ZoeDAM: Zero-Shot Metric Depth Anything Model

Justin Smith
University of Michigan
jbusmith@umich.edu

Nathan Yap
University of Michigan
nyap@umich.edu

Wensong Hu
University of Michigan
umhws@umich.edu

Yihang Liu
University of Michigan
yihangl@umich.edu

Abstract

Metric depth estimation, or metric depth prediction, is the process of estimating the distance between objects and the camera viewpoint. This is a critical component in autonomous driving, robot navigation, 3D reconstruction, augmented reality, and many other applications. This paper focuses on the monocular method, which uses a single RGB camera and improves the efficiency of prediction and the accuracy of metric depth by combining the advantages of DenseDepth, ZoeDepth, and DepthAnthing.

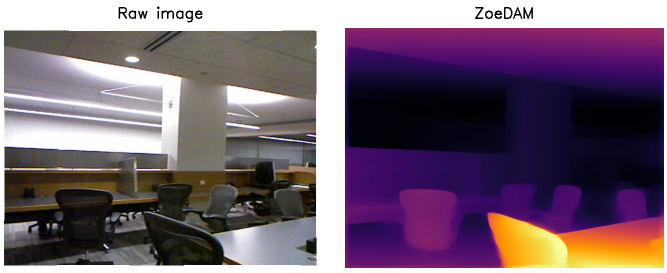

Figure 1. **Metric Depth Map Result of ZoeDAM**, input RGB images, output depth map

1. Introduction

Metric Depth estimation is a process of determining the absolute distance from the viewpoint to a surface. With the depth of information, the computer can perform a basic 3D reconstruction, continuously refreshing the 3D reconstruction helps robots to sense, navigate through, and interact with the surroundings. Many different types of sensors can be used for this purpose, and this paper chooses to employ a monocular RGB camera, the most affordable and common one. However, most of the work previously [1] [12] [18] [20]are focusing on the relative depth prediction without addressing the real world metric scale.

One of the SOTA works focusing on metric depth is ZoeDepth [3]. It notes that many current models are optimized for particular datasets, they either handle metric depth with absolute physical units or relative depth or depth consistency across pixels without absolute scale, but not both effectively. ZoeDepth model uses a hybrid approach to estimate both absolute and relative depth. ZoeDepth commenced with a preliminary training phase for relative depth across a range of datasets, after which the absolute depth was subjected to further refinement.

Depth Anything Model (DAM) [20] solves the dilemma between generalization and accuracy, by introducing the teacher-student training structure which involved semantic preservation loss. With this distillation-like training scheme, DAM still provides impressive predictions with unseen scenes.

This paper aims to combine the advantages of ZoeDepth and DAM so that the model estimate real world scale depth with metric that can use unlabeled data for training. We first introduce ZoeDepth by replaced the visual encoder in the ZeoDepth with DINOv2 and initialize the weight using DepthAnthing pre-trianing. Additionally, we introduce ZoeDAM+ attempted to challenge predict metric depth on unlabled data by adopt DAM training scheme to ZoeDAM.

2. Related Work

Monocular metric depth estimation. Monocular metric depth estimation (MMDE) predicts the absolute depth from camera origin to object from a single image with an RGB camera, this paper is focused on an RGB camera. Monocular depth can obtain metric depth [11] [5] [7] or relative depth [4] [6] [13] . Metric depth also known as absolution depth, predicts the distance between objects and the viewpoint of the camera with physical units and requires camera calibration with intrinsic and extrinsic parameters and more robust labeled datasets. Instead of precise dis-

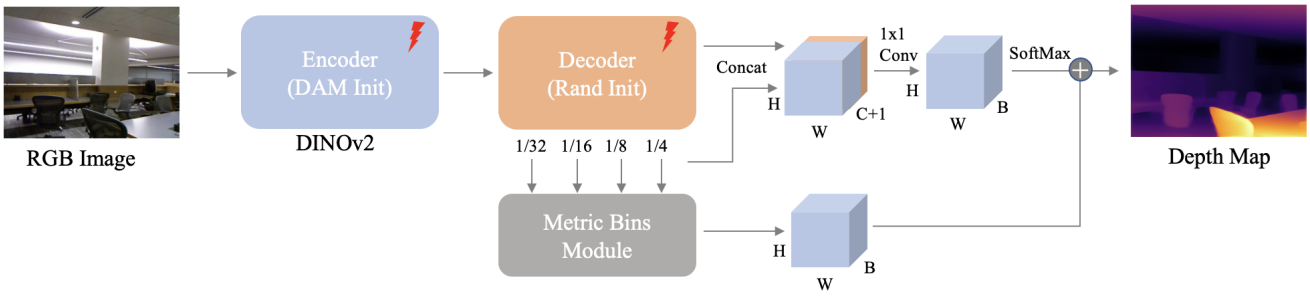

Figure 2. **ZoeDAM model structure**. We utilizes the same model structure as ZoeDepth, but change the MiDaS encoder to DINOv2 and initialize it with Depth Anything Model (DAM) DINOv2 checkpoint. The decoder is the same as ZoeDepth and DAM, which is DPT decoder, and initialized randomly

tance, relative depth focuses on distinguishing depth hierarchy among pixels, one pixel is closer to the viewpoint than another pixel (or objects, then object segmentation is required).

