MONOCULAR DEPTH ESTIMATION FOR NIGHT-TIME IMAGES

N. Khalefa[1]*, N. El-Sheimy[1]

[1] Dept. of Geomatics Engineering, University of Calgary,
2500 University Dr. N.W. Calgary, Alberta, Canada T2N 1N4 - (nour.khalefa, elsheimy)@ucalgary.ca

KEY WORDS: Monocular Depth Estimation, Image Translation, Generative Adversarial Network, Synthetic Data, Night.

ABSTRACT:

Depth estimation plays a pivotal role in numerous computer vision applications. However, depth estimation networks trained exclusively on daytime images tend to yield poor performance when applied to nighttime scenarios due to domain differences and variations in scene characteristics. In order to address this limitation, we conducted experiments involving the creation of a synthetic nighttime dataset by employing image translation techniques through a generative network. Subsequently, we utilized the generated images to fine-tune the depth estimation network, aiming to investigate the potential for enhancing task performance using generated data. We evaluated our approach by testing with the generated data, and we observed a noticeable improvement in the depth estimation task both before and after fine-tuning. Consequently, our approach yields results that are comparable to those achieved by networks specifically designed for daytime prediction. These findings highlight the effectiveness of utilizing synthetic data to enhance the performance of depth estimation tasks, particularly in nighttime settings.

1. INTRODUCTION

Depth estimation is a fundamental problem in computer vision that is critical for a wide range of applications, including navigation for autonomous vehicles, augmented reality, and scene understanding. Accurate depth estimation is also essential for tasks such as object detection, tracking, and segmentation, as well as 3D reconstruction. Stereo vision is one method that allows for an accurate estimation of absolute depth using multiple cameras. Another approach is using geometry-based methods such as Structure from Motion (SfM), which is widely used for 3D reconstruction and simultaneous localization and mapping (SLAM). SfM estimates 3D structures from a series of 2D image sequences by exploiting geometric constraints(Zhao et al., 2022). These methods tend to treat depth estimation as a purely geometrical problem, ignoring the content of the images. Monocular depth estimation seems ill-posed without a second input image to enable triangulation (Godard et al., 2018). Yet, the human brain can estimate depth or at least relative depth from a single image. Humans do this by exploiting several cues learned over time, such as perspective, the size of different objects relative to each other, lighting, shadows, and occlusions. By learning these cues, a deep learning model can be trained to estimate the depth from a single image. Many methods were developed to do so, and yielded great results on popular datasets such KITTI(Geiger et al., 2013) and Cityscapes(Cordts et al., 2016). Both datasets, as well as many others used for outdoor depth estimation, consist solely of daytime images. Depth estimation models that are trained using daytime images often exhibit poor performance when applied to night images (Vankadari et al., 2020). This can be caused by the significant differences between the visual characteristics of the two domains. Night images encounter two challenges that day images do not. Firstly, there are problems with low visibility and variable illuminance. Secondly, the varying illuminations, caused by flickering streetlights or moving cars, can violate the assumption of brightness consistency that is present in daytime images where all pixels are lit by the same light source, the sun (Wang et al., 2021). The collection of high-quality depth data is a complex and a costly process. That is why many approaches tried utilizing semi-supervised or self-supervised learning.

Our approach suggests a cost-efficient method of generating night-time images through the use of an image translation generative adversarial network (Zhu et al., 2017). Image translation is transforming one image from one domain to another. In this work, we apply this technique on a subset of day images from the KITTI dataset to generate their corresponding synthetic night images. We then use the generated synthetic night images to fine-tune a pre-trained depth estimation network (Godard et al., 2018), thereby improving its performance on night images.

The paper is structured into several sections. The first section is the Related Works, where previously attempted approaches for the problem are demonstrated. The second section focuses on Image Translation. Here, we delve into the architecture used for the task, elaborate on the training process of the network, and discuss the generation of synthetic night-images. This section aims to provide a detailed explanation of how the translation from one domain to another is achieved. Moving on, the third section revolves around the Depth Estimation Network. We delve into the details of this network, thoroughly explaining the process of fine-tuning it using the generated images. Lastly, we have the Conducted Experiments and Results section. In this section, we present the experiments carried out to validate the proposed approach. We also provide the corresponding results obtained from these experiments.

2. RELATED WORK

In this section, we present other relevant studies that have addressed the task of depth estimation, specifically focusing on their applicability to night images or similar conditions with limited available data.

[*] Corresponding author

https://doi.org/10.5194/isprs-archives-XLVIII-1-W2-2023-171-2023 |

2.1 Unsupervised and Self-supervised Techniques

These approaches can be employed to eliminate the necessity of collecting ground truth depth information, although the availability of images remains essential. Therefore, these methods prove valuable when only images are available. Many unsupervised and self-supervised techniques have been introduced, yielding positive outcomes for the task. For instance, in (Godard et al., 2018), a self-supervised mono depth estimation was carried out on the KITTI dataset. The architecture and concept of this network will be thoroughly explained in Section 4.

