DepthStar: Quick Monocular Depth Estimation for Robotics Applications

Keivalya Bhartendu Pandya pandya.kei
Gaurav Kothamachu Harish kothamachuharish.g
Rachel Lim lim.rac

Abstract

Depth estimation is a critical task in robotics for navigation, obstacle avoidance, and self-driving applications. Contemporary state-of-the-art methods typically fall into two distinct categories: designing complex networks capable of direct depth map regression, or partitioning the input into bins or windows to reduce computational complexity. This project adopts the former approach, as we wish to optimize primarily for inference speed, and secondarily depth-map accuracy. Moreover, the Monocular Depth Estimation (MDE) are limited due to high operational costs, and annotations could be affected by sensor noise, hence often low resolution. We propose DepthStar, an autoencoder model that attempts to understand the 3D scene from a single RGB image trained using synthetic dataset. The primary objective is to implement and conduct comparative analyses of CNN-based and Transformer-based architectures for depth estimation, optimizing for both accuracy and computational efficiency, with particular emphasis on edge device deployment. We evaluate DepthStar on NYU Depth v2 using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Finally, we aim to keep the model as tiny as possible while gaining the best accuracy suitable for robotics applications.

Inference: `https://huggingface.co/spaces/keivalya/depthstar`

GitHub Repository: `https://github.com/khgaurav/depthstar`

1 Introduction

Monocular depth estimation uses a single camera to generate depth information, without the use of additional sensors such as structured light sensors or LiDAR, or needing two images as in a stero camera setup. Depth information is critical for accurately perceiving the environment, and is used in applications ranging from image editing to scene reconstruction. It's crucial for many tasks in robotics such as navigation, obstacle avoidance, and object interaction [14, 28, 16].

Most state-of-the-art methods focus on high-resolution, metric depth outputs at the cost of large models and often higher inference times [5, 14]. Additionally, even with the increase in number of image-depth datasets in recent years, there are still a limited number of datasets and many focus on a small category of possible scenes.

Our model, DepthStar, uses a hybrid convolutional-attention model optimized for a small model size and fast interference time to create relative depth maps for low resolution images. We utilize synthetic datasets that leverage the strengths of large, state-of-the-art models to create depth maps to train on. While our model doesn't achieve the accuracy of other models when evaluated on NYUv2, we are able to show significantly faster inference times with a smaller model that performed well on lower

Submitted to Prof. Huaizu Jiang as the CS7150: Deep Learning Project.

resolution datasets. We are using open-source PyTorch library to implement standard transformers. Everything else in the code is designed by us.

2 Related Work

Early MDE methods leveraged known information about the scene. In 1970, Horn proposed the first shading method, which used a propagation method to find the shape of a smooth object given a known surface photometry and light source position [15]. In 1975, Bajscy and Lieberman classified textures to use as depth cues to determine distance in outdoor scenes [3]. Other methods used prior knowledge of objects or learned it from examples [13, 9] or user-specified information [1, 29].

In 2005, Saxena et al. [24] proposed a supervised learning approach that trained a Markov Random Field on outdoor images and their corresponding depth maps. They later extended this approach to reconstructing indoor scenes [8] and generating mesh models to recreate the scene at different angles [25]. Liu et al. [20] used semantic segmentation to generate labels for the scene which assisted in 3D reconstruction.

Deep learning approaches appeared in the 2010s, following the release of the depth datasets NYUv2 consisting of 1449 pairs of RGB images and Microsoft Kinect depth images of indoor scenes [21], and KITTI which includes 6 hours of traffic scenes collected on RGB cameras, a grayscale stereo camera, and a 3D LiDAR [12]. The first was from Eigen et al. [10] that used a coarse-to-fine approach with two CNN networks. This was followed by other supervised CNN approaches followed [18, 26], unsupervised CNN [11, 30], supervised RNN [7, 27], unsupervised RNN [27], supervised GAN [17], and unsupervised GAN [2, 6].