Zero-shot depth estimation. Traditional depth estimation requires intensive labeling, pairing images and depth map for training. Zero-shot depth estimation [3] uses transfer learning and unsupervised learning to generalize the usage of models, leveraging some form of auxiliary information to bridge the gaps between seen and unseen scenes [17] [8]. Depth cues can vary dramatically between different scenes and lighting conditions, making it difficult for a model to generalize well without direct experience. The abstract nature also causes lower accuracy than traditional methods.

Encoder-decoder for depth estimation. The encoder is typically a convolutional neural network (CNN) [1] that processes the input image. Its role is to extract and compress spatial hierarchies of features from the raw pixels. As the network depth increases, the spatial resolution decreases, while feature complexity and abstraction increase. This part of the network effectively reduces the dimensionality of the input, capturing the essence of the data in a lower-dimensional latent space. The decoder part gradually reconstructs the target output from the encoded representation. It typically uses up-sampling techniques such as transposed convolutions or up-sampling layers followed by convolutions, gradually increasing the spatial resolution while refining the details of the output. The decoder produces an output that matches the target dimensions, which can be a depth map for depth estimation. The SOTA encoder-decoder structure has shifted to vision transformer based model [4] [3] [20]. These depth prediction model utilize pre-trained visual feature encoder and randomly initialized decoder to achieve the same effect, but with better quality and efficiency.

3. Methodology

We adopt the Depth Anything Model (DAM) [20] training approach for ZoeDepth [3], which involves using a refined version of ZoeDepth with the encoder modified to DINOv2 [15] and initialized using the DAM pre-trained model as the teacher model, instead of the relative depth estimation model. The teacher model trained on labeled data will be referred to as ZoeDAM, while the student model trained on unlabeled data will be called ZoeDAM+. The first section is baseline reproduction, the second section is ZoeDAM model overview, and the third section introduced the ZoeDAM+ structure.

3.1. Baseline Reproduction

3.1.1 Average Depth Baseline

One simple baseline we generated from this dataset was the average depth map for a random subset of images. This average depth map was generated using 400 randomly selected scenes, where each scene's depth map was added together and averaged to generate a bare-minimum baseline to compare our generated model to Fig. 5.

