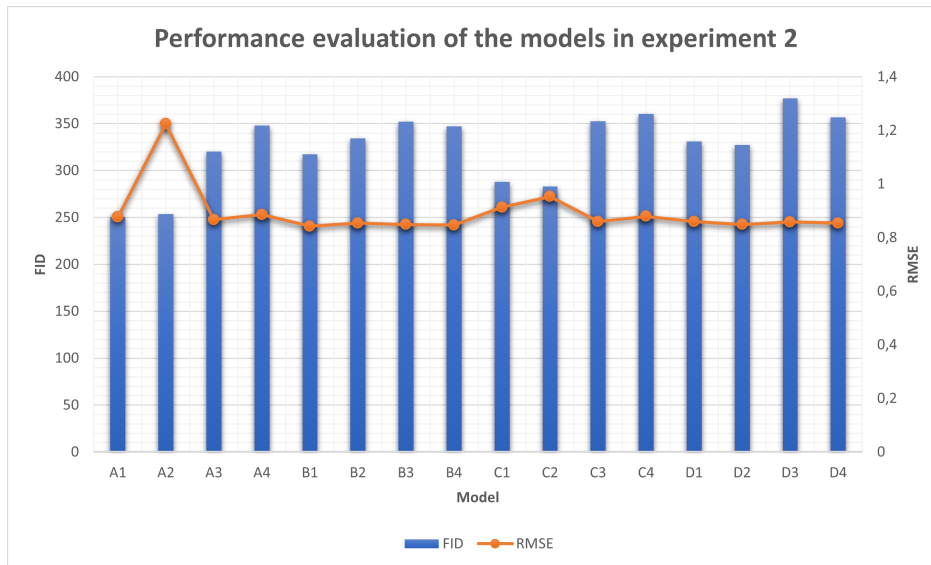


Figure 5.3: Performance evaluation of the models in experiment 2 using the FID score and the RMSE loss, where: the letter describes the column index from left to right (A; B; C; D) and the number (1; 2; 3; 4) describes the row index from top to bottom, this represents the position of the mode in Table 5.2

Using the results of this experiment, the most promising generator network architecture will be used in the experiment described in Section 5.5 to test how restricting the discriminator's ability to learn will influence the generated output. Mostly based

on visual inspection, the most promising generator network architecture is the one based on k-nearest neighbour up-sampling and the Leaky ReLU activation function, using RGB input images.

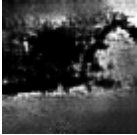

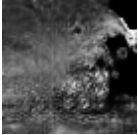


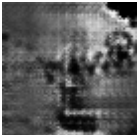
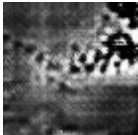

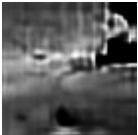


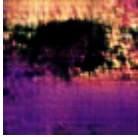
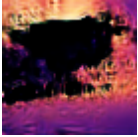

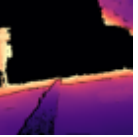
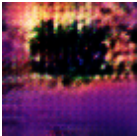
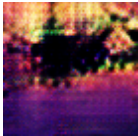
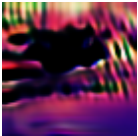
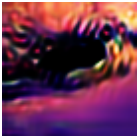
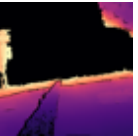
		Transposed convolutions		K-nearest neighbour upsampling		Ground truth
		ReLU	Leaky ReLU	ReLU	Leaky ReLU	
Without dropout	1C	 $FID \approx 250.3$ $RMSE \approx 0.876$	 $FID \approx 253.5$ $RMSE \approx 1.224$	 $FID \approx 320.4$ $RMSE \approx 0.867$	 $FID \approx 348.1$ $RMSE \approx 0.886$	 $FID = 0.000$ $RMSE = 0.000$
	1C	 $FID \approx 288.1$ $RMSE \approx 0.913$	 $FID \approx 282.9$ $RMSE \approx 0.953$	 $FID \approx 352.7$ $RMSE \approx 0.860$	 $FID \approx 360.1$ $RMSE \approx 0.879$	 $FID = 0.000$ $RMSE = 0.000$
Without dropout	3C	 $FID \approx 317.2$ $RMSE \approx 0.842$	 $FID \approx 334.2$ $RMSE \approx 0.854$	 $FID \approx 352.1$ $RMSE \approx 0.849$	 $FID \approx 347.0$ $RMSE \approx 0.847$	 $FID = 0.000$ $RMSE = 0.000$
	3C	 $FID \approx 331.1$ $RMSE \approx 0.860$	 $FID \approx 327.4$ $RMSE \approx 0.849$	 $FID \approx 377.0$ $RMSE \approx 0.858$	 $FID \approx 356.7$ $RMSE \approx 0.854$	 $FID = 0.000$ $RMSE = 0.000$

Table 5.2: Results of the experiment on various architectural changes in the generator model and *with or without* dropout layers in the discriminator model on grayscale images (1C) and coloured images (3C)

5.5 Experiment 3: the influence of restricting the discriminator’s ability to learn

The discriminator is according to the authors of the original much more powerful than the discriminator. According to their paper, this doesn’t seem to immediately affect the quality of the generated samples. Nonetheless, a discriminator that is too powerful generally renders the generators unable to produce a wide variety of samples and withholds the generator from improving any further. To illustrate this phenomenon: the discriminator can overfit on a certain feature in the generated images. An example of such a feature is that the distribution of the real data is a matrix of $n/255$, whilst the distribution of the generated samples is not. Hence, there is sufficient reason to examine whether restricting the discriminator’s ability to learn is beneficial for the generator’s ability to produce good quality depth maps.

5.5.1 Setup

So, in this experiment experiment the influence of disabling the discriminator learning is examined. Three methods have been explored; label flipping, label smoothing and adding Gaussian noise to the input image of the discriminator [157]. During initial development, the impact of these individual was minimal, therefore they are combined into one single learning restriction. This influence is examined against a discriminator architecture either with or without *dropout layers* that have 30% dropout, on coloured (3C) images only. The reason for using coloured images only, is that the results of Section 5.4 showed that using coloured images leads to the best results. For each model the FID score and the average RMSE has been calculated. This is done over the entire test set. The results of this experiments are shown in Table 5.3.

As the output samples in Table 5.3 clearly indicate, leaving the discriminator unrestricted and without dropout produced visually the best result. According to the FID score the similarity of the generated test samples has a value that is not so different from the others, except from the most-right sample. The visible sample looks like it’s the worst of all, but the FID suggests that the overall distribution has the most similarity with the ground truth distribution.

Adding dropout to the discriminator, without restricting its learning, seems to make it a little dumber. This is in line with how dropout layers should work. It prevents over-fitting on the training data. Restricting the discriminator’s learning seems to have a positive effect on the FID score in general, because for every scenario the FID score

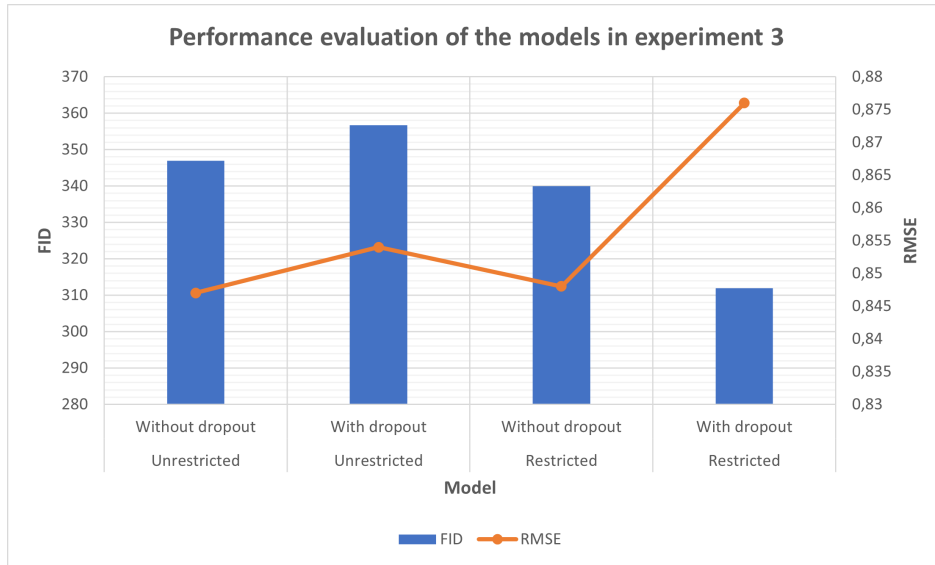


Figure 5.4: Performance evaluation of the models in experiment 3 using the FID score and the RMSE loss

is lower than when the discriminator's learning is unrestricted.

To summarize, restricting the discriminator's ability to learn and adding dropout layers to prevent over-fitting on the training data, seems to have a positive effect on the similarity between the generated and real distribution. Visual inspection of the generated samples does not yet seem to confirm this. It is however, important to point out that these models have only been trained for 20 epochs and are therefore not the final results.


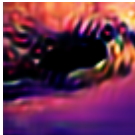
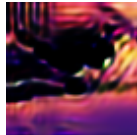


	Unrestricted discriminator learning		Restricted discriminator learning		Ground truth
	Without dropout	With dropout	Without dropout	With dropout	
3C					
	$FID \approx 347.0$ $RMSE \approx 0.847$	$FID \approx 356.7$ $RMSE \approx 0.854$	$FID \approx 340.0$ $RMSE \approx 0.848$	$FID \approx 311.9$ $RMSE \approx 0.876$	$FID = 0.000$ $RMSE = 0.000$

Table 5.3: Results of the experiment on using learning restrictions on the discriminator, i.e. label flipping, label smoothing and adding Gaussian noise to the input image of the discriminator, and *with* or *without* dropout layers in the discriminator on coloured images (3C)

Results and discussion

In this chapter the main results of this research are presented, which contains the description of the four most promising network configurations and evaluation those full-scale model output and performance.

6.1 Synthesizing stereo image-pairs

The first component of the overall network is the stereo image-pair synthesis network. Its evaluation is documented in a more compact manner as comparison to the evaluation of the depth estimation network. At first two generator network architectures are compared; the original network architecture of the CycleGAN [144] and the network architecture that performed best in the experiment conducted in Section 5.5. The performance of these two models is evaluated, both through: visual inspection and through comparing the results of the described evaluation metrics.

6.1.1 Model configuration and evaluation

At first two major network architectures, i.e. two generator models, are compared to each other and their output is evaluated using the FID score and RMSE loss metrics. Naturally, the FID score is calculated over the entire dataset at once, whereas the value of the RMSE loss is the average of the RMSE loss of every individual output sample. These results are shown in Table 6.1.