Intel Labs introduced the transformer-based model with Dense Prediction Transformers (DPT) [22]. They transform an image into tokens using patches or ResNet50, passed through multiple transformer layers, resassembled to get image-like representations at each stage, and then fused into a final depth image. In MIiDas 3.0 they extend this work to include different transformer backbones [23] and ZoeDepth adds an additional module to compute metric depth (absolute distance) in addition to relative depth [4]. Transformer models were also used for PatchFusion, which uses a tile-based framework to improve upon a base model [19], and Metric3D which proposed a new model to generalize across unknown camera models [28].

Current state of the art transformer-based models include include vision transformer (ViT) based models like Metric3D V2 [16]and DepthPro [5]. Distill Any Depth introduced two new strategies, cross-context distillation to learn local and global depth simultaneously and a multi-teacher student model where a teacher is randomly selected to produce pseudo labels for unlabeled data [14].

3 Methods

We propose *DepthStar*, a hybrid convolutional-attention architecture designed for efficient and accurate monocular depth estimation even on low-resolution 2D images. As shown in Fig. 1, we use a convolution encoder to extract spatial features, leverage global context using a transformer encoder stack, and use a convolution decoder to reconstruct the depth map.

We utilize synthetic datasets to create training datasets of representative, low resolution depth data. We generate synthetic dataset utilizing state-of-the-art models such as Distill Any Depth [14], as from the literature we realize that their method is current state-of-the-art with pixel-level average accuracy exceeding 96% on most benchmarks.

3.1 Residual Block

Each Residual Block follows a two-layer bottleneck structure with identity mapping, as introduced in ResNet. This helps in learning residual function, which has been empirically shown to improve final performance during training and inference.

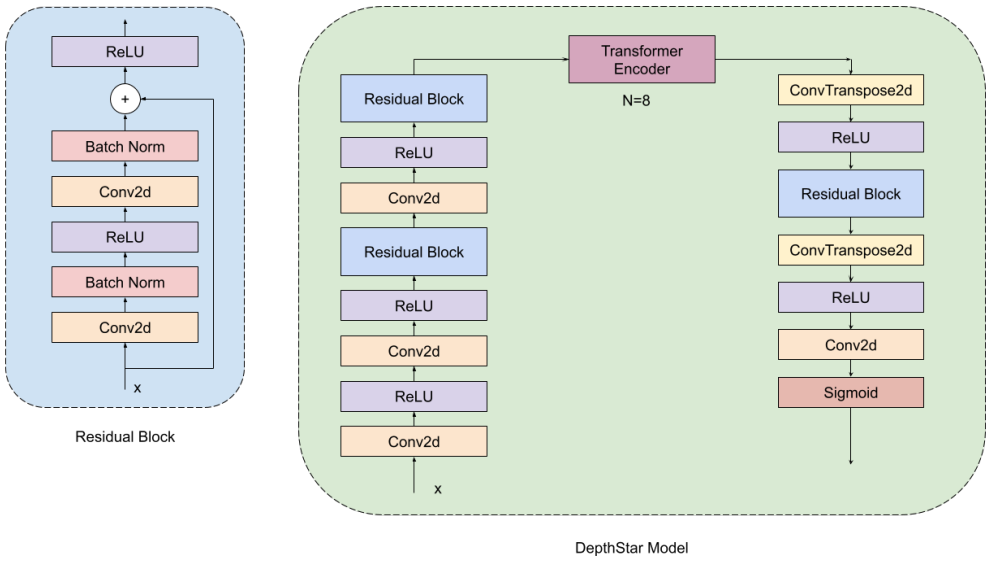

Figure 1: Overview of the ***DepthStar*** architecture. The model consists of a convolutional encoder for local feature extraction, a transformer-based bottleneck for global context aggregation, and a convolutional decoder for reconstructing the final depth map.

$$y = \mathbf{ReLU}(x + \mathcal{F}(x))$$

where,

$$\mathcal{F}(x) = \mathbf{BN}_2(\mathbf{Conv}_2(\mathbf{ReLU}(\mathbf{BN}_1(\mathbf{Conv}_1(x)))))$$

After further investigation, we realized that this block encourages feature reuse, that is, the network gives weighted importance to both low-level texture details and high-level semantic cues in the input image. The spatial and channel dimensions do not change with this operation.

