A LIGHTWEIGHT SELF-SUPERVISED TRAINING FRAMEWORK FOR MONOCULAR DEPTH ESTIMATION

Tim Heydrich, Yimin Yang

Lakehead University
Department of Computer Science
Thunder Bay, ON, Canada

*Shan Du**

The University of British Columbia Okanagan
Department of Computer Science,
Mathematics, Physics and Statistics
Kelowna, BC, Canada

ABSTRACT

Depth estimation attracts great interest in various sectors such as robotics, human computer interfaces, intelligent visual surveillance, and wearable augmented reality gear. Monocular depth estimation is of particular interest due to its low complexity and cost. Research in recent years was shifted away from supervised learning towards unsupervised or self-supervised approaches. While there have been great achievements, most of the research has focused on large heavy networks which are highly resource intensive that makes them unsuitable for systems with limited resources. We are particularly concerned about the increased complexity during training that current self-supervised approaches bring. In this paper, we propose a lightweight self-supervised training framework which utilizes computationally cheap methods to compute ground truth approximations. In particular, we utilize a stereo pair of images during training which are used to compute photometric reprojection loss and a disparity ground truth approximation. Due to the ground truth approximation, our framework is able to remove the need of pose estimation and the corresponding heavy prediction networks that current self-supervised methods have. In the experiments, we have demonstrated that our framework is capable of increasing the generator's performance at a fraction of the size required by the current state-of-the-art self-supervised approach.

***Index Terms*—** computer vision, depth estimation, deep learning, self-supervised

1. INTRODUCTION

Depth estimation is a fundamental and ill-posed problem of computer vision. There is a great interest in many areas from scene reconstruction [1] to augmented reality (AR) [2]. It has long been a key point of research, however, most traditional methods require multiple view points. Supervised learning of large deep convolutional neural networks (CNNs) overcame this issue [3, 4, 5]. In addition, CNNs are used in combination with passive sensors, cameras, which are usually cheaper and lighter than their active counterparts like LIDAR. However, supervised learning has the problem of requiring large amount of labeled data for training. This data is available to some degree for certain specific scenarios such as interior scenes with the NYUv2 dataset [6]. While there is data available there are some different scenes that are not at all or only limitedly covered in the datasets available. In order to allow for more versatile training, of various settings, unsupervised approaches were developed in recent years, the two most prominent being Godard et al. [7] and Zhou et al. [8]. Both of these approaches require two or more images during training but not during inference. Based on these proposed works, many new and improved training architectures were developed both unsupervised [9, 10] and self-supervised [2].

While most recent research has revolved around big heavy networks both for unsupervised and self-supervised networks, there are lightweight approaches out there such as MiniNet [10] and PyDNet [11]. Both MiniNet [10] as well as MonoDepthV2 [2] utilize secondary networks during training to boost their performance. In both cases, the networks are utilized to provide pose estimation between the images. These secondary networks are large heavy networks which are not utilized during inference. However, they drastically increase the complexity during training. We propose a novel self-supervised lightweight training framework which reduces the training complexity while maintaining an increase in performance. The training architecture we are proposing is specifically targeted for lightweight generator architectures.

Our novel self-supervised framework, shown in Figure 1, is able to boost the generator's performance while still maintaining a low complexity during training. Similar to other approaches, our proposed framework utilizes two input images at training time to calculate a disparity map ground truth approximation and compare it to the disparity prediction made by the generator. Furthermore, it is able to boost a lightweight target network's performance. While our novel approach does

*Corresponding author: Shan Du (shan.du@ubc.ca). This work was supported by the University of British Columbia Okanagan [GR017752], Lakehead University [11-50-16112406], and Vector Institute.

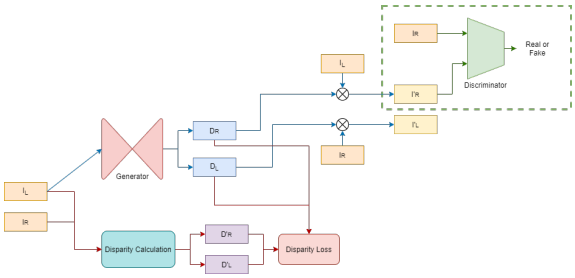

Fig. 1. Our proposed self-supervised framework overview. The blue arrows represent the work of [7]. The green arrows represent the work of [9] . The red arrows show our proposed addition for self-supervision.

not provide the same improvements as the current state-of-the-art approach MonoDepthV2, it is able to boost performance at only 3.21% of the size of MonoDepthV2. This decrease in size and complexity is particularly noticeable in a great reduction of training time. Our approach is able to complete each epoch at approximately $1/5$ of the time that MonoDepthV2 takes.