2.2 Approaches For Night Depth Estimation

Methods trained on daytime images exhibit poor performance when applied to nighttime images due to the presence of photometric inconsistencies. While lighting consistency is naturally assumed in daytime images, this assumption does not hold true for nighttime images. Lighting inconsistencies can arise from street lamps, car headlights, or variations in illuminance across different areas of the image. Unfortunately, only a limited number of approaches have specifically addressed the challenge of depth estimation in nighttime conditions.

In (Spencer et al., 2020), DeFeat-Net is introduced as a system capable of simultaneously learning depth from a single image and obtaining a dense feature representation of the environment, along with estimating ego-motion between consecutive frames. Notably, this is achieved through a fully self-supervised approach, eliminating the need for any ground truth data other than a monocular stream of images. Moreover, the learned features exhibit invariance across various weather and lighting conditions.

Another approach, proposed by (Vankadari et al., 2020), considers the problem as a domain adaptation challenge. The depth map is trained using daytime images, employing an encoder-decoder architecture. In addition to that, another encoder is trained using real-time nighttime images. To train the nighttime encoder, an adversarial domain feature adaptation technique is employed, where the night encoder acts as a generator aiming to generate feature maps from a nighttime image that resemble the feature maps obtained from daytime images. By doing so, the depth decoder becomes capable of decoding both the daytime and nighttime feature maps in a consistent manner.

3. IMAGE TRANSLATION

Our approach consists of two primary steps: generating night images using an image translation network and then utilizing the generated data to fine-tune the depth estimation networks. In this section, we provide an explanation of the fundamental concept behind the translation network, including the employed losses and the architecture of the network.

To begin with, the data generation process involved utilizing a network from (Zhu et al., 2017),which implemented a cycle generative adversarial network (GAN) architecture (Goodfellow et al., n.d.). First, we will provide an overview of the architecture of cycle GANs, followed by an explanation of the loss utilized during training. Finally, we will delve into a detailed description of the architecture of the specific network employed in our approach.

3.1 Cycle Generative Adversarial Networks

In a Generative Adversarial Network (GAN), two competing networks are designed. The generative model, denoted as G, aims to capture the data distribution of the training data and generate images that closely resemble the real data. On the other hand, the discriminative model strives to differentiate between real images from the training dataset and those generated by the generative model. The objective is for G to generate images that are indistinguishable from the target domain, while the discriminative model, denoted as D, tries to accurately classify real and fake images. This dynamic creates a learning process in which G minimizes the loss, while D maximizes the same loss, known as the adversarial loss.

In the context of Cycle GAN, the network aims to map between two domains. This involves two generative networks, G and F, as depicted in Figure 1. G maps from domain X to domain Y, while F performs the reverse mapping. Additionally, there are two discriminative networks, D_y, which discriminates domain Y images, and D_x , which discriminates domain X images. The goal here is not only to generate images in both domains but also to enable conditional mapping of scenes from one domain to the other. For instance, if we have a daytime image of a car parked in front of a building and we want to translate it into a nighttime scene, it is not sufficient for the generator to produce a realistic nighttime image. We also require the generator to generate the same scene with the car and the building at night. This is controlled by the cycle consistency loss, ensuring that the translated images preserve the essential elements of the original scene.

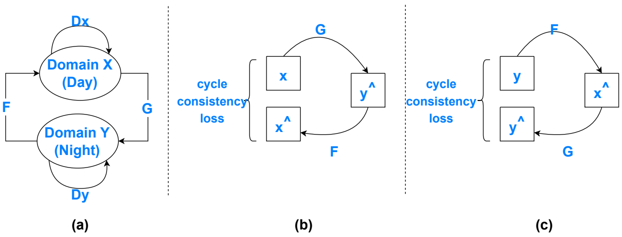

Figure 1. (a) cycle GAN, (b)(c) cycle consistency inspired by(Zhu et al., 2017)

3.2 Adversarial Loss

The adversarial loss is applied to both mapping functions G and F [11]. Let's consider G mapping from domain X to Y, with D_y responsible for distinguishing between generated samples by G and real samples from Y. The objective can be expressed as follows:

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x\sim p_{\text{data}}(x)}[1 - \log D_Y(G(x))] \quad (1)$$

where $mathcal L_{\text{GAN}}(G, D_Y, X, Y)$ = the adversial loss between G and
x, y = are samples from domains X and Y

Here, G aims to generate images that are similar enough to fool D_y into thinking they are real. Thus, G minimizes the objective, while D_y tries to maximize it by learning to differentiate between real and fake samples. This adversarial competition arises from both models striving to maximize and minimize the same objective.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-1-W2-2023-171-2023 | © Author(s) 2023. CC BY 4.0 License.

