Investigating the Impact of Encoder Architectures and Batch Size on Depth Estimation through Semantic Consistency

Iqra Nosheen[1], Talha Iqbal[2], Ihsan Ullah[1,2], Cathy Ennis[3], and Michael G. Madden[1,2]

[1]*School of Computer Science, University of Galway, Ireland.*
[2]*Insight SFI Research Centre for Data Analytics, University of Galway, Ireland.*
[3]*School of Computer Science, Technological University Dublin, Ireland.*

Abstract

Traditional methods for depth estimation rely on supervised learning with resource-intensive LiDAR data. Virtual synthetic datasets provide a cost-effective alternative, but bridging the domain gap between synthetic and real-world data remains a significant challenge. In existing work, this gap is addressed through domain adaptation techniques, aligning the feature distributions of synthetic (source) and real-world (target) domains. Our study explores the efficacy of different encoder architectures (ResNet variants with 35, 50, 101, 101-with-attention, and 152 convolution layers) and two batch sizes (2 and 4) for the depth estimation task. Our experiments show that ResNet101 without and with attention mechanisms provide the best performance across 2 and 4 batch sizes, respectively, compared to the other models. Conversely, the deeper architecture considered, ResNet152, shows the lowest performance, indicating that increasing the network depth does not necessarily lead to improved results for depth estimation tasks. This study's findings provide valuable insights for developing more effective depth estimation algorithms, and it suggests future directions in hyperparameter optimization and semantic consistency modeling.

Keywords: Depth Estimation, Semantic Consistency, Encoder Architectures, Batch sizes, Image translation

1 Introduction

Accurate depth estimation is important for a wide variety of computer vision applications such as autonomous driving, augmented reality, and robotics. Traditional depth estimation methods are based on supervised learning techniques, which require large annotated datasets with precise depth measurements. As noted by Shim and Kim (2024), these measurements are typically obtained through expensive and specialised equipment like Light Detection and Ranging (LiDAR) sensors. Since every dataset is designed with a specific application in mind, there is a requirement for databases for different applications, and most methods need to be re-trained for each application. The emergence of virtual synthetic data offers a cost-effective alternative. It allows for the generation of diverse and high-volume training datasets that can help improve the efficiency of models without the need for costly physical data collection tools. However, a significant challenge arises from the domain gap between synthetic and real data, where the models trained solely on synthetic data may struggle to generalise effectively to real-world scenarios. Domain adaptation techniques offer a solution, by enabling the transfer of models trained on fully annotated source datasets to non-annotated target datasets. This is achieved by aligning the feature distributions of the source and target domains, often using techniques such as adversarial training, domain-invariant feature learning, and self-training methods (Taghavi et al., 2024). The source domain in this context refers to the synthetic dataset utilized for the training of the model, and the target domain represents a real-world dataset on which the trained model is evaluated.

Recent studies, such as Taghavi et al. (2024) and Lopez-Rodriguez and Mikolajczyk (2023), have demonstrated that the incorporation of depth information during training can reduce the domain gap in semantic segmentation and instance detection, while the use of semantic information can also help to bridge the domain gap in depth estimation. Taghavi et al. (2024) applied semantic segmentation to enhance depth estimation, by employing a shared encoder-decoder architecture based on the Swin Transformer that simultaneously processes both tasks. Semantic segmentation provides contextual information about the scene that helps to improve the

https://doi.org/10.1049/icp.2024.3295

accuracy of the depth estimation task. The approach was evaluated using the Cityscapes dataset as the source and NYU Depth V2 dataset as the target. Zhou et al. (2024) have explored the simultaneous application of depth estimation and semantic segmentation to improve the accuracy of 360-degree image analysis. The study employed the U-Net architecture and evaluated the approach on Stanford2D3D dataset.

Thanh et al. (2024) employed ResNet-18 as the encoder in their depth estimation network, due to its robust feature extraction capabilities. They used a batch size of 8 during both the meta-training and online adaptation phases to effectively simulate domain shifts and iteratively update the model's parameters. Their study concluded that these settings significantly enhanced depth prediction accuracy and improved generalization across various domains. The study Basak et al. (2020) examined a depth estimation method by employing an encoder-decoder architecture. They used a pre-trained DenseNet-169 model as the encoder and an up-sampling network as the decoder. This approach yielded high-quality depth maps with fewer parameters compared to contemporary methods. The network was trained with a batch size of 4, and leveraged transfer learning to improve boundary detection and the accuracy of depth prediction.

In previous work as discussed above, researchers have used contrasting experimental setups and have achieved different performance results. Our research aims to delve deeper into this area by evaluating the impact of various encoder architectures and batch sizes on the depth estimation task. We assess how different levels of network complexity affect the accuracy of depth predictions and provide insights into how these algorithms utilise semantic information to enhance the performance of the model.