2. RELATED WORK

2.1. Supervised Approaches

Supervised approaches have greatly benefited from several large scale datasets such as NYUv2 [6], KITTI [12] and Cityscapes [13]. NYUv2 offers a large range of interior scenes with ground truth depth captured using a Kinect. KITTI and Cityscapes offer exterior scenes captured using a calibrated pair of stereo cameras. These datasets enabled a number of CNN based approaches [3, 4, 5, 14] that were able to excel at the task using supervised learning methods. Most of these approaches utilize an existing network architecture such as ResNet [15], DenseNet [16] or MobileNet [17] as feature encoder to which a specifically designed decoder is attached, often using skip-connections to improve accuracy, whose output is the depth prediction. These networks are able to achieve great results. However, as mentioned above, supervised learning comes with the disadvantage of requiring large amounts of labeled data for training. The datasets available cover a number of scenarios where depth estimation could be needed but there is still a number of various scenarios that are not present in the datasets.

Regarding lightweight design, the current state-of-the-art network is FastDepth [14] which utilizes Mobilenet as lightweight encoder and a custom decoder proposed by Wofk et al. FastDepth has a significant reduction in size and complexity with minimal impact on accuracy.

2.2. Unsupervised Approaches

Unsupervised approaches are able to overcome the need for labeled data. The most notable approach in recent years is Monodepth [7]. It utilizes a pair of stereo images during training but not at inference time. Monodepth utilizes a network structure similar to that of supervised models, however, instead of predicting a single depth image a number of disparity maps are predicted at different resolutions. These disparities are then used in combination with the input images to produce 'fake' images which can be compared with the original input images to evaluate network performance and calculate network loss. Monodepth is given a single input image and predicts the disparities for both the left and the right image. This approach is very effective in training a number of different network architectures. It is also the bases of several new and improved approaches published in recent years [9, 10, 18, 19, 20]. Poggi et al. [18] improved upon the work by using the binocular training images to simulate a trinocular setup which led to more accurate results. The most relevant to our framework is the work of Groenendijk et al. [9] where they proposed a number of different Generative Adversarial Networks (GANs) to be used in conjunction with the training structure proposed by [7]. Their approach utilizes the full resolution, same size as input, 'fake' image calculated from the disparity predictions not only to calculate reconstruction image loss but also for adversarial training. The discriminator is passed the original input image as well as the calculated image. A lightweight generator architecture PyDNet [11] utilizes the training framework proposed by Monodepth and is able to achieve good results. PyDNet uses a compact structure which does not have a clearly defined encoder and decoder, rather it combines feature extraction and upsampling.

The biggest challenge for unsupervised approaches is that while they are able to achieve adequate results without the need of labeled data they do not achieve the same performance as supervised models in most cases.

2.3. Self-Supervised Approaches

Self-Supervised approaches are able to offset the reduced performance of unsupervised methods. They are able to create a middle ground between unsupervised and supervised approaches. The most notable self-supervised method is MonodepthV2 [2]. There are similarities between Monodepth and MonodepthV2, where both try to minimize the photometric reprojection loss. However, unlike Monodepth, which uses a stereo pair of images for training, MonodepthV2 utilizes a temporal monocular video feed for training. In order to utilize said temporal features, MonodepthV2 uses a secondary network for pose estimation of the input images. MonodepthV2 has the option to be either trained on stereo inputs or on monocular temporal inputs. When training on stereo inputs, the pose network is not needed as the pose between the two inputs is given during training. Another approach which also utilizes temporal inputs is MiniNet [10]. Although self classified as unsupervised, it has several key similarities with MonodepthV2, where both utilize secondary

networks for pose estimation between the frames as well as try to minimize photometric reconstruction loss. MiniNet utilizes two shared weight networks for pose estimation, one for the frame before the input frame and one for the frame after the input frame. Both [2] and [10] utilize ResNet18 as their pose network which drastically increases the complexity and training time. Even with its shared weight approach, MiniNet still requires two passes through the pose network, one for each of the reference frames. While both MonoDepthV2 and MiniNet are capable of increasing the generator's performance, they do so at the cost of increased training time and resource consumption.