Similarly, a similar adversarial loss function $\mathcal{L}_{\text{GAN}}$(F, D_x, Y, X) is introduced for the mapping from domain Y to X, with the generator F and discriminator D_x.

3.3 Cycle Consistency Loss

In theory, the adversarial loss alone does not impose constraints on the generative networks to generate images similar to the source image. While they may generate images that closely resemble the target domain, they might not capture the essence of the input image. This misalignment with the original objective of the cycle GAN, which aims to translate an image while preserving the scene, imposes the need to introduce cycle consistency.

In Figure 1 (b), we observe the translation of image x from domain X to Y using G, followed by translating the result back to X using F, resulting in $\hat{x}$. Ideally, if both mapping functions G and F are perfect, x and $\hat{x}$ should be identical. Similarly, the cycle consistency is also defined in the opposite direction, as shown in Figure 1(c). To enforce this constraint, the cycle consistency loss is formulated as follows:

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[||F(G(x)) - x||]+ \\ \mathbb{E}_{y\sim p_{\text{data}}(y)}[||G(F(y)) - y||]\end{aligned} \tag{2}$$

where $\mathcal{L}_{\text{cyc}}(G, F)$ = the cycle loss
x, y = samples from domains X and Y
G, F = the generator functions

The cycle consistency loss ensures that the generated images from both mappings maintain consistency with the original input and output. Combining both the adversarial losses and the cycle consistency loss, the full objective function is obtained by summing equations (1) and (2).

3.4 Network Architecture

The architecture was adopted from (Johnson et al., 2016) that showed promising results. The generative network follows an encoder-decoder architecture consists of three convolutional layers, several residual blocks (He et al., 2015), two convolutional layers with a stride of ½, and a final convolutional layer that generates RGB images. During training, nine residual blocks were utilized to with image size of 256x256. For the discriminator, a PatchGans approach (Isola et al., 2016) was employed with a resolution of 70x70. The discriminator is trained to classify overlapping image patches. The patch architecture possesses fewer parameters compared to a full image discriminator, making it suitable for discriminating arbitrary image sizes.

3.5 Training

Initially, we utilized a pre-trained version of the network to generate the images. However, the results did not meet our expectations. Consequently, we proceeded to retrain the network from scratch. Our training was conducted on the Berkeley Deep Drive dataset(Yu et al., 2018), which contains images captured from the viewpoint of a car dashboard. For training purposes, we utilized a total of 12,454 daytime images and 22,884 nighttime images. The network underwent training for a total of 135 epochs. In Figure 2, we observe the original image alongside the translated nighttime images generated by both the pre-trained network and the network trained from scratch. The image generated by the pre-trained network exhibits scattered extra lights that should not be present, whereas these lights are absent in the version generated after the training process.

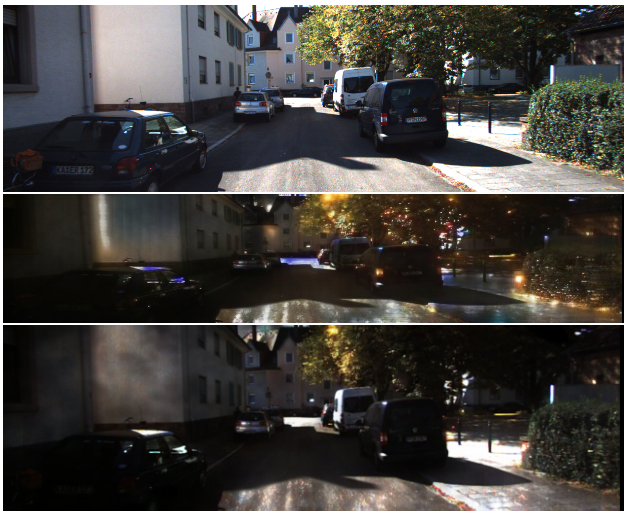

Figure 2. Arranged from top to bottom are the original image, the image translated by the pre-trained network, and the image translated by the network trained from scratch.

4. DEPTH ESTIMATION

The network utilized for depth estimation is inspired from the work of (Godard et al., 2018) and (Zhou et al., 2017). Their network was originally trained for depth estimation on the KITTI dataset (Geiger et al., 2013), which exclusively comprises daytime images. We performed fine-tuning on their network by incorporating the translated images generated from day to night. In this section, we will delve into the fundamental concepts employed by their network, explain the derivation of the loss function, and explore the network architecture.