	Generator		Performance			
	Activation	Upsampling	FID_A	FID_B	$RMSE_A$	$RMSE_B$
Model 1	ReLU	ConvTranspose	165.8	165.6	0.687	0.679
Model 2	LeReLU	KNN	226.3	225.7	0.693	0.693

Table 6.1: Overview of stereo image-pair synthesis models and their configuration, along with their evaluated performance using FID and RMSE

As is marked in bold letters in Table 6.1, the performance of *model 1* scores significantly lower than *model 2*. Therefore, it is clear that a network architecture that uses the ReLU activation function and up-samples using Transposed Convolutional layers performs better than the updated Leaky ReLU activation function with k-nearest neighbour upsampling.

6.1.2 Visual inspection of the synthesized views

Visual inspection only confirms these findings. Figure 6.2 shows how both models synthesize a right-view image given a left input-view image and how it compares to the ground truth right-view image. Naturally, the models can also synthesize a left-view from a given a right-view image, but this is not included in the Figure 6.2. As the samples have a size of only 100×100 pixels, Figure 6.1 contains an enlarged sample of both model outputs that allows for easier inspection.



(a) Synthesized right-view of model 1



(b) Synthesized right-view of model 2

Figure 6.1: Enlarged synthesized right-views of both models

Although visual inspection of images that have a high similarity, i.e. the left- and right view, can be hard. It can be observed that: the right-view image synthesized by *model 1* is slightly sharper than the right-view image synthesized by *model 2*, and also contains much less (colorful) imperfections.







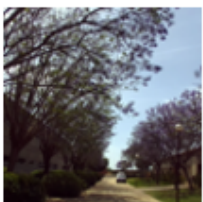
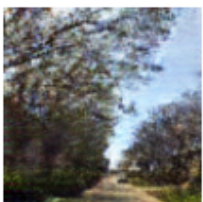
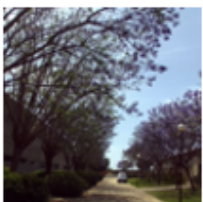
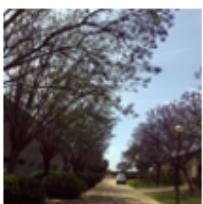
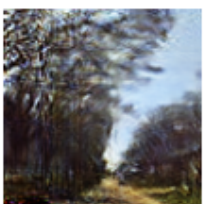
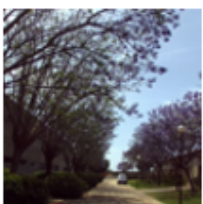
	Input left-view	Synthesized right-view	Ground truth right-view
Model 1			
Model 2			
Model 1			
Model 2			

Figure 6.2: Visual inspection of the output of the generator $G_{A \rightarrow B}$ of both model; this generator synthesizes a right-view image based on the input image (left-view)

6.1.3 Evaluation of the synthesized stereo-image pairs

To investigate the influence of synthesizing particular views in a stereo image-pair on quality of the depth estimation, three new datasets are constructed. The stereo image-pairs in these datasets consists either out of; no synthesized views, a synthesized left-view, a synthesized right-view or both a synthesized left- and right view.



Figure 6.3: A high-quality example of a stereo image, in which non-synthesized left- and right-views are merged into a single stereo view

The left-view and right-view image are combined into a stereo image-pair by reducing their transparency to 50% and placing them on top of each other, after which they are normalized again to match the needs of our network. The dataset that contains no synthesized views is the regular test dataset. Samples of these four datasets are shown in Figure 6.4 (on the next page).

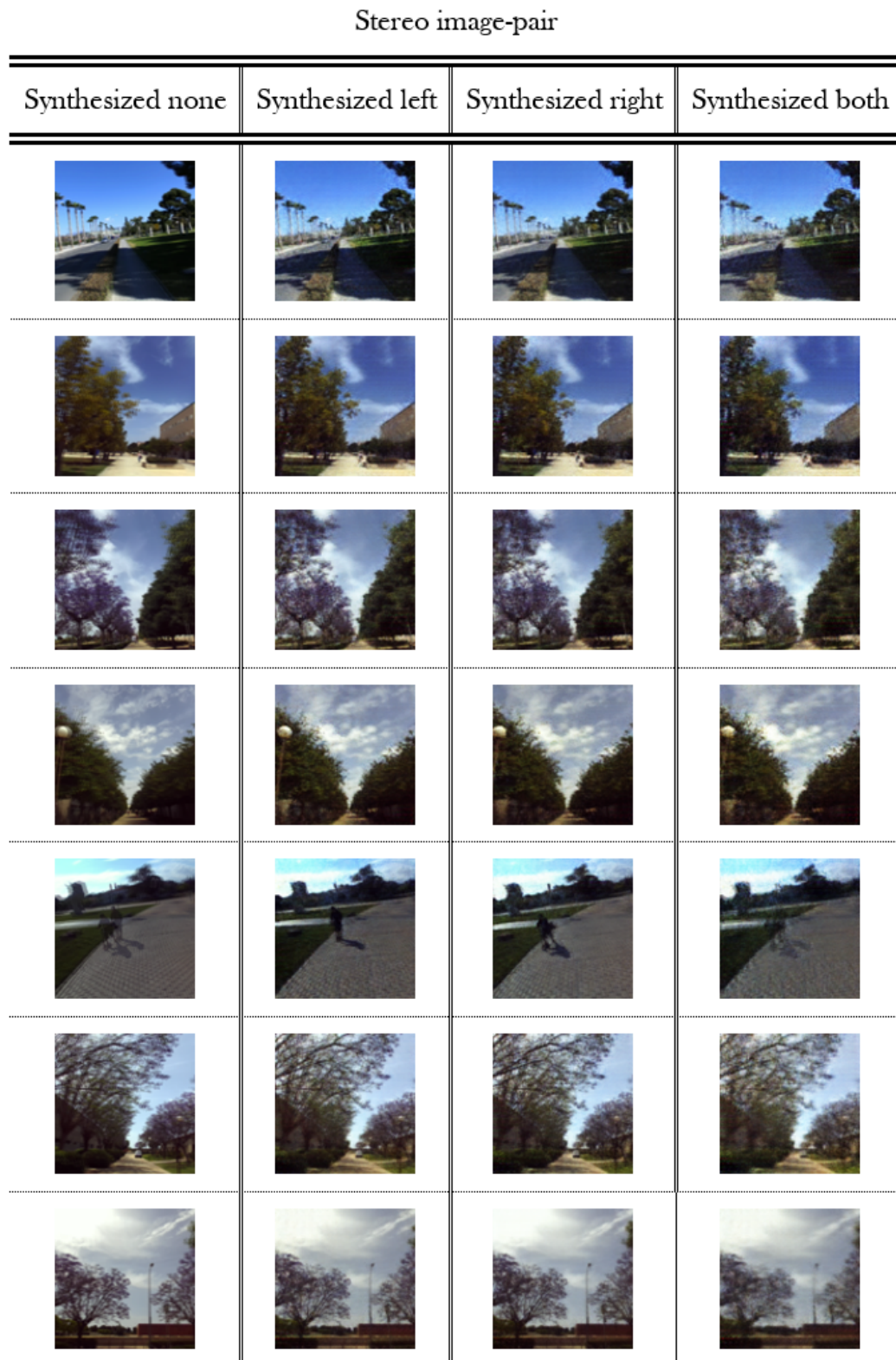


Figure 6.4: Visual inspection of created stereo image-pairs, where a left-view and right-view image are combined, which are either synthesized or real

6.1.4 Evaluation of rotation- and translation of objects in the scene

The individual images seem alright, but the most important aspect is to correctly synthesize a left- or right-view from a given single-view. Since this is not easily done by placing images in a document, there are two links provided to a *.GIF* file at my personal Google Drive, in which you can see the left-view and the right-view image alternate in a continuous loop. Using this method you can see how well the synthesis network is able to properly rotate- and translate certain objects in the scene.

These links are provided below, one link redirects you to the real stereo image-pair and one link redirects you to the synthesized stereo image-pair.

1. [Synthesized stereo image-pair \(click me\)](#)
2. [Realistic stereo image-pair \(click me\)](#)

6.2 Selecting the best model for depth estimation

The second component of the overall network is the depth estimation network, which estimates a depth map for a given stereo image-pair. It is evaluated on several aspects; (1) the four most promising combinations of network architectures and training techniques are used to train a full model; (2) the loss of these models is monitored during training; (3) then the four models are evaluated on the standard test set (containing non-synthesized stereo image-pairs), using the described evaluation metrics and through visual inspection; (4) at last, the best performing model is tested against the collection of datasets containing different types of synthesized stereo image-pairs, as is described in Section 6.1.

	Generator		Discriminator	
	Activation	Upsampling	Dropout	Restrictions
Model 1	ReLU	ConvTranspose	✗	✗
Model 2	LeReLU	KNN	✓	✓
Model 3	LeReLU	KNN	✗	✗
Model 4	LeReLU	KNN	✗	✓

Table 6.2: Overview of the four most promising model configurations for the depth estimation models

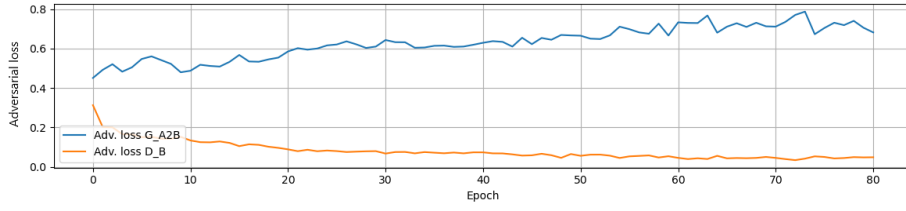
From the experiments conducted in Chapter 5, the four most promising model configurations have been selected. For each configuration a full model is trained, accordingly. The model configurations are summarized in Table 6.2.

6.3 Training the models

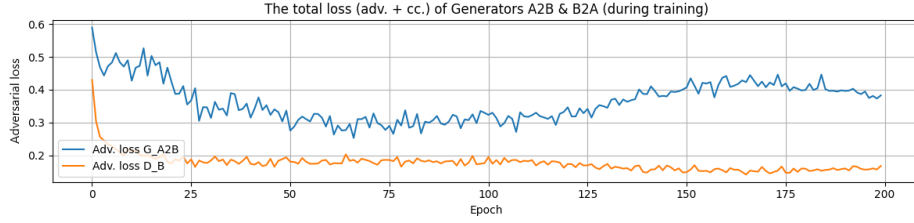
The four models, as are shown in Table 6.2, have been trained for 200 epochs, where the learning rate linearly starts to decay after 100 epochs. Further training details are discussed in 4.4.4. During training the losses of the generators $G_{A \rightarrow B}$, $G_{B \rightarrow A}$ and discriminators D_A , D_B have been tracked. This is done for each batch and with that, the average over the entire epoch is calculated and plotted. Similarly for the FID score and the RMSE loss.