3.2 Encoder

The encoder is designed to hierarchically extract spatial features from the input RGB image while progressively reducing its resolution. Initially, this takes place using convolutions and non-linearity, followed by residual connections to eliminate gradient vanishing, which we found to be the most effective method through experimentation. We initiated this block to make the model more aware about the overall context of the image, for example, if the image is based indoors or outdoors, and in which weights are learned to activate which neural pathway. In the upcoming section, we provide an analytical explanation on how and why this works better than other architecture.

$$[B, 3, H, W] \to [B, 64, H, W] \to [B, 128, H/2, W/2] \to [B, 256, H/4, W/4]$$

This effectively captures multiscale local patterns while preserving the ability to learn long-range dependencies downstream.

3.3 Bottleneck

CNNs often struggle to capture the global context of an image. We aim to capture this information by integrating Transformer encoder modules as a bottleneck of the autoencoder. This component models long-range interactions across spatial dimensions, enabling more informed depth predictions through

attention-based feature mixing. The output of the residual block is first flattened across its spatial dimensions and reshaped into a sequence of tokens.

$$[B, 256, H/4, W/4] \rightarrow [B, (H/4 \cdot W/4), 256]$$

Each layer in the Transformer encoder includes multi-head self-attention with 8 attention heads, followed by a feedforward network with dimensionality 512. Residual connections and layer normalization are applied around both sublayers, as standard in Transformer architectures.

$$\mathbf{Z} = TransformerEncoder(\mathbf{X}), \quad \mathbf{X} \in \mathbb{R}^{B \times N \times 256}, \quad N = \frac{H}{4} \cdot \frac{W}{4}$$

After attention-based processing, the output tokens $\mathbf{Z}$ are reshaped back into their original spatial structure.

$$[B, (H/4 \cdot W/4), 256] \rightarrow [B, 256, H/4, W/4]$$

This transformation allows the model to combine localized convolutional features with a global context, crucial for resolving ambiguities in monocular depth estimation.

3.4 Decoder

The decoder is designed to reconstruct high-resolution depth maps from the globally contextualized feature representations produced by the bottleneck. This decoding begins with reshaping $\mathbf{Z}$ from the reshaped output of the bottleneck. This tensor passes through a sequence of transposed convolution layers which progressively upsample the feature map while reducing its channel dimensionality.

$$[B, 256, H/4, W/4] \rightarrow [B, 128, H/2, W/2] \rightarrow [B, 64, H, W] \rightarrow [B, 1, H, W]$$

This ensures that high-level semantic features encoded in the bottleneck are effectively translated into fine-grained spatial predictions.

3.5 Hyperparameters

The choice of hyperparameters was done by combining empirical results along with computational constraints we had for training this network. We tried the limited grid-search approach by trying several embedding dimensions 256, 512, 768; number of transformer layers to be 4, 6, 8; number of attention heads 4 and 8. We found that aforementioned configuration achieved best validation accuracy while being inside the memory limit. All experiments were conducted using a single NVIDIA T4 GPU on Google Colab. Through ablation (see Table 2), we verified the contribution of residual blocks and the Transformer bottleneck.

4 Experiments

In this section, we discuss the constraints we impose on our research to simulate real-world robotics challenges, discuss how synthetic data was generated and validated, and present an ablation study explains the significance of each block in our model.

4.1 Experiment Settings

We imposed several challenges on ourselves to simulate real-world robotic challenges, such as low-resolution observation, robust environment that encounters indoor and outdoor objects with variable light intensity, and quick inference. We still make sure that our implementation is accurate to an acceptable threshold, and the model must be light-weight such that it must efficiently run on edge devices.

Additionally, given our limited compute budget as compared to state-of-the-art algorithms, we made sure that we utilize our resources as efficiently as possible.

Table 1: Summary of architectural and training hyperparameters used for the DepthStar model.