4.1 Self-supervised learning

Self-supervised learning is a form of unsupervised learning wherein the data itself acts as the source of supervision. It involves defining an auxiliary task, known as the pretext task, which guides the loss function for the primary task. Typically, the outcome of the pretext task is not of primary concern. Instead, the focus lies on the intermediate representation. In this case, image reconstruction serves as the pretext task (Godard et al., 2018). The ultimate goal is not the final result of the reconstruction, but rather the intermediate variable utilized in the process, which is the depth in this particular scenario.

4.2 Self-supervised Loss

The framework proposed by (Godard et al., 2018) and (Zhou et al., 2017) involves training two networks simultaneously: a CNN for single view depth estimation and a camera pose estimation network. The supervision signal is derived from a pretext task known as view synthesis. In this task, the network aims to predict the view of a target frame, denoted as $\mathbf{I_t}$, based on the depth map of that frame, other images capturing the same scene from different poses (referred to as source frames), and the pose mapping between the target and source frames. The source frames $\mathbf{I_{t-1}}$ and $\mathbf{I_{t+1}}$ are selected as the previous and following frames in a frame sequence relative to $\mathbf{I_t}$. The pose network predicts the relative pose between consecutive frames.

To reconstruct the target view $\mathbf{I_t}$, pixels are sampled from a source view $\mathbf{I_s}$ using the predicted depth map $\hat{\mathbf{D_t}}$ and the relative pose $\hat{\mathbf{T_{t\to s}}}$. Let $\mathbf{p_t}$ denote the pixel coordinate in $\mathbf{I_t}$ and

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-1-W2-2023-171-2023 | © Author(s) 2023. CC BY 4.0 License.

$\mathbf{K}$ denote the camera intrinsics. The projection of $\mathbf{p_t}$ into $\mathbf{I_s}$, representing the pixel coordinates of the corresponding pixel in $\mathbf{I_s}$, can be determined as follows:

$$\mathbf{p_s} \approx \mathbf{K}\hat{\mathbf{T}_{\mathbf{t}\to\mathbf{s}}}\hat{\mathbf{D}}_{\mathbf{t}}(\mathbf{p_t})\mathbf{K}^{-1}\mathbf{p_t} \tag{3}$$

Applying the same process for each pixel in $\mathbf{I_t}$ while considering $\mathbf{I_{t-1}}$ and $\mathbf{I_{t+1}}$ as the source frames, this way, we project pixels of the target frame onto the source frames. The pixel value of every pixel in the target is predicted by interpolating the values of $\mathbf{p_s}$ and its neighboring pixels of both source frames. By following this procedure, an estimated target frame $\mathbf{I'_t}$ is obtained. The depth network is trained by minimizing the photometric reprojection error $\mathbf{L_p}$, where $\mathbf{p_e}$ represents the photometric reconstruction error:

$$\mathbf{L_p} = \sum_{t'} \mathbf{p_e}(\mathbf{I_t}, \mathbf{I'_t}) \tag{4}$$

Here $\mathbf{p_e}$ is a photometric reconstruction error, e.g. the L1 distance in pixel space between the original target frame and the predicted.

4.3 Network Architecture

The depth estimation network employs a U-Net architecture (Weng and Zhu, 2015), which consists of an encoder-decoder network with skip connections. The encoder network is based on ResNet18 (He et al., 2015), and the weights are initialized using pretrained weights from ImageNet (Russakovsky et al., 2014).

For the pose estimation network, the architecture is derived from (Wang et al., 2017). It also utilizes ResNet18 (He et al., 2015) as its foundation. The network takes two frames as input and produces a single 6-degrees of freedom (DOF) relative pose between the frames.

In the training process for monocular depth estimation, a sequence of three consecutive frames is utilized, and the pose is estimated between every two consecutive frames within that sequence. To augment the data, horizontal flipping is applied, and there is a 50% chance of altering the brightness, contrast, saturation, and hue jitter. The augmentation is performed on all three input images in a consistent manner.

The models are implemented using PyTorch (Paszke et al., n.d.) and trained using the Adam optimizer (Kingma and Ba, 2014) for 20 epochs. A patch size of 12 is used, and the training is conducted on the KITTI dataset (Geiger et al., 2013). Both the input and output images have a resolution of 640x192. During training, the learning rate starts at 10^{-4} for the first 15 epochs and then drops to 10^{-5} for the remaining five epochs.

The training process described above was conducted by the original authors exclusively using daytime images from the KITTI dataset. In the following sections we will describe our fine-tuning process.

5. EXPERIMENTS

Firstly, we will discuss the results of the image translation and highlight some of the challenges encountered. Subsequently, we will present the various scenarios employed to evaluate the performance of the depth estimation network. The test set utilized in all of our experiments consists of selected images from the KITTI dataset that have undergone translation from day to night.