The losses of generator $G_{A \rightarrow B}$ and discriminator D_B are plotted for each model in a sub-figure, that can be found in Figure 6.5. The losses of generator $G_{B \rightarrow A}$ and discriminator D_A are not shown, since we are most interested in generating images in domain B , i.e. a depth map. In addition, the per-epoch FID score is also plotted for each model and this can be found in Figure 6.6. The RMSE loss is not included in the results chapter, as this is a less valuable evaluation metric.

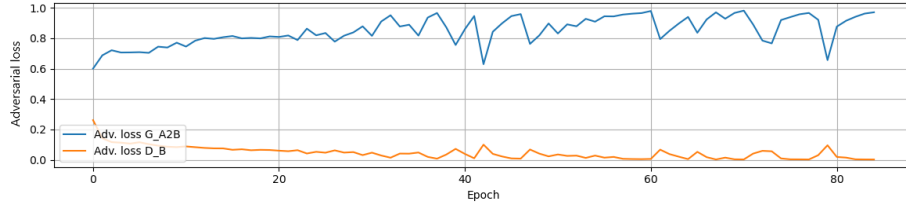
6.3.1 Tracking the adversarial loss



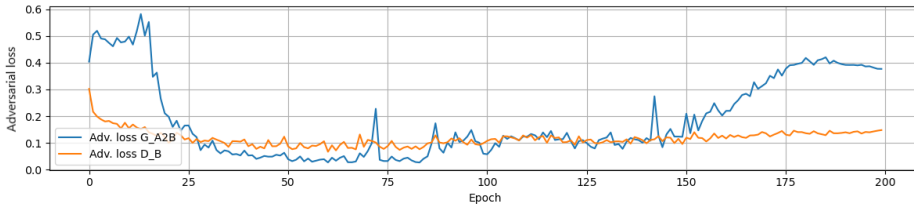
(a) Model 1 (plot above): per-epoch adversarial loss of the $G_{A \rightarrow B}$ (blue) and D_B (orange) during the training



(b) Model 2 (plot above): per-epoch adversarial loss of the $G_{A \rightarrow B}$ (blue) and D_B (orange) during the training



(c) Model 3 (plot above): per-epoch adversarial loss of the $G_{A \rightarrow B}$ (blue) and D_B (orange) during the training



(d) Model 4 (plot above): per-epoch adversarial loss of the $G_{A \rightarrow B}$ (blue) and D_B (orange) during the training

Figure 6.5: Plots of the per-epoch adversarial loss of the $G_{A \rightarrow B}$ (blue) and D_B (orange) during training of, respectively: (a) model 1; (b) model 2; (c) model 3 and (d) model 4

Normal GANs do not converge, with the exception of the Wasserstein GAN, but generally it can be observed that the loss of both the generator and the discriminator stabilizes out over time. Closer inspection of the plots above shows us that **model 1** and **model 3** are **GAN failure** modes. For this reason training has been stopped early on, because it failed very soon and did not seem to improve anymore. The discriminator loss keeps decreasing right from the start, whereas the generator loss keeps increasing in a similar fashion. This learns us that the discriminator is much more

powerful than the generator and so, becomes increasingly better at distinguishing the fake from the real samples created by the generator. The gradient of the generator vanishes, because the discriminator is not providing enough useful information to the generator. As a result, the generator can not make any progress, whilst the discriminator keeps improving. Notably, the divergence of the generator and discriminator loss is only observed in the models where the discriminator is not subjected to any learning restrictions or contains dropout layers.

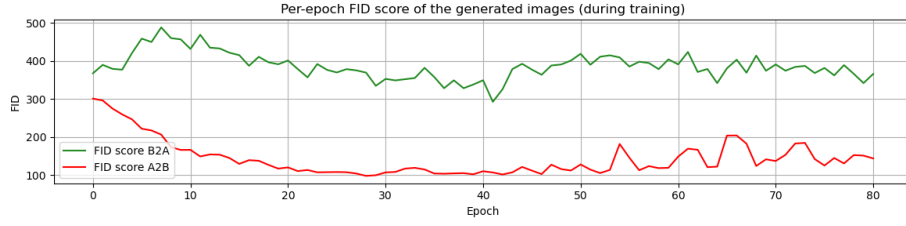
This divergence between generator- and discriminator loss can also lead to mode collapse, which is a scenario in which the generator only learns to produce one single output (mode) or a small subset of outputs (modes). Mode collapse is however not always a bad thing, because it is also the phase in which the generator over-optimizes one single mode. This means that the generator has found one mode or a few modes that fool the discriminator, i.e. look realistic. A generator that is over-optimizing a certain mode, will improve the quality of that mode and in doing so, the output sample becomes more and more realistic.

Model 2 and **model 4** seem to perform significantly better. Both losses steadily decrease over time and it seems like the gradient, suggesting that the quality of the generated samples steadily improves and the discriminator doesn't dominate the generator. Therefore, both networks learn in a similar fashion at a comparable pace. Notably is that the loss of the generator in model 4 drops very quickly after approximately 20 epochs and even drops a little below the discriminator loss. After which the losses seemed to converge for another 125 epochs. At epoch 150 the generator loss suddenly increases, suggesting that the generator couldn't fool the discriminator anymore. However, after another 25 epochs it drops again and it's loss actually drops again whilst the discriminator loss shows an upwards trend. It looks like that with some more training, assuming this trend continues, the quality of the generated samples would increase even more.

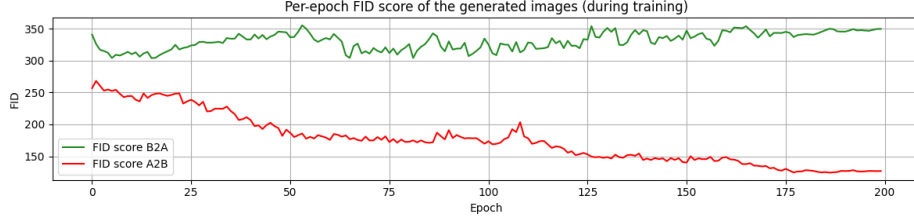
6.3.2 Tracking the FID score

*As a **note upfront**: each sub-plot of Figure 6.6 contains the FID scores of both \hat{a} and \hat{b} . Given that our interest lies in generating a depth map from a stereo image-pair $\hat{b} = G_{A \rightarrow B}(a)$ (red) and not in generating a stereo image-pair from a depth map $\hat{a} = G_{B \rightarrow A}(b)$ (green), the following section will mostly address the red line in the plots.*

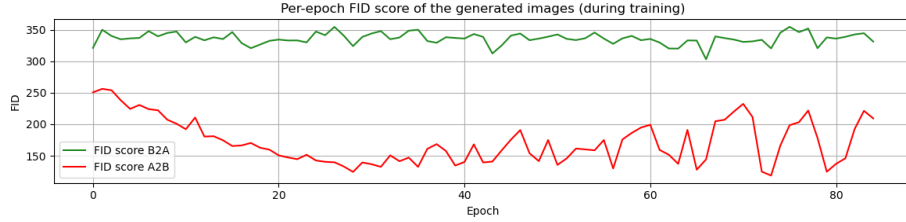
The observed behavior of the loss function of **model 1** and **model 3** can also found



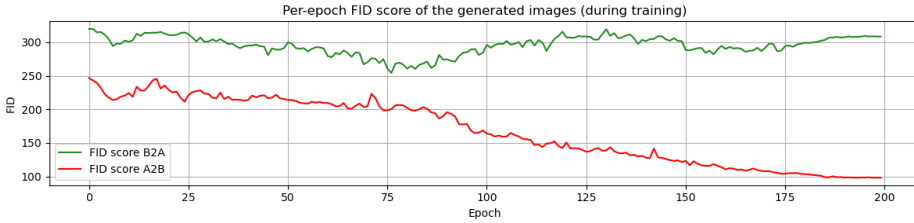
(a) Model 1 (*above*): per-epoch FID score of all the generated samples $\hat{b} = G_{A \rightarrow B}(a)$ (red) and $\hat{a} = G_{B \rightarrow A}(b)$ (green)



(b) Model 2 (*above*): per-epoch FID score of all the generated samples $\hat{b} = G_{A \rightarrow B}(a)$ (red) and $\hat{a} = G_{B \rightarrow A}(b)$ (green)



(c) Model 3 (*above*): per-epoch FID score of all the generated samples $\hat{b} = G_{A \rightarrow B}(a)$ (red) and $\hat{a} = G_{B \rightarrow A}(b)$ (green)



(d) Model 4 (*above*): per-epoch FID score of all the generated samples $\hat{b} = G_{A \rightarrow B}(a)$ (red) and $\hat{a} = G_{B \rightarrow A}(b)$ (green)

Figure 6.6: Plots of the per-epoch FID score of the the generated depth maps $\hat{a} = G_{B \rightarrow A}(b)$ (green) and generated stereo image-pairs $\hat{b} = G_{A \rightarrow B}(a)$ (red) during training of, respectively: (a) model 1; (b) model 2; (c) model 3 and (d) model 4

back in the per-epoch FID evaluation plots. Both models show an initial decrease of FID score during the first 20 epochs, after which it no further improves and starts to oscillate. In contrast, the FID scores of **Model 2** and **model 4** slowly decrease for the first, respectively, 50 and 75 epochs. After that point the pace at which it decreases picks up slightly and continues to drop for the remainder of the training.

The FID score of the other translation $\hat{a} = G_{B \rightarrow A}(b)$ (green) doesn't really drop in any of the models over all iterations. It is however not surprising, since it has to reconstruct a colourful- and detailed stereo image-pair from a depth map holding relatively little information about the original scene.

6.4 Testing the models

After training the four models have been tested on a separate test dataset, that contains non-synthesized stereo image-pairs. The results have been evaluated using the FID score and RMSE loss, and can be found in Table 6.3 and visualized in Figure 6.7. After finding the best performing model, it will be tested using the constructed datasets containing synthesized stereo image-pairs, this is demonstrated in Section 6.5

	FID_A	FID_B	$RMSE_A$	$RMSE_B$
Model 1	377.7	206.1	1.329	0.866
Model 2	391.7	180.0	1.294	0.867
Model 3	375.2	263.2	1.278	0.885
Model 4	353.2	156.6	1.275	0.846

Table 6.3: The FID scores and RMSE losses of four models on the test dataset containing non-synthesized stereo image-pairs

It can be observed that the FID scores of **model 1** and **model 3** is relatively high, suggesting that the similarity between the generated depth maps and the actual depth maps is rather low. **Model 2** and **model 4** on the other hand perform significantly better, because their FID score is much lower and therefore: the distribution of the generated samples resembles the real distribution much better. Unfortunately, the RMSE value proves itself a less usable evaluation metric for our research, because all the values are very close to each other. Nonetheless, model 4 is to be the best performing model.