Hyperparameter	**Value**
Input Image Size	$3 \times 32 \times 32$
Initial Convolutions	Conv2d(3→64), Conv2d(64→128, stride=2)
Residual Blocks (✓)	Yes
Residual Block Channels	128, 512
Bottleneck Embedding Dim	512
Transformer Encoder Layers	8
Transformer Attention Heads	8
Feedforward Dimension	2048
Decoder Upsampling	ConvTranspose2d(512→128), ConvTranspose2d(128→64)
Final Output Layers	Conv2d(64→1), Sigmoid
Activation Function	ReLU
Loss Function	MAE / RMSE
Optimizer	Adam
Learning Rate	1×10^{-3}
Batch Size	16
Training Epochs	250

Table 2: Ablation study showing the individual and combined **effects of Residual Blocks and Transformer Encoder** on DepthStar's performance.

Residual Block	Transformer-Encoder	RMSE ↓	MAE ↓
✗	✗	0.0878	0.0542
✗	✓	0.0226	0.0125
✓	✗	0.0180	0.0108
✓	✓	**0.0110**	**0.0068**

4.2 Dataset

To generate training data for our depth estimation model without relying on expensive or hard-to-acquire real-world ground truth depth maps, we leveraged the pre-trained Distill-Any-Depth model to synthesize depth labels on the CIFAR-10 dataset. Although CIFAR-10 was originally curated for classification tasks, its diversity of low-resolution natural scenes – ranging from indoor to outdoor – makes it a useful proxy for real-world robotics perception under constrained sensory input.

Initially, we trained and tested our model on benchmark datasets such as NYUv2, however, we found that soon became computationally expensive and deviated from the challenges we wanted to explore in this study. There were several more challenges to this approach, which are discussed in Section A.1.

4.3 Ablation Study

Just for simplicity and consistency in experiments, we sample 10k images from CIFAR-10 as a training set for the network. As shown in Table 2, we compare how residual block and transformer-encoder impact the RMSE and MAE of the model. We made several observations while studying the effects of each component on the end results.

5 Conclusion

In this work, we present DepthStar, a compact and efficient monocular depth estimation model tailored for low-resolution images and edge-compute environments. Motivated by real-world robotics challenges, our model design emphasizes lightweight computation, fast inference, and strong gen-

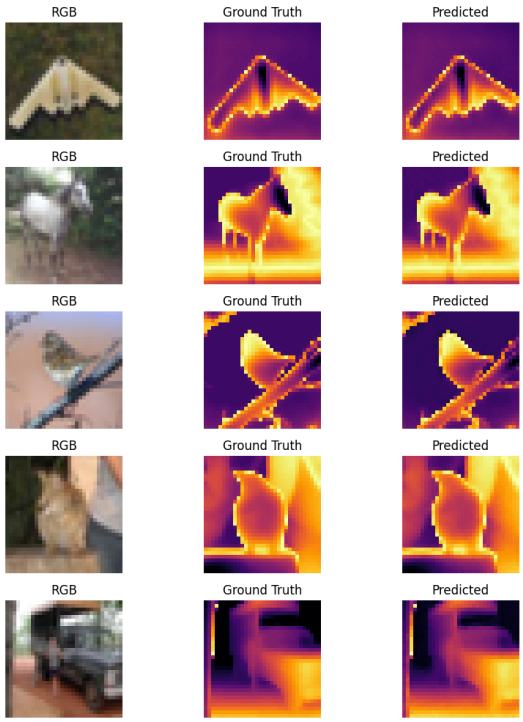

Figure 2: **Residual Block ✓, Transformer Encoder ✓.** The result exhibits the best visual quality, with sharp depth boundaries, preserved spatial detail, and coherent global structure—confirming the complementary.

Table 3: Quantitative results from the ablation study on CIFAR-10 (10k samples). We evaluate the impact of residual blocks and transformer encoder on depth estimation performance using RMSE and MAE metrics. The combination of both components yields the best results, demonstrating their complementary effectiveness in capturing local details and global structure.