2 Methodology: Dataset and Experimental Setup

In our study, we employ the KITTI dataset and the Virtual KITTI (vKITTI) dataset Lopez-Rodriguez and Mikolajczyk (2023), serving as the target and source domains, respectively; see the supplementary materials for dataset samples. KITTI is a benchmark dataset of real-world traffic that includes 42,382 rectified stereo image pairs with a resolution of 375 × 1242 pixels. In our experiments, we use depth maps from the KITTI dataset for evaluation purposes. We use the vKITTI dataset as the synthetic source domain. This a virtual dataset created in the Unity game engine, and comprises of 21,260 image-depth pairs with detailed annotations, including depth maps and semantic segmentation labels. Originally, the vKITTI and KITTI images had a resolution of 375 × 1242 pixels, but in our experiments, they are resized to 192 × 640 pixels for training and evaluation, to improve computational efficiency.

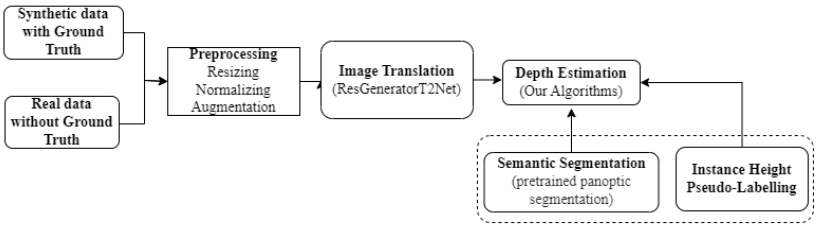

Figure 1: Block Diagram to illustrate the experimental flow

Our research investigates the impact of different encoder architectures on depth estimation accuracy, with a specific focus on how varying levels of network complexity influence this task. We use ResNet-34, ResNet-50, ResNet-110 and ResNet-152 (convolution-layered) models. Additionally, we add an attention mechanism to some models to evaluate whether this will enhance their feature extraction capabilities by focusing on the relevant region of interest. Figure 1 illustrates the complete workflow of our experimental setup.

To evaluate the results, we use Root Mean Square Error (RMSE), log Root Mean Square Error (RMSE log), Absolute Relative Difference (AbsRel), and Squared Relative Error (SqRel) and accuracies at three threshold values ($\delta < 1.25$, $\delta < 1.25^2$ and $\delta < 1.25^3$) as evaluation metrics, following the approach of Khan et al. (2021).

The experiments were conducted on a 48GB NVIDIA RTX 6000 GPU. Initially, models were trained separately for depth and semantic depth estimation using the batch size of 2 and 4. Subsequently, the final model was obtained through joint training of both models, using a batch size of 2. Each model was trained for a variable number of epochs, stopping when the loss function showed no significant improvement on manual inspection Ling et al. (2021).

Table 1: Experimental results of different depth estimation algorithms with training batch size 2

Cap 80m	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.253^3$ ↑
ResNet34	0.172	1.248	6.351	0.264	0.733	0.904	0.961
ResNet50	0.173	1.217	6.179	0.262	0.732	0.907	0.963
ResNet101 with Attention	0.166	1.162	6.032	0.253	0.752	0.912	0.964
ResNet101 without Attention	**0.161**	**1.144**	**6.010**	**0.250**	**0.764**	**0.915**	**0.965**
ResNet152	0.177	1.278	6.418	0.268	0.723	0.901	0.960
Cap 50m	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.253^3$ ↑
ResNet34	0.165	0.921	4.536	0.242	0.751	0.920	0.970
ResNet50	0.166	0.911	4.457	0.242	0.750	0.992	0.971
ResNet101 with Attention	0.159	0.873	4.373	0.234	0.769	0.926	0.972
ResNet101 without Attention	**0.153**	**0.845**	**4.321**	**0.230**	**0.781**	**0.929**	**0.973**
ResNet152	0.170	0.941	4.555	0.246	0.743	0.9180	0.969

Table 2: Experimental results of different depth estimation algorithms with training batch size 4