6.4.1 Visual inspection of the estimated depth maps per model

Ultimately, it is important that the generated output looks realistic and resembles the original distribution of the real depth maps. In Figure 6.8 snapshots of the generated depth maps by each model are shown. Having a low FID score doesn't always ensures a better quality of samples. For instance: model 1 has a lower FID than model 3, but visual inspection of the data learns us that model 1 collapsed (much heavier than

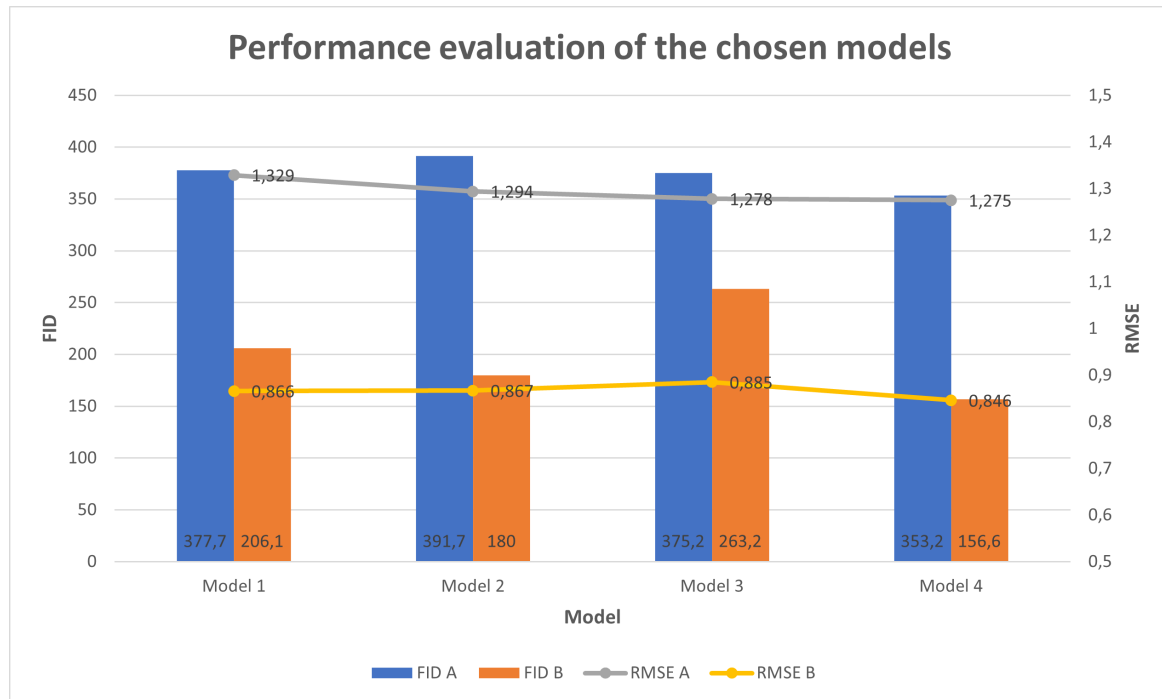


Figure 6.7: Performance evaluation of the four models using the FID score and RMSE loss

model 3).

In contrast **model 2** and **model 4** seem to perform quite well. In this case, the models didn't fail and therefore it checks out that: the lower the FID score, the better the generated output is.

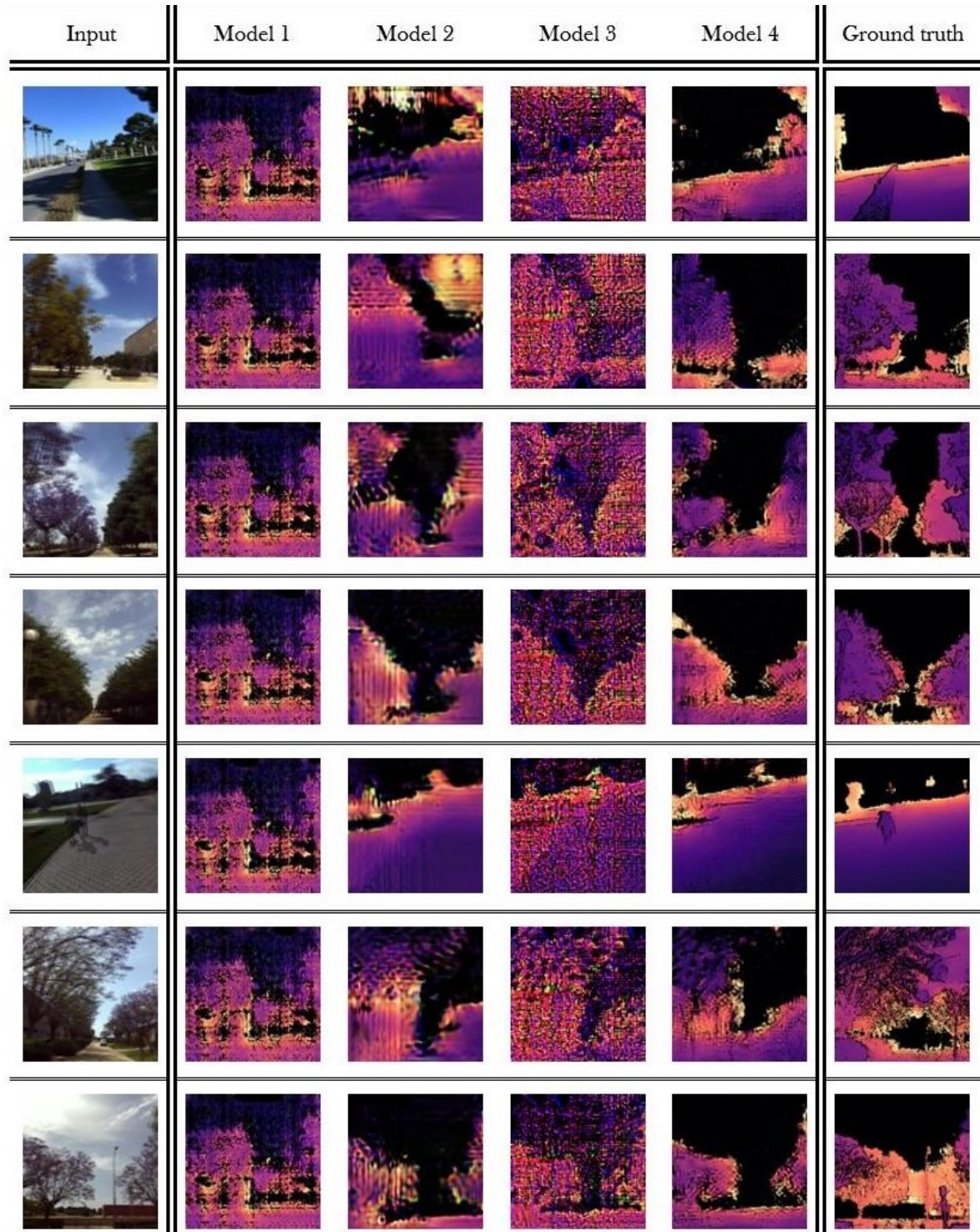


Figure 6.8: Snapshots of generated outputs of the models compared to the ground truth, given its stereo image-pair input

6.5 Testing the final model on synthesized stereo image-pairs

According to Table 6.3, model 4 is the best performing model and will therefore be used to test the influence of synthesizing stereo image-pairs.

Synthesized	FID_A	FID_B	$RMSE_A$	$RMSE_B$
None	353.2	156.6	1.275	0.846
Left	322.1	153.1	1.209	0.841
Right	327.9	156.1	1.215	0.840
Both	329.5	160.8	1.239	0.841

Table 6.4: The FID scores and RMSE losses of the outputs that are generated based on the constructed datasets (see Section 6.1.3) containing synthesized stereo views of domains A and B

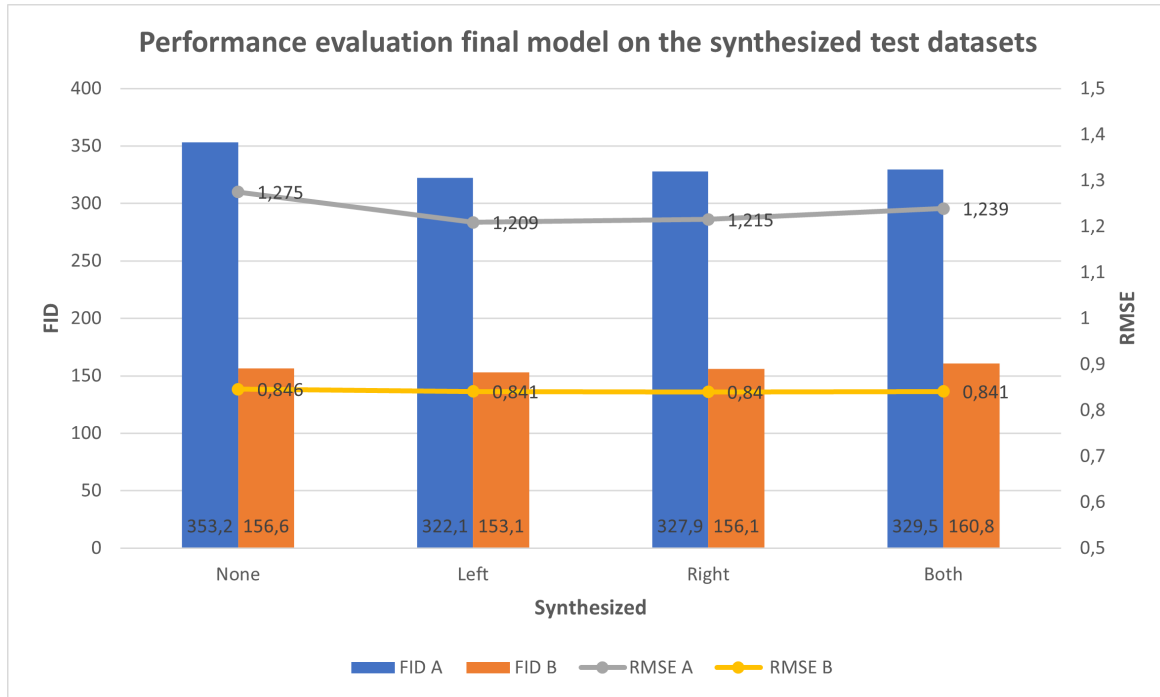


Figure 6.9: Plots of the FID scores (bars) and RMSE losses (lines) for each (non-) synthesized test set, as is shown in table 6.4