5.1 Incompatible resolution challenge

The first challenge we encountered in our work arose from utilizing two different networks. The image translation network from (Zhu et al., 2017) produced images with a fixed resolution of 256x256, irrespective of the input resolution. However, this resolution was incompatible with the depth estimation network, which expected inputs of size 640x192. Additionally, the images in the KITTI dataset had dimensions of 1241x376.

Resizing the images resulted in significant degradation in quality. To address this issue, we employed a strategy of dividing the images into sub-images and feeding them to the network individually. Subsequently, the translated sub-images were combined to form the final translated image.

Initially, we experimented with dividing the image into four non-overlapping sub-images. As depicted in Figure 3, it was evident that the different divisions were easily distinguishable. Each pixel in the input image contributed to the overall color palette of the output image, resulting in a fragmented appearance. To mitigate this effect, we adopted a different approach and divided the image into overlapping sub-images with a horizontal shift of 20 pixels. In the final image, each pixel's value was calculated as the average of all values from the sub-images that contained that pixel. As shown in Figure 4, the region in the middle was present in all four sub-images, resulting in the values of that region in the final image being the average across the four sub-images. It's worth noting that we used more than four sub-images by applying a 20-pixel shift, which ultimately resulted in a final image dimension of 640x192. Figure 3 demonstrates the noticeable improvement achieved through this approach.

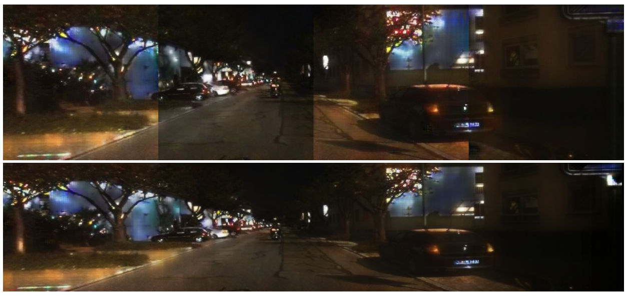

Figure 3. Comparison of the non-overlapping division (up) and overlapping division versions (down).

5.2 Translation Results

The pre-trained translation model's results were not consistently perfect, with certain common errors observed in some translated images. Figure 2 demonstrates an instance where the network erroneously predicted additional non-existent lights on the left side of the first image. The network's objective is to learn how to illuminate lights that are not naturally lit during the day but should appear at night, such as car headlights. However, there are instances where the network mistakenly identifies other image elements as lights when they are not. To

Figure 4. Demonstration of the 20 pixel shift

address this issue, we conducted training from scratch on the Berkeley Deep Drive dataset, resulting in a significant reduction of such errors.

5.3 Evaluation Metrics for Depth Estimation

We follow the evaluation metrics employed by (Godard et al., 2018), which consist of error metrics where lower values indicate better performance, as well as accuracy metrics where higher values indicate better performance.

5.3.1 Relative error using the absolute

$$\text{AbsRel} = \frac{1}{n}\sum_{1}^{n}\left|\frac{g_t - p_{\text{red}}}{g_t}\right| \tag{5}$$

Where g_t is the ground truth depth map generated from the Velodyne sensor points included in the KITTI dataset, and p_{red} is the predicted depth map.

5.3.2 Relative error using square

$$\text{SqRel} = \frac{1}{n}\sum_{1}^{n}\left(\frac{g_t - p_{\text{red}}}{g_t}\right)^2 \tag{6}$$

5.3.3 Root mean square error

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{1}^{n}(g_t - p_{\text{red}})^2} \tag{7}$$

5.3.4 Root mean square error of the log

$$\text{RMSE}_{\text{log}} = \sqrt{\frac{1}{n}\sum_{1}^{n}(\log(g_t) - \log(p_{\text{red}}))^2} \tag{8}$$

5.3.5 Accuracy metric using threshold

$$\delta = \max\left(\frac{g_t}{p_{\text{red}}}, \frac{p_{\text{red}}}{g_t}\right) \tag{9}$$

$$\delta < 1.25 = \frac{\text{pixels where } \delta < 1.25}{n} \tag{10}$$

$$\delta < 1.25^2 = \frac{\text{pixels where } \delta < 1.25^2}{n} \tag{11}$$

$$\delta < 1.25^3 = \frac{\text{pixels where } \delta < 1.25^3}{n} \tag{12}$$

where δ = threshold
n = number of samples

5.4 Fine Tuning Parameters

The parameters that were chosen for the fine tuning were: a learning rate of 10^4 , training with Adam (Kingma and Ba, 2014), batch size was 10 and smoothness term for regularization λ was 0.001. All the training in the following scenarios was conducted for 22 epochs.