Model	RMSE ↓	MAE ↓	Inference time (ms) ↓	Model Size (No. of params)
DepthStar	0.0110	0.0068	**4.4**	**32.378 M**
Distill Any Depth †	0.0000	0.0000	21.3	335.316 M
DepthPro *	–	–	3664.4	951.991 M
DPT 3.1	0.2178	0.1630	27193.0	343.987 M

† As the synthetic data (or ground truth) is generated using this model, we get no error between the test data and inference output.

* DepthPro is a metric MDE model instead of a relative MDE, hence the current dataset cannot be used to benchmark this model.

eralization, while being trained exclusively on synthetically generated pseudo-ground-truth depth maps.

Our experiments demonstrate that DepthStar outperforms or rivals state-of-the-art depth estimation models across multiple axes of performance. Specifically, it achieves the lowest RMSE and MAE among all evaluated models, while being orders of magnitude faster in inference and significantly smaller in model size compared to heavyweight models like DPT 3.1 or DepthPro. These results highlight the strength of our model in balancing accuracy, efficiency, and deployability.

As DepthStar is trained on synthetic data generated by a model which is prone to making mistakes, we often find it difficult to outperform state-of-the-art results. Although in several cases we prioritize speed while taking a slight dip in accuracy, we should also consider consequences of errors in

deployment settings. We are actively addressing this issue by coming up with smarter model architectures and hyperparameter tuning.

Key Takeaways:

- A hybrid lightweight architecture can deliver decent accuracy on low-resolution data while preserving real-time inference capability.
- Synthetic depth generated by strong teacher models (e.g., Distill Any Depth) offers a viable and scalable path for training student models under limited data regimes.
- Compared to DPT 3.1 and DepthPro, DepthStar achieves **6000x faster inference** and **10x smaller model size**, without a substantial drop in performance.

References

[1] I. R. A. CRIMINISI and A. ZISSERMAN. Single view metrology. In *International Journal of Computer Vision 40(2)*, page 123–148, 2000.

[2] F. Aleotti, F. Tosi, M. Poggi, and S. Mattoccia. Generative adversarial networks for unsupervised monocular depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[3] R. Bajcsy and L. Lieberman. Texture gradient as a depth cue. In *Computer Graphics and Image Processing*, pages 52–67, 1976.

[4] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth, 2023. URL `https://arxiv.org/abs/2302.12288`.

[5] A. Bochkovskii, A. Delaunoy, H. Germain, M. Santos, Y. Zhou, S. R. Richter, and V. Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv*, 2024. URL `https://arxiv.org/abs/2410.02073`.

[6] A. CS Kumar, S. M. Bhandarkar, and M. Prasad. Monocular depth prediction using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[7] A. CS Kumar, S. M. Bhandarkar, and M. Prasad. Depthnet: A recurrent neural network architecture for monocular depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 283–291, 2018.

[8] E. Delage, H. Lee, and A. Y. Ng. Automatic single-image 3d reconstructions of indoor manhattan world scenes. In S. Thrun, R. Brooks, and H. Durrant-Whyte, editors, *Robotics Research*, pages 305–321, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-48113-3.

[9] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2366–2374, Cambridge, MA, USA, 2014. MIT Press.

[10] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *CoRR*, abs/1406.2283, 2014. URL `http://arxiv.org/abs/1406.2283`.

[11] R. Garg, V. K. B. G, and I. D. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. *CoRR*, abs/1603.04992, 2016. URL `http://arxiv.org/abs/1603.04992`.

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

[13] F. Han and S.-C. Zhu. Bayesian reconstruction of 3d shapes and scenes from a single image. In *First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis, 2003. HLK 2003.*, pages 12–20, 2003. doi: 10.1109/HLK.2003.1240854.

[14] X. He, D. Guo, H. Li, R. Li, Y. Cui, and C. Zhang. Distill any depth: Distillation creates a stronger monocular depth estimator. *arXiv preprint arXiv: 2502.19204*, 2025.

[15] B. K. Horn. *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View*. PhD thesis, MIT, 1970.