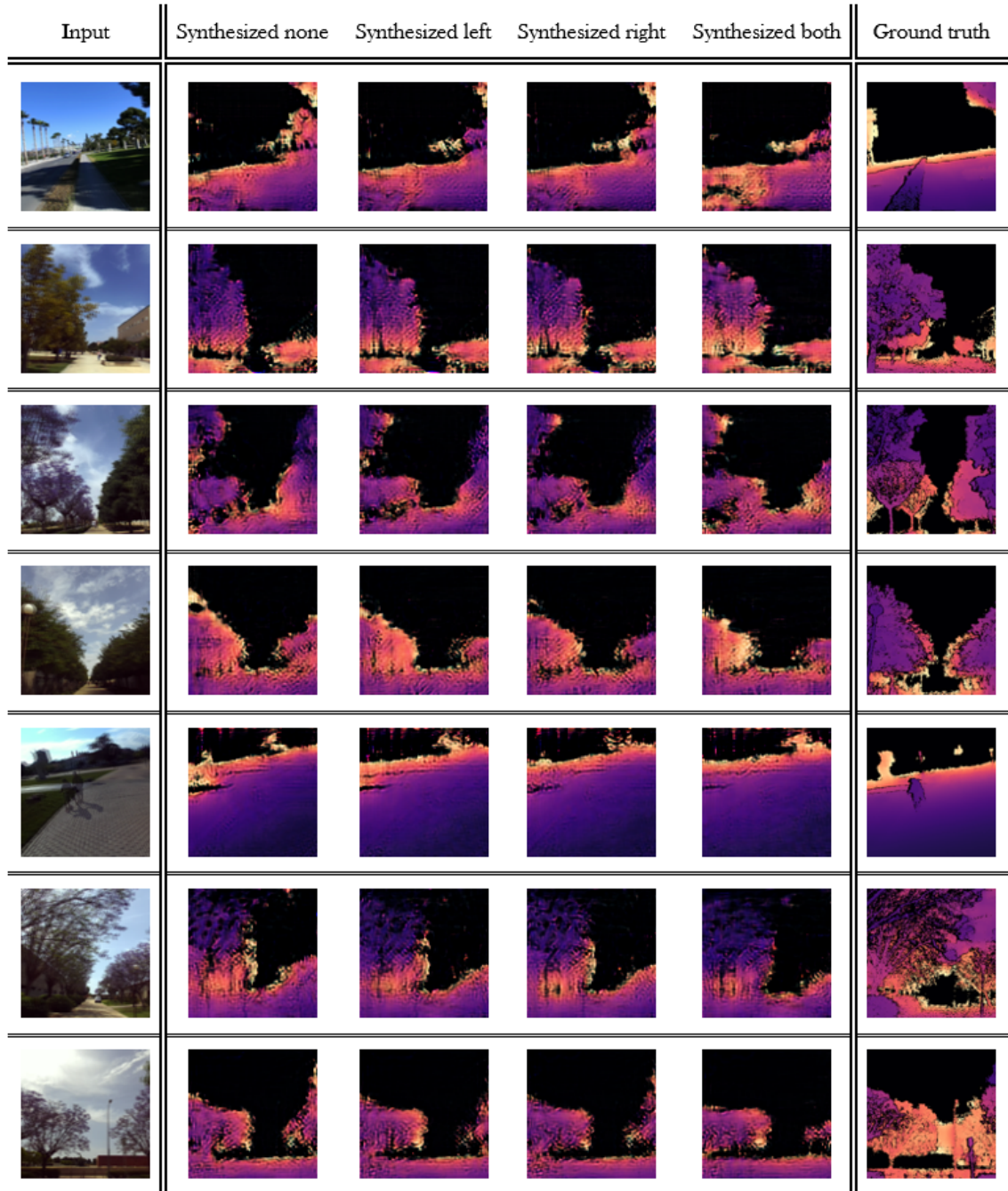


Figure 6.10: Visual inspection of the generated samples using model 4 on the test dataset, where, respectively: no views, left-view, right-view or both views are synthesized

Discussion

In this chapter the results of this dissertation are discussed, but there is a priori a foreword that accounts to the literature review results. Since to me, the results of the literature review contribute to the overall results as well.

7.1 Discussion of the literature review

Prior to synthesizing a stereo image-pair given a single-view image, where after a generative adversarial network estimates the depth given that synthesized stereo view, a lot of research has been done. The topic of this research is rather novel and a similar setup of this research has not been observed in the review literature. Working on novel techniques requires a lot of knowledge on this topic, both broad and specific. All this knowledge about a topic could be placed in a hypothetical pyramid, where conventional- and traditional knowledge is placed at the bottom, whereas specific- and novel knowledge is placed at the top.

My philosophy was, that in order to build upon the top of this hypothetical pyramid of knowledge, you have to have an understanding of what's underneath. The research is structured according this type of reasoning and philosophy. In Chapter 2 the necessary background information is presented to the reader, but it also served an educational purpose for me. In order to grasp what a neural network is doing, while making a depth estimation, I have to have a sense of what geometry is hidden underneath. With that geometrical knowledge, you get to understand how the traditional (hardware) systems- and sensors measure distance, and how ground truth depth data is gathered.

In a similar manner, I attempted to build a profound knowledge base on neural networks and deep learning in general: by starting at the bottom of our hypothetical pyramid of knowledge. In Chapter 3 the development of convolutional neural networks

up until this point has been reviewed. Starting with unsupervised neural networks and the state-of-the-art at that time, and how they gradually evolved into non-end-to-end and eventually end-to-end networks. Discussing the state-of-the-art for every category and weighing their pro's and con's. The research up to this point has built a solid understanding of CNNs, which essentially are one of the two core principles of GANs. The other of course being the method of training it.

The findings of this large (literature) review (together with coding along) have learned me: (1) how to convolutional neural networks function at an architectural level, allowing me to adjust and improve their architecture and their development over time; (2) how generative adversarial networks really work on every level allowing me to tweak on both an architectural level and a higher level, i.e. training methods and working with the loss functions. This in-depth knowledge and understand of how the theory relates to actual lines of code, enabled me later on to build my model from scratch.

7.1.1 Concrete results

Apart from a lot of knowledge, the concrete results of my literature review showed me that most GANs perform an noise-to-image translation task, because they have to generate something. More recent GANs are able to perform an image-to-image task. The most popular models are at the time of writing Pix2Pix and CycleGAN. Since those type of models were, at the time of writing, one of the few promising models that were able to perform image-to-image translation tasks, it was clear which direction I had to go.

In terms of training time and operation speed, modern-day models require vast amounts of processing power in order to be trained at a reasonable pace. However, with a solid graphics card in your computer you can get along fine, during development. Cloud-based machine learning is also a good, but expensive solution once you have to deal with enormous amounts of data. In contrast, the pre-trained models can process data most of the time at speeds that are (near) real-time and are therefore very applicable to real-time applications.

7.2 Discussion of the research results

Initial evaluation of the experiments that have been conducted in Chapter 5 paved the road ahead. During development of the model and through research, there seemed to

be three main aspects that influenced the quality of the results: (1) the identity loss function; (2) the network architecture of both the generator and the discriminator; and (3) the training techniques. The influences of each parameter have been tested in a setup in which everything is the same, but that parameter.

The findings have learned me that (a) identity loss only benefits a GAN *if* the original colours need to be preserved in the result; (b) k-nearest upsampling and the Leaky ReLU activation function seem to improve the quality of the estimation depth map, as opposed to the original method of the original model using Transposed convolutions and the ReLU activation function. Dropout in the discriminator doesn't seem to improve the process, it rather slows down the learning pace of the generator. And finally (c) restricting the discriminator's ability to learn and therefore giving the generator more time to learn. It seemed to benefit the quality of the output distribution in terms of FID score, but visually the result seemed to become worse.

*In regard to **sub-question 1**, about the quality of the synthesized views.* In Section 6.1 two models have been compared to determine which model is the best for synthesize stereo image-pairs. It did not need to take long that the original model architecture of the CycleGAN was superior to a model based on a Leaky ReLU activation function and KNN upsampling. Visual inspection of the result showed us that, but the FID score as well.

*In regard to **sub-question 2**, about the similarity between the generated depth map and the (non-)synthesized stereo image-pairs.* The depth estimation network, the four most promising models from the experiments in Chapter 5 have been selected and trained for much longer. One of those models was the original CycleGAN model. Very early on it was clear that the discriminator model for this application was way too powerful. The models (1 and 3) that did not have any discriminator learning restrictions collapsed far before their training was complete. The other two models (2 and 4), where the discriminator was restricted, showed much more promising results. The FID score kept decreasing over time and the and the losses of the generator and discriminator did not completely diverge right from the start. In both models however, the generator loss starts to increase at some point. Generally this indicates that the discriminator has become too strong and the generator starts to produce other outputs in order to remedy this. If the generator is not able to remedy this, the mode will collapse. However, in both models the generator loss stabilizes again and even slightly seems to drop or start dropping.

Testing the most successful model on the four test datasets that have been created,

using the stereo image-pair synthesis network, showed some surprising results (Section 6.5). Namely that, when the depth is estimation of a stereo image-pair that contains one synthesized view, the depth estimation is more accurate (i.e. higher similarity with the ground truth) than when it has no synthesized views in the stereo image-pair. The best results are achieved when a left-view image is synthesized from the right-view image. However, using two synthesized views in the stereo image-pair, produces results that are worse than using when two non-synthesized views are used. This makes me wonder on which information the network is making its prediction. Since, the best results are achieved when there's one realistic view and one synthesized view. This looks to me that the network is mostly making its prediction based on the information from the realistic part of the stereo view.

*In regard to **sub-question 3**, about the performance of the depth estimation network in regard to the quality of the ground truth depth map.* Another noteworthy result is that the GAN has been trained on a small dataset in which the ground truth data has been estimated using an existing depth estimation network. It turned out that the generator was able to produce better results than the ground truth actually was. One of the best examples is shown in Figure 6.10, in the bottom two images. In the ground truth depth, the sky is orange or purple, which estimates it to be nearby. The estimated depth map does not reproduce the same results, but estimates those areas to be too far away to measure. This is actually more true than the ground truth image, because the sky is indeed too far away to measure.

The point of this was to test whether GANs are able to learn in a similar fashion as humans: show how something is done for given number of times, while some of the times this is done incorrectly. The idea is that humans learn to understand that if something is done most of the time correctly, but sometimes incorrect, they learn to overcome those flaws. This is due to the way indirect way of learning: the generator tries to fool the discriminator, rather than the generator trying to minimize the distance to a given image.