5.5 Different Scenarios and Quantitative results

The test set is 697 images translated from KITTI(Geiger et al., 2013) from day to night. These images were used to evaluate the next scenarios.

- We initially evaluated the pretrained network from (Godard et al., 2018) on the original daytime images of the test set, without performing any fine-tuning or image translation on our part.
- We then evaluated the performance of the pretrained network on the translated night images of the test set without any fine-tuning.
- A total of 39810 images were generated for training purposes, along with an additional 4424 images for validation. For each image that underwent translation for training or validation, the preceding and subsequent frames were also translated. It's important to note that the generated data was not subjected to any filtering; it was all utilized for fine-tuning the network. Subsequently, the network's performance was evaluated on the same test set.
- The images underwent a filtering process using the GANs' discriminator network. This discriminator acts as a classifier, determining whether images are genuine nighttime images or not, and assigning a score ranging from 0 to 1, where a score of 1 indicates a real image. The training images were filtered based on this score, selecting only those with a score higher than 0.85. As a result, 3600 images were chosen for training, while the validation and test sets remained unchanged.
- The filtering process was repeated using the same methodology as before, but this time employing a cutoff score of 0.7. As a result, 17293 images were chosen for training.

As observed from Table 1, the first two rows serve as the baseline for our comparison. The test on daytime images represents the ideal scenario, showcasing the performance of the network trained specifically on daytime images. If our test results approach those of the daytime images, it indicates that our depth estimation works well at night, similar to how the pretrained version performs during the day.

The second row corresponds to the test on translated nighttime images using the pretrained network without any fine-tuning. This serves as our starting point for improvement. Subsequently, the remaining rows in the table demonstrate our tests

Scenario	The lower, the better				The higher, the better		
	AbsRel	SqRel	RMSE	RMSE_{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Daytime images (no fine-tuning)	0.115	0.905	4.863	0.193	0.877	0.959	0.981
Nigttime images (no fine-tuning)	0.177	1.481	6.550	0.271	0.737	0.903	0.957
Nigttime images (fine-tuned using all data)	0.132	1.057	5.489	0.219	0.836	0.944	0.975
Nigttime images (fine-tuned, filtered score of 0.85)	0.148	1.232	5.619	0.234	0.813	0.935	0.971
Nigttime images (fine-tuned, filtered score of 0.7)	0.133	1.064	5.385	0.216	0.842	0.946	0.976

Table 1. Evaluation of different training and test scenarios

after the fine-tuning process. We observe a significant enhancement compared to the initial nighttime test, although the performance has not yet reached the level of the daytime test.

Further analysis involves the filtering test, where we selectively choose translated images based on their authenticity score as determined by the discriminator. Initially, we select all images above a score of 0.7, amounting to approximately 17 thousand images. This filtering mildly improves some of the evaluation criteria compared to not filtering at all. However, when using a stricter threshold of 0.85, resulting in 3600 images, the performance is worse than not filtering at all. This observation indicates that the variety and size of the training set play a crucial role in the overall outcome.

Table 2 presents a comparison of various depth estimation methods conducted by (Godard et al., 2018) using daytime images from the KITTI dataset. The methods are denoted by D, S, and M, representing the use of depth ground truth for supervision, self-supervised stereo vision, and self-supervised monovision, respectively. In the last row, we showcase our best results evaluated on nighttime images. It is important to note that the test is not perfect since it was not conducted on the same data. However, our intention is to demonstrate the performance of our model in its specific task (night depth estimation) compared to different models designed for day depth estimation.

As observed from the table, the evaluation results of our model fall somewhere between the results of the other approaches. It is evident that our model does not perform as well on nighttime images as it does on daytime images. Nevertheless, it remains comparable to other methods specifically developed for daytime depth estimation. It is worth noting that the other models were tested on original daytime images, while our model was evaluated on generated nighttime images.

5.6 Qualitative Test

The model was tested on actual nighttime images obtained from the Berkeley Deep Drive dataset(Yu et al., 2018). Although we do not possess the ground truth depth information for this dataset, we utilized it solely for qualitative purposes, comparing the appearance of the depth maps generated by the model before and after fine-tuning. The Berkeley dataset comprises images captured in various environments, including nighttime scenes. The results can be observed in the Figure 5.

6. CONCLUSION

Our approach has demonstrated remarkable results in the task of depth estimation. Based on the conducted experiments, we conclude that image translation holds immense potential as an affordable image synthesis tool for generating data that can be utilized by various tasks. However, it requires further refinement and examination to understand the impact of data on training. Furthermore, image translation holds promise beyond day and night scenarios, such as simulating different seasons or transforming images from a simulated environment to resemble those captured in real-life scenes.