3. PROPOSED SELF-SUPERVISED FRAMEWORK

3.1. General Framework Explanation

Similar to existing state-of-the-art unsupervised and self-supervised methods, our framework requires a stereo pair of images at training time. However, we take advantage of the need for stereo images and propose that one can utilize said stereo pair to calculate a disparity estimation which can be utilized as ground truth approximation. A general overview of our proposed framework can be seen in Figure 1. The generator represents the target network of the framework, the goal is to boost its performance. Our proposed framework maintains the advances made by MonoDepth [7] utilizing their left-right consistency and photometric loss calculation. We further utilize the insight given by Groenendijk et al. [9] into adversarial learning to further help the target network, further explained in 3.2. Our final addition is our proposed use of ground truth approximation disparity maps. The inherent ambiguity of disparity map calculation, due to for example occluded pixels, leads to a 'ground truth' that is not 100% accurate but offers the generator additional information during training. Lightweight design was one of our key concerns when designing this framework. To this end, we decide to utilize a specifically small discriminator with low complexity. In addition, when comparing the computational cost of using additional neural networks for pose estimation, as is the case with MonodepthV2 [2] and MiniNet [10], disparity calculations are inexpensive and trivial. While we still maintain the need for an additional network, the discriminator, it is significantly smaller and less complex than the heavy networks used by MonoDepthV2 and MiniNet.

3.2. Adversarial learning

The utilization of adversarial training to boost network performance was proposed by Groenendijk et al. [9]. They explored a number of different discriminator architectures and generator settings. They found that normalization in the generator had the biggest impact on network performance in adversarial training. Furthermore, they found that a simple discriminator like the one we intend to use usually negatively impacts the performance of the target generator. However, as detailed in 4.3 we discovered that utilizing a discriminator along with our self-supervision improves overall performance. Inspired by [9], we utilize a small lightweight discriminator for our proposed structure. This discriminator consists of 3 fully connected layers. Furthermore, we utilize the WGAN architecture as basis for our proposed framework.

3.3. Disparity calculation

Disparity calculation is a long standing problem in computer vision. There are several approaches to solve it, but the most relevant for us is the Block Matching algorithm. We utilize it as it is simple and efficient in finding corresponding blocks in the image pairs. We use the Sum of Absolute Differences [21] to determine which blocks are similar and determine the disparity value. We chose a block size of 15 pixels as too large block size would result in overly smooth images and a small size would result in increased noise. We found that a block size of 15 resulted in adequate results that were able to boost network performance. Two disparity maps are calculated, one with the left input image as source and one with the right input image as source.

3.4. Disparity Loss

Several different loss functions were tested to determine the optimal one. We chose the L_1-norm as our loss function for the disparities as it is simple and proved to be most effective when compared to other more complex ones. Other loss functions tested were the L_2-norm, Structural Similarity (SSIM) as well as combinations of L_1, L_2 and SSIM. The overall loss for the predicted disparities with respect to the calculated disparities is calculated separately for the left and right disparity predictions. The resulting L_1 losses are then combined with equal weights and are added to the photometric reconstruction loss of the left-right consistency. The L_1 loss of our self-supervision is given equal weight as the photometric reconstruction loss.

4. EXPERIMENTS

4.1. Experimental Setup

Our framework and each network used with it have been trained and evaluated using the KITTI [12] dataset in particular the Eigen split [22]. The Eigen split is used to ensure a fair comparison between our framework and other methods. The split consists of 22.6K training images, 888 validation images and 697 test images. The implementation of our framework as well as tested generator networks is done in Pytorch [23]. Our training set up follows that of Groenendijk et al. [9] to ensure fair comparison. This means that all models are trained over 50 epochs in mini-batches of 8 using an Adam optimizer and plateau learning rate scheduler [24] with an initial learning

rate of 10^{-4}. None of the models used in our framework have any pre-trained weights. The training and evaluation of this model is performed on a Nvidia GTX 1080Ti. Models are evaluated quantitatively using the following metrics widely used by other methods [7, 11, 9], Absolute Relative Distance (Abs Rel), Squared Relative Distance (Sq Rel), Root Mean Squared Error (RMSE), Root Mean Squared Logarithmic Error (RMSE log), and accuracy within threshold t (δt , with t $\in [1.25, 1.25^2, 1.25^3]$).