[16] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[17] H. Jung, Y. Kim, D. Min, C. Oh, and K. Sohn. Depth prediction from a single image with conditional adversarial networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1717–1721, 2017. doi: 10.1109/ICIP.2017.8296575.

[18] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 239–248, 2016. doi: 10.1109/3DV.2016.32.

[19] Z. Li, S. F. Bhat, and P. Wonka. Patchfusion: An end-to-end tile-based framework for high-resolution monocular metric depth estimation. 2024.

[20] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1253–1260, 2010. doi: 10.1109/CVPR.2010.5539823.

[21] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[22] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12179–12188, October 2021.

[23] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022.

[24] A. Saxena, S. Chung, and A. Ng. Learning depth from single monocular images. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005. URL `https://proceedings.neurips.cc/paper_files/paper/2005/file/17d8da815fa21c57af9829fb0a869602-Paper.pdf`.

[25] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009. doi: 10.1109/TPAMI.2008.132.

[26] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[27] R. Wang, S. M. Pizer, and J. Frahm. Recurrent neural network for (un-)supervised learning of monocular videovisual odometry and depth. *CoRR*, abs/1904.07087, 2019. URL `http://arxiv.org/abs/1904.07087`.

[28] W. Yin, C. Zhang, H. Chen, Z. Cai, G. Yu, K. Wang, X. Chen, and C. Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[29] L. Zhang, G. Dugas-Phocion, J.-S. Samson, and S. Seitzt. Single view modeling of free-form scenes. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001. doi: 10.1109/CVPR.2001.990638.

[30] 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 (CVPR)*, July 2017.

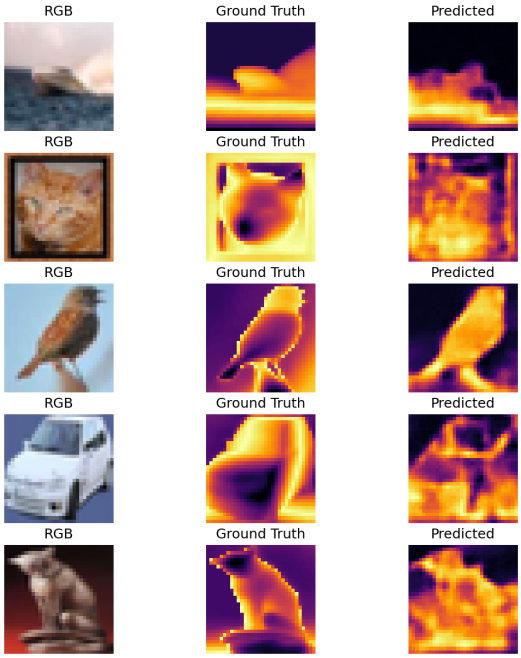

Figure 3: **Qualitative results from training DepthStar on NYUv2**. The low-resolution input (left), predicted depth (middle), and ground truth (right) illustrate the degradation in spatial detail and the prevalence of flat or noisy depth maps due to uninformative patches.

Table 4: **Performance comparison of DepthStar when trained on synthetic CIFAR-10 depth data versus NYUv2.** The model trained on NYUv2 exhibits higher error, and ineffectiveness of low-resolution random crops from high-resolution indoor datasets. This supports our decision to shift to synthetic training data that better aligns with our low-resolution input constraint.

Model	RMSE ↓	MAE ↓
DepthStar	0.0110	0.0068
DepthStar on NYUv2	0.3443	0.0792

A Appendix

A.1 Training on NYUv2 Dataset

Initially, we began by training on large dataset by random-cropping 32 x 32 low-resolution images for DepthStar to generate matching input-target pairs. However, after spending a significant amount of time, we realized that the images dataset contains high-resolution indoor scenes, where crucial depth cues often rely on fine-grained spatial structure. After cropping, many regions lost semantic coherence, that is, either they became noisy, or just resulted in flat depth profiles that undermined supervision quality. These low-resolution crops sampled uninformative patches like blank walls, floors, etc. leading to poor gradient signals and slow convergence during training. The final results we gained by training our model on NYUv2 dataset and inferring it on unseen data is shown in Fig. 3.