ACKNOWLEDGEMENTS

This research has been supported by funding of Prof. Naser El-Sheimy from NSERC CREATE and Canada Research Chairs programs. This research was also enabled in part by support provided by the Research Computing Services group at the University of Calgary.

REFERENCES

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December, 3213-3223. https://arxiv.org/abs/1604.01685v2.

Geiger, A., Lenz, P., Stiller, C., Urtasun, R., 2013. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, 32, 1231-1237. https://click.endnote.com/viewer?doi=10.1177

Godard, C., Aodha, O. M., Firman, M., Brostow, G., 2018. Digging Into Self-Supervised Monocular Depth Estimation. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October, 3827-3837. https://arxiv.org/abs/1806.01260v4.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., n.d. Generative Adversarial Nets. http://www.github.com/goodfeli/adversarial.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December, 770-778. https://arxiv.org/abs/1512.03385v1.

Isola, P., Zhu, J. Y., Zhou, T., Efros, A. A., 2016. Image-to-Image Translation with Conditional Adversarial Networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January, 5967-5976. https://arxiv.org/abs/1611.07004v3.

Johnson, J., Alahi, A., Fei-Fei, L., 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9906 LNCS, 694-711. https://arxiv.org/abs/1603.08155v1.

Kingma, D. P., Ba, J. L., 2014. Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://arxiv.org/abs/1412.6980v9.

Method	Train	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen	D	0.203	1.548	6.307	0.282	0.702	0.890	0.890
Liu	D	0.201	1.584	6.471	0.273	0.680	0.898	0.967
Klodt	D*M	0.166	1.490	5.998	-	0.778	0.919	0.966
AdaDepth	D*	0.167	1.257	5.578	0.237	0.771	0.922	0.971
Kuznietsov	DS	0.113	0.741	4.621	0.189	0.862	0.960	0.986
DVSO	D $*$ S	0.097	0.734	4.442	0.187	0.888	0.958	0.980
SVSM FT	DS	0.094	0.626	4.252	0.177	0.891	0.965	0.984
Guo	DS	0.096	0.641	4.095	0.168	0.892	0.967	0.986
DORN	D	0.072	0.307	2.727	0.120	0.932	0.984	0.994
Zhou †	M	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Yang	M	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Mahjourian	M	0.163	1.240	6.220	0.250	0.762	0.916	0.968
GeoNet †	M	0.149	1.060	5.567	0.226	0.796	0.935	0.975
DDVO	M	0.151	1.257	5.583	0.228	0.810	0.936	0.974
DF-Net	M	0.150	1.124	5.507	0.223	0.806	0.933	0.973
LEGO	M	0.162	1.352	6.276	0.252	-	-	-
Ranjan	M	0.148	1.149	5.464	0.226	0.815	0.935	0.973
EPC ++	M	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Struct2depth '(M)'	M	0.141	1.026	5.291	0.215	0.816	0.945	0.979
Monodepth 2 w/o pretraining	M	0.132	1.044	5.142	0.210	0.845	0.948	0.977
Monodepth2	M	0.115	0.903	4.863	0.193	0.877	0.959	0.981
Monodepth2 (1024 × 320)	M	0.115	0.882	4.701	0.190	0.879	0.961	0.982
Ours nighttime images	M	0.133	1.064	5.385	0.216	0.842	0.946	0.976

Table 2. Comparison of different depth estimation approaches as reported by (Godard et al., 2018) against our approch