4.2. Comparison to State-of-the-Art Self-Supervised Solution

Table 1. Comparison of our proposed self-supervised approach with MonoDepthV2. Highlighted results taken from [25] as they have performed a quantitative analysis of PyDNet and FastDepth trained on MonoDepthV2.

Generator	Supervision	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑	Training Size (Params. in M.)
PyDNet ([11])	L-R Unsupervised	0.163	1.399	6.253	0.262	0.759	0.911	0.961	1.970
PyDNet	MonoDepthV2	0.154	1.307	**5.556**	0.229	**0.812**	**0.932**	**0.970**	+11.690
PyDNet	Proposed	**0.145**	**1.154**	5.775	**0.238**	0.792	0.924	0.969	**+0.375**
FastDepth	L-R Unsupervised	0.370	5.868	9.859	1.084	0.614	0.781	0.868	4.02
FastDepth	MonoDepthV2	**0.156**	**1.260**	**5.628**	**0.231**	**0.801**	**0.930**	**0.971**	+11.690
FastDepth	Proposed	0.182	2.221	6.340	0.688	0.787	0.904	0.946	**+0.375**

We compare our proposed framework with the current state-of-the-art self-supervised approach MonodepthV2 [2]. We compare both frameworks using PyDNet and FastDepth as prediction network. The comparison is performed not only on quantitative network performance but also on the size increase during training. The goal is to determine the trade off between size and improved results for each of the two frameworks. The comparison can be viewed in Table 1. The network's performances are evaluated using the metrics that are mentioned in 4.1 and the size increase is measured by the increase in parameter count in millions at training time. From the comparison it becomes apparent that both our novel self-supervised approach as well as MonoDepthV2 approach improve overall network performance for both PyDNet as well as FastDepth. While MonoDepthV2 is able to achieve overall better improvements, there is a significant size difference between the two frameworks. Our approach is able to increase overall performance while requiring $\sim$3.21% of the size needed for MonoDepthV2. Furthermore, the reduction in size and complexity greatly reduces the training time. While MonoDepthV2 requires approximately 2500 seconds per epoch, $\sim$30 hours total, our approach only requires roughly 450-500 seconds, $\sim$6.5 hours total. This time is on an Nvidia GTX 1080 Ti using PyDNet as core prediction network.

4.3. Ablation Study: Framework Parts Evaluation

In order to determine optimal efficiency as well as impact on the target generator, we evaluate each part of the framework independently as well as together. The results of that comparison can be found in Table 2. The different components are evaluated for three separate target architectures, one heavy VGG [26] based and two lightweight ones, PyDNet and FastDepth. Training for each step is performed as lined out in Section 4.1. It needs to be noted that for PyDNet, two baselines are given, one is the official one taken from the paper [11] and the second one is from the Pytorch implementation of Monodepth and PyDNet. The test results clearly show improved performance for the lightweight networks when using the complete framework. The addition of our self-supervised disparity loss by itself positively impacts the δ accuracies but negatively impacts the other error metrics. Furthermore, while the addition of the discriminator by itself has an overall negative impact, when it is used in combination with our self-supervision it is able to further provide an overall increase in performance. When evaluating our framework on the heavy network, it becomes apparent that our self-supervised approach does not result in an increased network performance but rather the opposite. We believe this is because heavy networks tend to have better baseline performance compared to lightweight networks. Furthermore, their ability to make disparity predictions could supersede the accuracy of the ones acquired through the use of a simple conventional methods. These results show that our self-supervised framework is capable of increasing network performance for lightweight networks.

Table 2. Evaluation of different framework components, compared with baseline. Baseline is given by training networks on [7] training framework exclusively. PyDNet is given two baselines, one from the original paper and one from a Pytorch [23] implementation used for our framework.