*In regard to **sub-question 4 and 5**, about the accuracy around ill-posed regions and edges, and its ability to generalize well on unseen data.* The test set that has been used is unseen data to the network, but the scenes do have similarities to the training data, since they are extracted from the same enormous dataset. It was very difficult to find new stereo data with ground truth depth maps that resemble the ones used in this research. Most datasets either did not contain stereo image-pairs or contained ground truth depth map that were either point clouds (a lot more black and some points at which the laser came back into the sensor) or were post-processing very differently.

This major gap between datasets and how depth is visualized made it very difficult to compare and evaluate produced results. Nonetheless, given that the test dataset is still unseen data to the network it performed reasonable well.

It's performance to accurately estimate depth around ill-posed regions and edges seemed pretty poor. Performing a visual inspection of the generated samples quickly made clear that one of the weaknesses of the model was to accurately predict depth around ill-posed regions, edges and finer details. I assume that this is mostly due to the fact that the network received relatively little training on a relatively small dataset. It turned out that a much more training using much more training data, is definitely desired to improved the results. Despite that, the network demonstrated very well that, even with little amounts of training data, it is able to learn and filter out the flaws that exist in the ground truth data. So, according to my own opinion, the results show promising results for the application of GANs in depth estimation, but certainly require much more training to reach their full potential.

Conclusions and recommendations

In this chapter the overall conclusions of this dissertation are presented, along with the recommendations for future work.

8.1 Conclusions

The conclusion of this research is that existing depth estimation models can certainly be improved by the incorporation of generative adversarial networks, due to their indirect way of learning. The current depth estimation methods are mostly based on training large models on accurate ground truth data and are therefore not widely applicable in many real-world scenarios. Synthesizing stereo image-pairs from single-view images seem like a plausible solution to improve the availability of training data for depth estimation networks based on stereo image-pairs. In this research we could not entirely proof that this method is actually able to more accurately predict depth as compared to the current state-of-the-art. Nonetheless, we could demonstrate that it is able for a GAN to learn to overcome the flaws that exist in the estimated depth maps of those networks. Therefore, showing its potential for complex learning tasks such as depth estimation from stereo image-pairs.

The aim of this research was to explore the possibilities with generative adversarial networks in relation to stereo image synthesis and depth estimation, and how this could be an improvement to the existing depth estimation networks. This research presented: (a) a semi-novel method to synthesize stereo image-pairs from a single-view image; and (b) demonstrated that depth estimation on stereo image-pairs using GANs *can* not only improve the quality of existing depth estimation networks, but also overcome the flaws that exist in the predictions of those networks.

The first part of this research, the stereo image-pair synthesis, seemed to work

very well. The model responsible for stereo image-pair synthesis, was successful in synthesizing a left- or right-view from a single-view image, creating a stereo image-pair. Although it was hard to visualize in the report, the model was able to rotate and translate different objects in the scene in the correct manner (see Section 6.1.4, or follow: [link 1](#) (real) and [link 2](#) (synthesized)). Using traditional synthesis methods this was very hard to do, since individual objects exist at different depths in the scene and therefore have a different rotation and translation when the viewer's perspective is changed.

Learning a GAN to estimate depth on a given stereo image-pair was a much harder task, although it succeeded in some parts. It has been trained on ground truth data that contained flaws, since it was not captured using expensive sensors, but predicted using another depth estimation network. The entire point was to prove that GANs can learn to overcome the flaws that exist in their training data, which we were able to prove. Running the model on the test sets showed that for some samples the GAN could make a more sensible estimation than was depicted in the ground truth. The best example of this is that the ground truth depth contained information that stated that the air was estimated to be nearby, while in reality it is too far away to measure. Our model was able to estimate this correctly, while also being correct in its estimation about the other objects in the scene. This demonstrates the potential of GANs being trained on ground truth data that has been acquired using software and in doing so, it could replace a lot of expensive hardware that was usually required to gather ground truth data.

Our model did perform moderately in terms of its accuracy and precision, especially at the regions that contain fine details. In deep learning the expression "bigger is better" is usually true, so I think it is safe to assume that providing much more training time and training data to our model will enable it to become much more accurate and precise. Potentially even to become better at the current, traditional neural networks for depth estimation. This however, is speculation and a something for future work.

8.2 Recommendations and future work

Despite hopeful prospects at the start of this research, given the new ideas and approaches, with respect to the state-of-the-art at that time. The major bottleneck turned out to be the lack of computational resources to train the models. Despite the fact those resources exist, I should have intervened earlier and changed my course of action. Instead of continuing with developing- and training the models locally, I should have shifted this to an environment that provided more computational resources. Un-

fortunately, this also came with learning how to work in those new environments (like Microsoft Azure or Google AI) and temporarily stopping with the work I was currently working on. Making this change in time, would have enabled me to train models much quicker and therefore allow me to conduct more- and/or better experiments.

Another, yet smaller, bottleneck in this research was the use of ground truth depth maps that are not always as accurate as depth maps captured with real sensors. One could argue that this causes the model to never fully learn the complex task of depth estimation. On the one hand you could say that: it learns to improve upon the flawed ground truth depth maps, but on the other hand you could say that it will never truly learn it - because it has never been able to see the absolute and correct depth data. Therefore the other recommendation that I have for future work is that you find a solution for this issue. Possible solution are to mix flawed data into the real dataset or distort the real dataset in some way.

Bibliography

- [1] V. Nityananda and J. C. Read, “Stereopsis in animals: Evolution, function and mechanisms,” *Journal of Experimental Biology*, vol. 220, no. 14, pp. 2502–2512, 2017.
- [2] N. Smolyanskiy, A. Kamenev, and S. Birchfield, “On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2018-June, pp. 1120–1128, 2018.
- [3] F. Khan, S. Salahuddin, and H. Javidnia, “Deep learning-based monocular depth estimation methods—a state-of-the-art review,” *Sensors (Switzerland)*, vol. 20, no. 8, pp. 1–16, 2020.
- [4] D. Scharstein, R. Szeliski, and R. Zabih, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *Proceedings - IEEE Workshop on Stereo and Multi-Baseline Vision, SMBV 2001*, no. 1, pp. 131–140, 2001.
- [5] C. Godard, O. M. Aodha, M. Firman, and G. Brostow, “Digging into self-supervised monocular depth estimation,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, no. 1, pp. 3827–3837, 2019.
- [6] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 2650–2658, 2015.
- [7] R. Chen, F. Mahmood, A. Yuille, and N. J. Durr, “Rethinking Monocular Depth Estimation with Adversarial Training,” 2018. [Online]. Available: <http://arxiv.org/abs/1808.07528>
- [8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in Neural Information Processing Systems*, vol. 3, no. January, pp. 2672–2680, 2014.

- [9] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pp. 1–16, 2016.
- [10] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5967–5976, 2017.
- [11] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 4396–4405, 2019.
- [12] Y. Wang, Z. Lai, G. Huang, B. H. Wang, L. Van Der Maaten, M. Campbell, and K. Q. Weinberger, “Anytime stereo image depth estimation on mobile devices,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 5893–5900, 2019.
- [13] Y. Chen, W. Li, X. Chen, and L. Van Gool, “Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 1841–1850, 2019.
- [14] A. Atapour-Abarghouei and T. P. Breckon, “Real-Time Monocular Depth Estimation Using Synthetic Data with Domain Adaptation via Image Style Transfer,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2800–2810, 2018.
- [15] K. Wang, L. Lin, C. Jiang, C. Qian, and P. Wei, “3D Human Pose Machines with Self-Supervised Learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 5, pp. 1069–1082, 2020.
- [16] Y. Chen, F. Wu, Z. Wang, Y. Song, Y. Ling, and L. Bao, “Self-Supervised Learning of Detailed 3D Face Reconstruction,” *IEEE Transactions on Image Processing*, vol. 29, pp. 8696–8705, 2020.
- [17] A. Atapour-Abarghouei and T. P. Breckon, “To Complete or to Estimate, That is the Question: A Multi-Task Approach to Depth Completion and Monocular Depth Estimation,” *Proceedings - 2019 International Conference on 3D Vision, 3DV 2019*, pp. 183–193, 2019.

- [18] H. Dharmo, K. Tateno, I. Laina, N. Navab, and F. Tombari, "Peeking behind objects: Layered depth prediction from a single image," *Pattern Recognition Letters*, vol. 125, pp. 333–340, 2019. [Online]. Available: <https://doi.org/10.1016/j.patrec.2019.05.007>
- [19] W. Y. San, T. Zhang, S. Chen, A. Wiliem, D. Stefanelli, and B. C. Lovell, "Early Experience of Depth Estimation on Intricate Objects using Generative Adversarial Networks," *2018 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2018*, no. 1, 2019.
- [20] S. Kim, S. Member, D. Min, S. Member, S. Kim, K. Sohn, and S. Member, "Unified Confidence Estimation Networks for Robust Stereo Matching," vol. 28, no. 3, pp. 1299–1313, 2019.
- [21] L. C. Moreno, "Automated Privacy-Preserving Video Processing through Anonymized 3D Scene Reconstruction," 2019.
- [22] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Y. L. Tian, A. Ekin, J. Connell, C. F. Shu, and M. Lu, "Enabling video privacy through computer vision," *IEEE Security and Privacy*, vol. 3, no. 3, pp. 50–57, 2005.
- [23] S. Ribaric, A. Ariyaeinia, and N. Pavesic, "De-identification for privacy protection in multimedia content: A survey," *Signal Processing: Image Communication*, vol. 47, pp. 131–151, 2016.
- [24] C. Liu, L. Song, J. Zhang, K. Chen, and J. Xu, "Self-Supervised Learning for Specified Latent Representation," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 1, pp. 47–59, 2020.
- [25] M. Hermann, B. Ruf, M. Weinmann, and S. Hinz, "Self-Supervised Learning for Monocular Depth Estimation from Aerial Imagery," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 5, no. 2, pp. 357–364, 2020.
- [26] Q. Guo and Z. Wang, "A Self-Supervised Learning Framework for Road Center-line Extraction from High-Resolution Remote Sensing Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4451–4461, 2020.
- [27] L. Huynh, P. Nguyen-Ha, J. Matas, E. Rahtu, and J. Heikkila, "Guiding Monocular Depth Estimation Using Depth-Attention Volume," vol. L, 2020. [Online]. Available: <http://arxiv.org/abs/2004.02760>