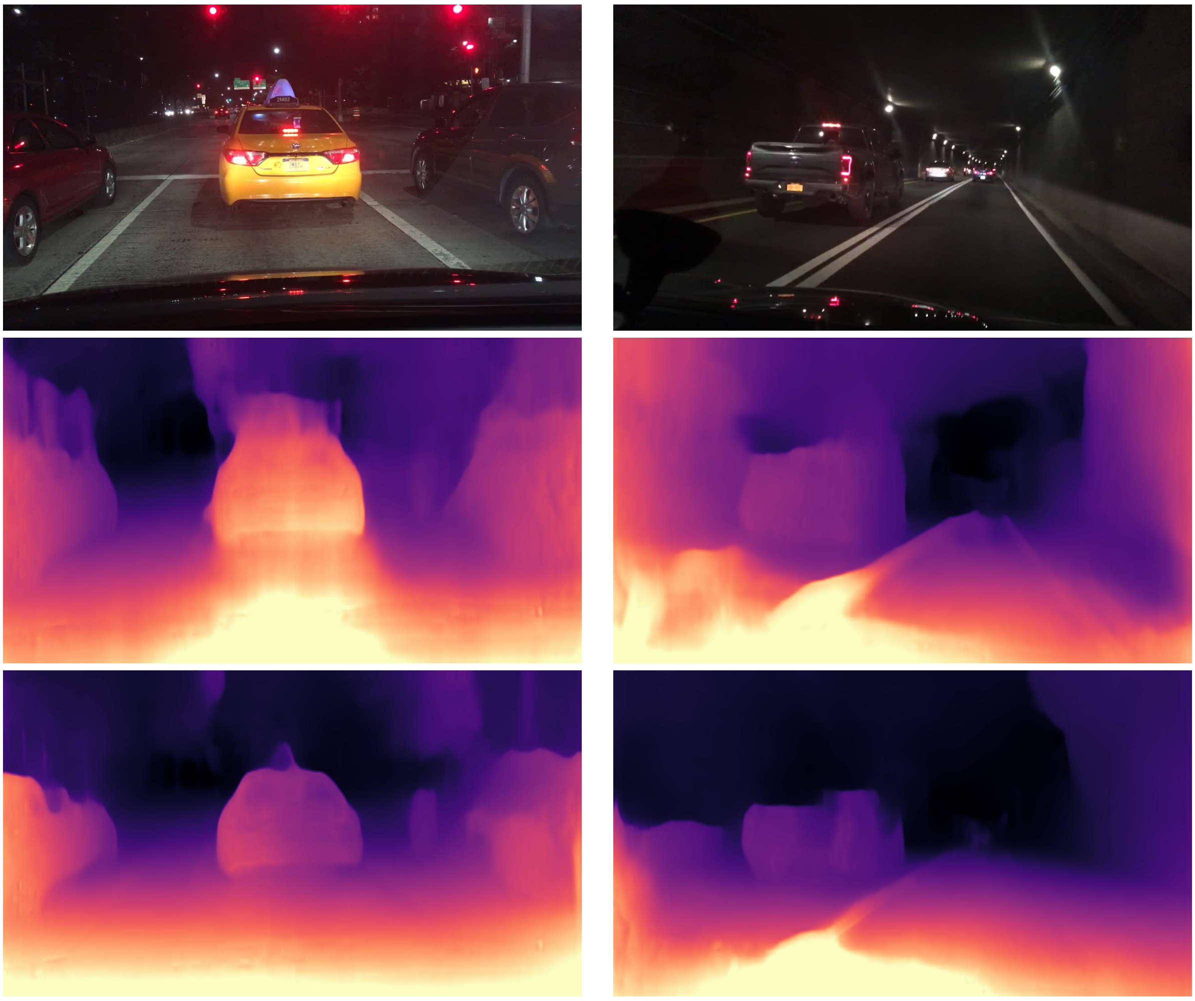

Figure 5. From top to bottom: the original images, the depth maps before fine-tuning, and the depth maps after fine-tuning.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., Facebook, Z. D., Research, A. I., Lin, Z., Desmaison, A., Antiga, L., Srl, O., Lerer, A., n.d. Automatic differentiation in PyTorch.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, L., 2014. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115, 211-252. https://arxiv.org/abs/1409.0575v3.

Spencer, J., Bowden, R., Hadfield, S., 2020. DeFeat-Net: General Monocular Depth via Simultaneous Unsupervised Representation Learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 14390-14401. https://arxiv.org/abs/2003.13446v1.

Vankadari, M., Garg, S., Majumder, A., Kumar, S., Behera, A., 2020. Unsupervised Monocular Depth Estimation for Night-time Images using Adversarial Domain Feature Adaptation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12373 LNCS, 443-459. https://arxiv.org/abs/2010.01402v1.

Wang, C., Buenaposada, J. M., Zhu, R., Lucey, S., 2017. Learning Depth from Monocular Videos using Direct Methods. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2022-2030. https://arxiv.org/abs/1712.00175v1.

Wang, K., Zhang, Z., Yan, Z., Li, X., Xu, B., Li, J., Yang, J., 2021. Regularizing nighttime weirdness: Efficient self-supervised monocular depth estimation in the dark. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16055–16064.

Weng, W., Zhu, X., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *IEEE Access*, 9, 16591-16603. https://arxiv.org/abs/1505.04597v1.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T., 2018. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2633-2642. https://arxiv.org/abs/1805.04687v2.

Zhao, C., Tang, Y., Sun, Q., 2022. Unsupervised Monocular Depth Estimation in Highly Complex Environments. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Zhou, T., Brown, M., Snavely, N., Lowe, D. G., 2017. Unsupervised Learning of Depth and Ego-Motion from Video. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January, 6612-6621. https://arxiv.org/abs/1704.07813v2.

Zhu, J. Y., Park, T., Isola, P., Efros, A. A., 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-October, 2242-2251. https://arxiv.org/abs/1703.10593v7.