Cap 80m	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.253^3$ ↑
ResNet34	0.167	1.176	5.750	0.245	0.762	0.918	0.968
ResNet50	0.169	1.205	5.900	0.249	0.755	0.916	0.967
ResNet101 with Attention	0.171	1.196	5.672	0.249	0.756	0.917	0.966
ResNet101 without Attention	0.184	1.411	6.795	0.282	0.712	0.891	0.952
ResNet152	0.174	1.210	5.915	0.250	0.748	0.915	0.966
Cap 50m	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.253^3$ ↑
ResNet34	0.161	0.916	4.279	0.229	0.778	0.929	0.973
ResNet50	0.163	0.926	4.342	0.232	0.771	0.928	0.973
ResNet101 with Attention	0.165	0.953	4.263	0.235	0.771	0.926	0.971
ResNet101 without Attention	0.175	1.024	4.775	0.256	0.732	0.909	0.965
ResNet152	0.167	0.928	4.338	0.233	0.766	0.927	0.973

3 Results

This study evaluates the performance of various encoder architectures for depth estimation. The two primary experiments were conducted with training batch sizes of 2 and 4, and with maximum depths capped at 80 and 50 metres, respectively. Table 1 reports the results with a batch size of 2 while Table 2 illustrates the results with a batch size of 4.

The experiments reveal that for the batch size of 4, ResNet101 with attention performs better than other architectures. Surprisingly, however, for batch size of 2, ResNet101 without attention outperforms all other architectures in the depth estimation task, based on the RMSE values and other metrics that we consider.

ResNet152 shows the lowest performance, indicating that increasing the network depth does not necessarily translate to better results for this task. Overall, the performance of all the models improves when the depth capped value is reduced from 80m to 50m; not unexpectedly, a smaller depth range leads to more accurate depth estimation.

Moreover, the ResNet34, ResNet50 and ResNet152 (3 out of 5 implemented models) show a general improvement in performance with a batch size of 4 compared to batch size 2. This is because the gradient computed from a larger batch is more likely to approximate the true gradient of the entire dataset, leading to more stable updates Zhang et al. (2019). However, we note that even our larger batch size is only 4, so there

may be other factors at play.

4 Conclusion

This study provides a comprehensive analysis that can guide the design of more effective depth estimation algorithms in various applications. The results highlight the importance of the choice of encoder architecture and batch size. According to the RMSE values, in table 1 and 2, the encoder architecture based on ResNET 101 (without attention for batch size 2 while with attention for batch size 4) produces the best depth estimation results. Additionally, using a larger batch size (4 in our case) and reduced depth cap (50m in our case) provided the best model performance due to better gradient estimation and more stable as well as generalized learning.

These findings suggest that future depth estimation models should consider incorporating attention mechanisms and optimizing the batch size along with depth cap to achieve better performance. Future studies could further explore the interplay between batch size and other hyperparameters, such as learning rate and regularization methods, to improve the contextual understanding and robustness across diverse real-world scenarios, resulting in better depth estimation through semantic consistency.

Acknowledgments

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224 and SFI/12/RC/2289_P2 at the Insight SFI Research Centre Galway. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission

References

Basak, H., Ghosal, S., Sarkar, M., Das, M., and Chattopadhyay, S. (2020). Monocular depth estimation using encoder-decoder architecture and transfer learning from single rgb image. In *2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, pages 1–6. IEEE.

Khan, F., Hussain, S., Basak, S., Lemley, J., and Corcoran, P. (2021). An efficient encoder–decoder model for portrait depth estimation from single images trained on pixel-accurate synthetic data. *Neural Networks*, 142:479–491.

Ling, C., Zhang, X., and Chen, H. (2021). Unsupervised monocular depth estimation using attention and multi-warp reconstruction. *IEEE Transactions on Multimedia*, 24:2938–2949.

Lopez-Rodriguez, A. and Mikolajczyk, K. (2023). Desc: Domain adaptation for depth estimation via semantic consistency. *International Journal of Computer Vision*, 131(3):752–771.

Shim, D. and Kim, H. J. (2024). Divide: Learning a domain-invariant geometric space for depth estimation. *IEEE Robotics and Automation Letters*.

Taghavi, P., Langari, R., and Pandey, G. (2024). Swinmtl: A shared architecture for simultaneous depth estimation and semantic segmentation from monocular camera images. *arXiv preprint arXiv:2403.10662*.

Thanh, P. T. H., Bui, M. Q. V., Nguyen, D. D., Pham, T. V., Duy, T. V. T., and Naotake, N. (2024). Transfer multi-source knowledge via scale-aware online domain adaptation in depth estimation for autonomous driving. *Image and Vision Computing*, 141:104871.

Zhang, G., Li, L., Nado, Z., Martens, J., Sachdeva, S., Dahl, G., Shallue, C., and Grosse, R. B. (2019). Which algorithmic choices matter at which batch sizes? insights from a noisy quadratic model. *Advances in neural information processing systems*, 32.

Zhou, J., Wu, Y., Lim, H., and Kim, H. (2024). Omnidirectional depth estimation for semantic segmentation. In *2024 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–4. IEEE.