- [28] S. Li, F. Xue, X. Wang, Z. Yan, and H. Zha, “Sequential adversarial learning for self-supervised deep visual odometry,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, no. Iccv, pp. 2851–2860, 2019.
- [29] European Union, “General Data Protection Regulation,” 2016. [Online]. Available: <https://gdpr-info.eu/>
- [30] B. Generative, A. Networks, M. Gan, P. C. Analy, N. Learning, G. A. Networks, L.-s. Gan, T. Gans, J. Shannon, J. Shannon, and W. Gan, “Understanding GANs: the LGQ setting,” pp. 1–13, 2018.
- [31] I. Gemp and S. Mahadevan, “Global convergence to the equilibrium of gans using variational inequalities,” *arXiv*, 2018.
- [32] S. Liu and K. Chaudhuri, “The inductive bias of restricted f-GANs,” *arXiv*, 2018.
- [33] P. Grnarova, K. Y. Levy, A. Lucchi, T. Hofmann, and A. Krause, “An online learning approach to generative adversarial networks,” *arXiv*, pp. 1–27, 2017.
- [34] F. A. Oliehoek, R. Savani, J. Gallego-Posada, E. Van Der Pol, E. D. De Jong, and R. Groß, “GANs: Generative adversarial network games,” *arXiv*, 2017.
- [35] F. A. Oliehoek, R. Savani, J. Gallego, E. van der Pol, and R. Groß, “Beyond Local Nash Equilibria for Adversarial Networks,” *Communications in Computer and Information Science*, vol. 1021, pp. 73–89, 2019.
- [36] P. Revuelta, B. Ruiz, and J. M. Snchez Pe, “Depth Estimation - An Introduction,” *Current Advancements in Stereo Vision*, 2012.
- [37] M. Bleyer, “Segmentation-based stereo and motion with occlusions,” 2006.
- [38] R. Y. Tsai, “A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses,” *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.
- [39] H. Hirschmüller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.
- [40] L. Zhang and S. M. Seitz, “Estimating optimal parameters for MRF stereo from a single image pair,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 331–342, 2007.
- [41] F. Tombari, S. Mattoccia, L. D. Stefano, and E. Addimanda, “Classification and evaluation of cost aggregation methods for stereo correspondence,” *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008.

- [42] H. Hattori and A. Maki, "Stereo Matching with Direct Surface Orientation Recovery," pp. 36.1–36.11, 2013.
- [43] R. D. Arnold, "Automated Stereo Perception," no. March, 1983.
- [44] K. J. Yoon and I. S. Kweon, "Stereo matching with symmetric cost functions," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, no. February, pp. 2371–2377, 2006.
- [45] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister, "High-quality real-time stereo using adaptive cost aggregation and dynamic programming," *Proceedings - Third International Symposium on 3D Data Processing, Visualization, and Transmission, 3DPVT 2006*, no. May, pp. 798–805, 2006.
- [46] S. Mattoccia, F. Tombari, and L. D. Stefano, "Localization Within a Scanline Optimization Framework," pp. 517–527, 2007.
- [47] S. A. Adhyapak, "Stereo matching via selective multiple windows," *Journal of Electronic Imaging*, vol. 16, no. 1, p. 013012, 2007.
- [48] O. Veksler, "Fast variable window for stereo correspondence using integral images," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2003.
- [49] P. R. Innocent, H. Hirschmuller, and J. M. Garibaldi, "Real-time correlation - based stereo vision with reduced error borders," *International Journal of Computer Vision*, vol. 47, pp. 229–246, 2002.
- [50] Y. Boykov, O. Veksler, and R. I. Zabi, "A variable window approach to early vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1283–1294, 1998.
- [51] H. Hirschmüller and D. Scharstein, "Evaluation of cost functions for stereo matching," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.
- [52] J. Jiao, R. Wang, W. Wang, S. Dong, Z. Wang, and W. Gao, "Local stereo matching with improved matching cost and disparity refinement," *IEEE Multimedia*, vol. 21, no. 4, pp. 16–27, 2014.
- [53] A. Shaked and L. Wolf, "Improved stereo matching with constant highway networks and reflective confidence learning," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, no. vi, pp. 6901–6910, 2017.

- [54] A. Seki, M. Pollefeys, T. Corporation, E. T. Zürich, and Microsoft, “SGM-Nets: Semi-global matching with neural networks,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, no. 1, pp. 6640–6649, 2017.
- [55] J. Žbontar and Y. Lecun, “Stereo matching by training a convolutional neural network to compare image patches,” *Journal of Machine Learning Research*, vol. 17, pp. 1–32, 2016.
- [56] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, “On Building an Accurate Stereo Matching System on Graphics Hardware,” *In Computer Vision Workshops (ICCVWorkshops)*, 2011.
- [57] S. Pillai, R. Ambruş, and A. Gaidon, “SuperDepth: Self-supervised, super-resolved monocular depth estimation,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, no. Figure 1, pp. 9250–9256, 2019.
- [58] R. Ji, K. Li, Y. Wang, X. Sun, F. Guo, X. Guo, Y. Wu, F. Huang, and J. Luo, “Semi-Supervised Adversarial Monocular Depth Estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2410–2422, 2020.
- [59] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, “Monocular Depth Estimation Using Multi-Scale Continuous CRFs as Sequential Deep Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 6, pp. 1426–1440, 2019.
- [60] E. Delage, H. Lee, and A. Y. Ng, “A dynamic Bayesian network model for autonomous 3d reconstruction from a single indoor image,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2418–2428, 2006.
- [61] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille, “Towards unified depth and semantic prediction from a single image,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 2800–2809, 2015.
- [62] T. Shen, L. Zhou, Z. Luo, Y. Yao, S. Li, J. Zhang, T. Fang, and L. Quan, “Self-supervised learning of depth and motion under photometric inconsistency,” *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, pp. 4044–4053, 2019.

- [63] Godard, “Godard 2017,” *Cvpr*, pp. 270–279, 2017. [Online]. Available: <http://visual.cs.ucl.ac.uk/pubs/monoDepth/>
- [64] Q. Shu, S. Liu, J. Wang, Q. Lai, and Z. Zhou, “Image Classification Algorithm Named OCFC Based on Self-supervised Learning,” *Proceedings of 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference, ITOEC 2020*, no. Itoec, pp. 589–594, 2020.
- [65] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14, 2015.
- [66] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 1–9, 2015.
- [67] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, 2016.
- [68] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1800–1807, 2017.
- [69] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5987–5995, 2017.
- [70] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017.
- [71] 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,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 3213–3223, 2016.
- [72] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi, and B. Zhou, “DrivingStereo,” pp. 899–908, 2019. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/papers/Yang_DrivingStereo_A_Large-Scale_Dataset_for_Stereo_Matching_in_Autonomous_Driving_CVPR_2019_paper.pdf

- [73] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research,” *The International Journal of Robotics Research*, no. October, pp. 1–6, 2013.
- [74] D. Scharstein and R. Szeliski, “High-accuracy stereo depth maps using structured light,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, no. June, pp. 195–202, 2003.
- [75] P. H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J. B. Huang, “DeepMVS: Learning Multi-view Stereopsis,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2821–2830, 2018.
- [76] J. Cho, D. Min, Y. Kim, and K. Sohn, “A Large RGB-D Dataset for Semi-supervised Monocular Depth Estimation,” pp. 1–13, 2019. [Online]. Available: <http://arxiv.org/abs/1904.10230>
- [77] I. Vasiljevic, N. Kolkin, S. Zhang, R. Luo, H. Wang, F. Z. Dai, A. F. Daniele, M. Mostajabi, S. Basart, M. R. Walter, and G. Shakhnarovich, “DIODE: A Dense Indoor and Outdoor DEpth Dataset,” 2019. [Online]. Available: <http://arxiv.org/abs/1908.00463>
- [78] Z. Bauer, F. Gomez-Donoso, E. Cruz, S. Orts-Escolano, and M. Cazorla, “UASOL, a large-scale high-resolution outdoor stereo dataset,” *Scientific Data*, vol. 6, no. 1, pp. 1–15, 2019. [Online]. Available: <http://dx.doi.org/10.1038/s41597-019-0168-5>
- [79] J. Watson, O. Mac Aodha, D. Turmukhambetov, G. J. Brostow, and M. Firman, “Learning Stereo from Single Images,” pp. 1–32, 2020. [Online]. Available: <http://arxiv.org/abs/2008.01484>
- [80] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li, “Group-wise correlation stereo network,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 3268–3277, 2019.
- [81] R. Qin, X. Huang, W. Liu, and C. Xiao, “Pairwise Stereo Image Disparity and Semantics Estimation with the Combination of U-Net and Pyramid Stereo Matching Network,” pp. 4971–4974, 2019.
- [82] Z. Yin, T. Darrell, and F. Yu, “Hierarchical Discrete Distribution Decomposition for Match Density Estimation,” *arXiv*, 2018.
- [83] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, “GA-net: Guided aggregation net for end-to-end stereo matching,” *Proceedings of the IEEE Computer Society*

- Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 185–194, 2019.
- [84] A. C. Kumar, S. M. Bhandarkar, and M. Prasad, “Monocular depth prediction using generative adversarial networks,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2018-June, pp. 413–421, 2018.
- [85] F. Tosi, F. Aleotti, M. Poggi, and S. Mattoccia, “Learning monocular depth estimation infusing traditional stereo knowledge,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 9791–9801, 2019.
- [86] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, “A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 4040–4048, 2016.
- [87] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie, “Learning from noisy large-scale datasets with minimal supervision,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6575–6583, 2017.
- [88] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, “Learning from massive noisy labeled data for image classification,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 2691–2699, 2015.
- [89] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, “Revisiting Unreasonable Effectiveness of Data in Deep Learning Era,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 843–852, 2017.
- [90] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten, “Exploring the Limits of Weakly Supervised Pretraining,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11206 LNCS, pp. 185–201, 2018.
- [91] J. L. Schönberger, S. N. Sinha, and M. Pollefeys, “Learning to fuse proposals from multiple scanline optimizations in semi-global matching,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11217 LNCS, pp. 758–775, 2018.

- [92] Q. Yang, “A non-local cost aggregation method for stereo matching,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1402–1409, 2012.
- [93] F. Zhang, L. Dai, S. Xiang, and X. Zhang, “Segment graph based image filtering: Fast structure-preserving smoothing,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, no. d, pp. 361–369, 2015.
- [94] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” *Advances in Neural Information Processing Systems*, vol. 3, no. January, pp. 2366–2374, 2014.
- [95] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, “End-to-End Learning of Geometry and Context for Deep Stereo Regression,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 66–75, 2017.
- [96] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6612–6621, 2017.
- [97] W. Luo, A. G. Schwing, and R. Urtasun, “Efficient deep learning for stereo matching,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 5695–5703, 2016.
- [98] K. Zhou, X. Meng, and B. Cheng, “Review of Stereo Matching Algorithms Based on Deep Learning,” *Computational Intelligence and Neuroscience*, vol. 2020, no. 1, 2020.
- [99] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, “Deep stereo: Learning to predict new views from the world’s imagery,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 5515–5524, 2016.
- [100] J. Xie, R. Girshick, and A. Farhadi, “Deep3D: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9908 LNCS, pp. 842–857, 2016.
- [101] R. Garg, B. G. Vijay Kumar, G. Carneiro, and I. Reid, “Unsupervised CNN for single view depth estimation: Geometry to the rescue,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9912 LNCS, pp. 740–756, 2016.