In Table 4, please note that we are not comparing inference time as both models were tested on different GPUs, and hence comparing their inference time would not be justified in this case.

These challenges led us to pivot toward synthetic data generation using CIFAR-10 and Distill Any Depth (as described in Table 4), as this dataset has more diverse composition of object-centric scenes and natural textures turned out to be better aligned with our low-resolution image input constraint.

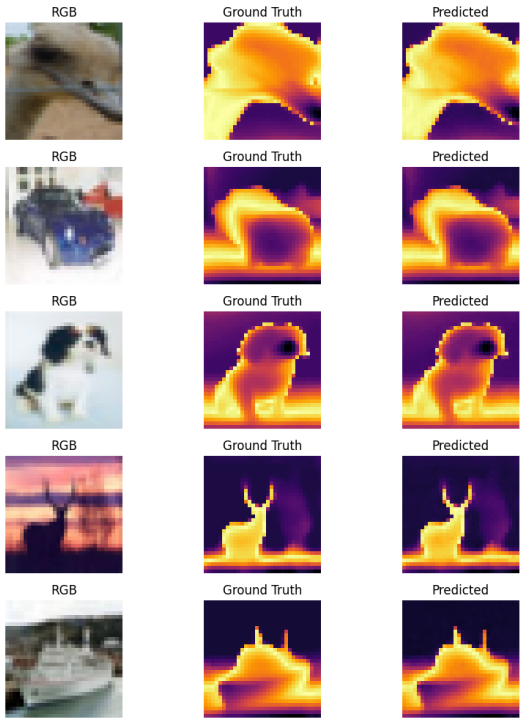

Figure 4: **Residual Block ✓, Transformer Encoder ✗.** The residual block preserves local spatial details and improves convergence, resulting in sharper object boundaries despite lacking global context.

A.2 More Experiments

Throughout our ablation study, we try to understand how each layer is impacting the final output and what hyperparameters can be tuned accordingly for best performance. We observe that the residual block improves convergence stability and preserved pixel-level details as seen in Fig. 4. On the other hand, the transformer module captures coherent global geometry/context, and sharp depth boundaries. We can see this in Fig. 5, as even though we get slightly blurred outputs, we do get global information about the presence of some objects.

Moreover, when we removed both (residual blocks and transformer-encoder), and observed that the model output was more blur and inaccurate at several places (as seen in Fig. 6).

Figure 5: **Residual Block ✗, Transformer Encoder ✓.** The model captures global structure and object layout but lacks fine-grained precision, leading to slightly blurred predictions.

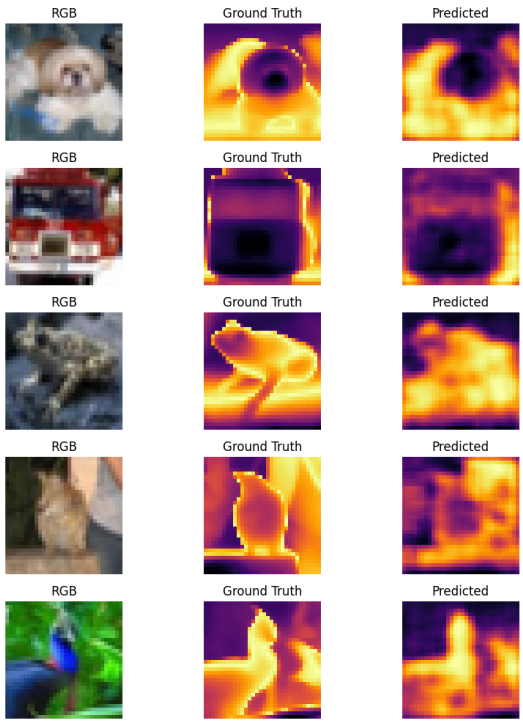

Figure 6: **Residual Block ✗, Transformer Encoder ✗.** The result is significantly degraded, with blurred predictions and poor object separation, indicating the importance of both modules.