Network Architecture	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
VGG Baseline	**0.151**	**1.325**	**5.876**	**0.246**	**0.791**	**0.920**	**0.965**
VGG GAN	0.152	1.357	6.003	0.249	0.788	0.917	0.963
VGG Self	**0.151**	1.335	5.951	0.248	0.789	**0.920**	0.964
VGG Self + GAN	0.159	1.467	6.152	0.256	0.784	0.915	0.961
PyDNet Baseline ([11])	0.163	1.399	6.253	0.262	0.759	0.911	0.961
PyDNet Baseline (Pytorch)	**0.145**	1.246	5.798	0.242	0.786	0.923	0.968
PyDNet GAN	0.160	1.462	6.156	0.256	0.770	0.910	0.962
PyDNet Self	0.149	1.316	5.810	0.245	**0.792**	0.923	0.967
PyDNet Self + GAN	**0.145**	**1.154**	**5.775**	**0.238**	**0.792**	**0.924**	**0.969**
FastDepth Baseline	0.370	5.868	9.859	1.084	0.614	0.781	0.868
FastDepth GAN	0.207	2.680	6.824	0.731	0.760	0.890	0.938
FastDepth Self	0.193	2.416	6.440	0.696	0.785	0.900	0.943
FastDepth Self + GAN	**0.182**	**2.221**	**6.340**	**0.688**	**0.787**	**0.904**	**0.946**

5. CONCLUSIONS

In this work, we proposed a novel lightweight self-supervised training framework for monocular depth estimation. Our framework specifically targets lightweight networks and aims to increase their performance at marginal cost to size and complexity. Our framework requires a stereo image pair to be given at training time to compute disparity ground truth approximations. Our method eliminates the need for pose estimation that other state-of-the-art frameworks have. For this reason we are able to significantly reduce the overall complexity of self-supervised learning. Our novel framework is able to compete with the current state-of-the-art method in terms of performance while significantly reducing the size. While we cannot exceed current performance in all metrics we believe that the trade off between size and performance is worth it.

6. REFERENCES

[1] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2noise: Learning image restoration without clean data," *Proceedings of the 35th International Conference on Machine Learning (ICML)*, vol. 80, pp. 2965–2974, 2018. [Online]. Available: https://arxiv.org/abs/1803.04189

[2] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3828–3838.

[3] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," *Proceeding of IEEE International Conference on Computer Vision (ICCV)*, 2015. [Online]. Available: http://arxiv.org/abs/1411.4734

[4] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," *Proceedings of Fourth International Conference on 3D Vision (3DV)*, 2016. [Online]. Available: http://arxiv.org/abs/1606.00373

[5] Z. Hao, Y. Li, S. You, and F. Lu, "Detail preserving depth estimation from a single image using attention guided networks," *Proceedings of International Conference on 3D Vision (3DV)*, 2018. [Online]. Available: http://arxiv.org/abs/1809.00646

[6] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European conference on computer vision*. Springer, 2012, pp. 746–760.

[7] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 270–279.

[8] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1851–1858.

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

[10] J. Liu, Q. Li, R. Cao, W. Tang, and G. Qiu, "Mininet: An extremely lightweight convolutional neural network for real-time unsupervised monocular depth estimation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 166, p. 255–267, Aug 2020.

[11] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "Towards real-time unsupervised monocular depth estimation on cpu," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 5848–5854.

[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] D. Wofk, F. Ma, T. Yang, S. Karaman, and V. Sze, "Fastdepth: Fast monocular depth estimation on embedded systems," *Proceedings of International Conference on Robotics and Automation (ICRA)*, pp. 6101–6108, 2019. [Online]. Available: https://arxiv.org/abs/1903.03273

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

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[18] M. Poggi, F. Tosi, and S. Mattoccia, "Learning monocular depth estimation with unsupervised trinocular assumptions," in *2018 International conference on 3d vision (3DV)*. IEEE, 2018, pp. 324–333.

[19] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 340–349.

[20] P.-Y. Chen, A. H. Liu, Y.-C. Liu, and Y.-C. F. Wang, "Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2624–2632.

[21] R. A. Hamzah, R. A. Rahim, and Z. M. Noh, "Sum of absolute differences algorithm in stereo correspondence problem for stereo matching in computer vision application," in *2010 3rd International Conference on Computer Science and Information Technology*, vol. 1, 2010, pp. 652–657.

[22] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *arXiv preprint arXiv:1406.2283*, 2014.

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

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

[25] F. Aleotti, G. Zaccaroni, L. Bartolomei, M. Poggi, F. Tosi, and S. Mattoccia, "Real-time single image depth perception in the wild with handheld devices," *Sensors*, vol. 21, p. 15, 12 2020.

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