- [102] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6602–6611, 2017.
- [103] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha, “Unsupervised deep learning for optical flow estimation,” *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, no. Hollingworth 2004, pp. 1495–1501, 2017.
- [104] Z. Yin and J. Shi, “GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1983–1992, 2018. [Online]. Available: [geonet:UnsupervisedLearningofDenseDepth, OpticalFlowandCameraPosehttp://arxiv.org/abs/1803.02276v2](http://arxiv.org/abs/1803.02276v2)
- [105] J. Žbontar and Y. Le Cun, “Computing the stereo matching cost with a convolutional neural network,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, no. 1, pp. 1592–1599, 2015.
- [106] T. Taniai, Y. Matsushita, Y. Sato, and T. Naemura, “Continuous 3D Label Stereo Matching Using Local Expansion Moves,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 11, pp. 2725–2739, 2018.
- [107] P. Brandao, E. Mazomenos, and D. Stoyanov, “Widening siamese architectures for stereo matching,” *Pattern Recognition Letters*, vol. 120, pp. 75–81, 2019. [Online]. Available: <https://doi.org/10.1016/j.patrec.2018.12.002>
- [108] I. H. Md Yusof, M. An, and M. H. Barghi, “Integration of lean construction considerations into design process of construction projects,” *Proceedings of the 31st Annual Association of Researchers in Construction Management Conference, ARCOM 2015*, no. i, pp. 885–894, 2015.
- [109] J. Žbontar and Y. Lecun, “Stereo matching by training a convolutional neural network to compare image patches,” *Journal of Machine Learning Research*, vol. 17, pp. 1–32, 2016.
- [110] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang, “A deep visual correspondence embedding model for stereo matching costs,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 972–980, 2015.
- [111] H. Park and K. M. Lee, “Look wider to match image patches with convolutional neural networks,” *arXiv*, vol. 24, no. 12, pp. 1788–1792, 2017.

- [112] P. Knöbelreiter, C. Reinbacher, A. Shekhovtsov, and T. Pock, “End-to-end training of hybrid CNN-CRF models for stereo,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1456–1465, 2017.
- [113] S. Gidaris and N. Komodakis, “Detect, replace, refine: Deep structured prediction for pixel wise labeling,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 7187–7196, 2017.
- [114] F. Güney and A. Geiger, “Displets: Resolving stereo ambiguities using object knowledge,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 4165–4175, 2015.
- [115] A. Dosovitskiy, P. Fischery, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, “FlowNet: Learning optical flow with convolutional networks,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 2758–2766, 2015.
- [116] X. Song, X. Zhao, L. Fang, H. Hu, and Y. Yu, “EdgeStereo: An Effective Multi-task Learning Network for Stereo Matching and Edge Detection,” *International Journal of Computer Vision*, vol. 128, no. 4, pp. 910–930, 2020.
- [117] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “FlowNet 2.0: Evolution of optical flow estimation with deep networks,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1647–1655, 2017.
- [118] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan, “Cascade Residual Learning: A Two-Stage Convolutional Neural Network for Stereo Matching,” *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, vol. 2018-Janua, pp. 878–886, 2017.
- [119] Z. Liang, Y. Feng, Y. Guo, H. Liu, W. Chen, L. Qiao, L. Zhou, and J. Zhang, “Learning for Disparity Estimation Through Feature Constancy,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2811–2820, 2018.
- [120] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia, “SegStereo: Exploiting semantic information for disparity estimation,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11211 LNCS, pp. 660–676, 2018.

- [121] X. Song, X. Zhao, H. Hu, and L. Fang, “EdgeStereo: A Context Integrated Residual Pyramid Network for Stereo Matching,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11365 LNCS, pp. 20–35, 2019.
- [122] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, “End-to-end learning of geometry and context for deep stereo regression,” *arXiv*, 2017.
- [123] J. R. Chang and Y. S. Chen, “Pyramid Stereo Matching Network,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 5410–5418, 2018.
- [124] C. Lu, H. Uchiyama, D. Thomas, A. Shimada, and R. ichiro Taniguchi, “Sparse cost volume for efficient stereo matching,” *Remote Sensing*, vol. 10, no. 11, pp. 1–12, 2018.
- [125] S. Tulyakov, A. Ivanov, and F. Fleuret, “Practical deep stereo (PDS): Toward applications-friendly deep stereo matching,” *Advances in Neural Information Processing Systems*, vol. 2018-Decem, pp. 5871–5881, 2018.
- [126] L. Yu, Y. Wang, Y. Wu, and Y. Jia, “Deep stereo matching with explicit cost aggregation sub-architecture,” *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 7517–7524, 2018.
- [127] R. Slossberg, A. Wetzler, and R. Kimmel, “Deep Stereo Matching with Dense CRF Priors,” 2016. [Online]. Available: <http://arxiv.org/abs/1612.01725>
- [128] Z. Jie, P. Wang, Y. Ling, B. Zhao, Y. Wei, J. Feng, and W. Liu, “Left-Right Comparative Recurrent Model for Stereo Matching,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3838–3846, 2018.
- [129] M. Poggi and S. Mattoccia, “Learning to predict stereo reliability enforcing local consistency of confidence maps,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 4541–4550, 2017.
- [130] A. Paturel, “Game Theory Game Theory,” *Naval War College Review*, vol. 14, no. 5, pp. 16–42, 2014. [Online]. Available: <https://digital-commons.usnwc.edu/nwc-review/vol14/iss5/3>

- [131] 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, 2016.
- [132] M. M. Puscas, D. Xu, A. Pilzer, and N. Sebe, "Structured Coupled Generative Adversarial Networks for Unsupervised Monocular Depth Estimation," *Proceedings - 2019 International Conference on 3D Vision, 3DV 2019*, pp. 18–26, 2019.
- [133] L. Jing and Y. Tian, "Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.
- [134] R. K. Thakur and S. Mukherjee, "A Conditional Adversarial Network for Scene Flow Estimation," *arXiv*, 2019.
- [135] A. Atapour-Abarghouei, S. Akcay, G. Payen de La Garanderie, and T. P. Breckon, "Generative adversarial framework for depth filling via Wasserstein metric, cosine transform and domain transfer," *Pattern Recognition*, vol. 91, pp. 232–244, 2019. [Online]. Available: <https://doi.org/10.1016/j.patcog.2019.02.010>
- [136] Z. Wu, X. Wu, X. Zhang, S. Wang, and L. Ju, "Spatial correspondence with generative adversarial network: Learning depth from monocular videos," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, no. Iccv, pp. 7493–7503, 2019.
- [137] J. Chi, J. Gao, L. Qi, S. Zhang, J. Dong, and H. Yu, "Depth estimation of a single RGB image with semi-supervised two-stage regression," *ACM International Conference Proceeding Series*, pp. 97–102, 2019.
- [138] L. P. Matias, M. Sons, J. R. Souza, D. F. Wolf, and C. Stiller, "VeIGAN: Vectorial inpainting generative adversarial network for depth maps object removal," *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2019-June, no. Iv, pp. 310–316, 2019.
- [139] X. Liu, A. Sinha, M. Ishii, G. D. Hager, A. Reiter, R. H. Taylor, and M. Unberath, "Dense Depth Estimation in Monocular Endoscopy with Self-Supervised Learning Methods," *IEEE Transactions on Medical Imaging*, vol. 39, no. 5, pp. 1438–1447, 2020.
- [140] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

- [141] P. Wang, C. Lin, B. Xu, W. Che, and Q. Wang, “LOW-FREQUENCY GUIDED SELF-SUPERVISED LEARNING FOR HIGH-FIDELITY 3D FACE RECONSTRUCTION IN THE WILD Institute of Automation , Chinese Academy of Sciences , China University of Chinese Academy of Sciences , China,” pp. 1–6, 2020.
- [142] M. Tatarchenko, A. Dosovitskiy, and T. Brox, “Multi-view 3D models from single images with a convolutional network,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9911 LNCS, pp. 322–337, 2016.
- [143] Y. Li, W. Dong, J. Chen, S. Cao, H. Zhou, Y. Zhu, J. Wu, L. Lan, W. Sun, T. Qian, K. Ma, H. Xu, and Y. Zheng, “Efficient and Effective Training of COVID-19 Classification Networks with Self-supervised Dual-track Learning to Rank,” *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 10, pp. 1–1, 2020.
- [144] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, pp. 2242–2251, 2017.
- [145] M. Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” *Advances in Neural Information Processing Systems*, vol. 2017-December, no. Nips, pp. 701–709, 2017.
- [146] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. B. Tenenbaum, “Deep convolutional inverse graphics network,” *Advances in Neural Information Processing Systems*, vol. 2015-Janua, pp. 2539–2547, 2015.
- [147] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, “View synthesis by appearance flow,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9908 LNCS, pp. 286–301, 2016.
- [148] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg, “Transformation-grounded image generation network for novel 3D view synthesis,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 702–711, 2017.
- [149] Y. M. Lo, C. C. Chang, D. L. Way, and Z. C. Shih, “Generation of stereo images based on a view synthesis network,” *Applied Sciences (Switzerland)*, vol. 10, no. 9, pp. 1–15, 2020.

- [150] P. Hedman, S. Alsisan, R. Szeliski, and J. Kopf, “Casual 3D photography,” *ACM Transactions on Graphics*, vol. 36, no. 6, 2017.
- [151] B. Zhao, X. Wu, Z. Q. Cheng, H. Liu, Z. Jie, and J. Feng, “Multi-view image generation from a single-view,” *MM 2018 - Proceedings of the 2018 ACM Multimedia Conference*, pp. 383–391, 2018.
- [152] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, “Learning from simulated and unsupervised images through adversarial training,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 2242–2251, 2017.
- [153] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” *Advances in Neural Information Processing Systems*, vol. 2017-December, no. Nips, pp. 6627–6638, 2017.
- [154] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature,” *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014.
- [155] L. Lu, Y. Shin, Y. Su, and G. E. Karniadakis, “Dying ReLU and initialization: Theory and numerical examples,” *Communications in Computational Physics*, vol. 28, no. 5, pp. 1671–1706, 2020.
- [156] Y. Kinoshita and H. Kiya, “Checkerboard-Artifact-Free Image-Enhancement Network Considering Local and Global Features,” *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2020 - Proceedings*, pp. 1139–1144, 2020.
- [157] H. Guo, “Regularization via Adaptive Pairwise Label Smoothing,” vol. 108, 2020. [Online]. Available: <http://arxiv.org/abs/2012.01559>