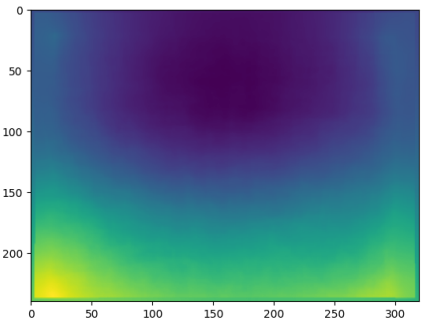

Figure 3. Average depth map generated from subset of images

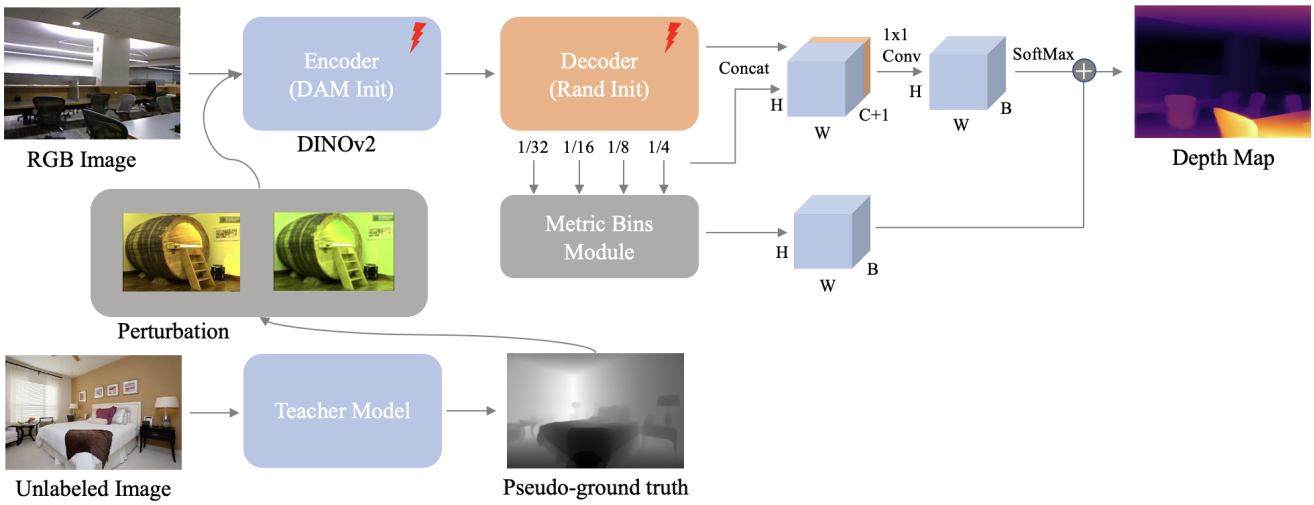

Figure 4. **ZoeDAM+ model structure**. The Teacher Model T is used to train a new student model S with perturbed unlabeled images.

3.1.2 DenseDepth Baseline

Another baseline included in our project is the Encoder-Decoder architecture described in the DenseDepth paper [1, 16]. This paper contributes a Unet architecture which leverages transfer learning for its encoder; specific methods for computing loss and augmenting data; and a specific use case in photo-realistic indoor scenes.

3.1.3 Encoder Pretrained Weights

Within our specific use case, we were using the off-the-shelf DenseNet169 encoder from the Torchvision library [10]. As the DenseDepth pretrained model was no longer available online, we had to settle with training our model from scratch using the default encoder weights.

3.1.4 Data Formatting and Augmentation

Using the database file downloaded from NYU Depth Dataset [14], which included 1449 images and their ground-truth depths, we split the data into 400 test images and 1049 training images. These images were then transformed with random horizontal flips and channel swaps as described in the DenseDepth paper and implemented GitHub repository [1]. These images were then normalized to range from 0 to 255 using numpy [9] to fit within unsigned 8-bit integer tensors and loaded into custom Pytorch Dataloader classes [16] which we reimplemented ourselves to pass into DenseDepth's model architecture.

3.2. ZoeDAM

ZoeDAM is the combination of two model ZoeDepth [3] and Depth Anything Model (DAM) [20], where ZoeDepth is the SOTA model in metric depth estimation and DAM is SOTA model in relative depth estimation.

The original DAM [20] model utilized the teacher-student training scheme that is similar to distillation to empower the model to performe better in zero-shot scenario. However, their work is primely focused on relative depth estimation. Thus, we intend to apply the DAM training approach to ZoeDepth, which means we want to use the fine-tuned ZoeDepth which the encoder is changed to DINOv2 [15] and initialized with DAM pre-trained model as the teacher model instead of the relative depth estimation model. We are calling the teacher model that trained on the labeled data ZoeDAM, and the student model that trained on the unlabeled data ZoeDAM+.

This subsection introducing the structure of ZoeDAM, and the evaluation results are given in Section 4.

3.2.1 Model Framework

An overview of the structure of ZoeDAM model is shown in Fig. 2. We use the MiDaS [4] and ZoeDepth [3] training approach for encoder-decoder (relative depth estimation) training, specifically, MiDaS is using Vision Transformers for Dense Prediction (DPT) [1] encoder-decoder where the encoder and decoder can be replaced and modified as demand.

Specifically, we modified the encoder part with DINOv2 [15], which is a more sophisticated transformer based robust visual feature encoder, and initialized with pre-trained checkpoint that has been provided in Depth Anything Model. For decoder, we use the defualt DPT decoder and initialized randomly.

ZoeDepth introduced Metric Bin Module, which receives multiscale features from the DPT decoder and pre-

dicts the centers of bins that will be used for metric depth prediction. Unlike methods that begin with a limited number of bins at the bottleneck and divide them later [2], this module predicts all bin centers at the bottleneck stage and modifies them in later decoder layers [3]. The output of Metric Bin Module is $\mathbb{R}^{H\times W\times B}$ dimension.

Additionally, the model takes the before-bin features and relative depth decoder output, and concatenate them to produce a $\mathbb{R}^{H\times W\times C+1}$ feature that is passed through a 1×1 Convolution layer and output a $\mathbb{R}^{H\times W\times B}$. This feature will be added with Metric Bin Module after a SoftMax function, and produce final depth map result.

Figure 5. Example of CutMix performed on an image.

Dataset	Indoor	Outdoor	Label Method	# Images
Labeled Datasets				
NYUv2	✓	✓	Microsoft Kinect	1449
Unlabeled Datasets				
BDD100K		✓	None	10000
LSUN	✓		None	52945

Table 1. Overview of Datasets

3.3. ZoeDAM+

Using ZoeDAM's combination of ZoeDepth and DAM, ZoeDAM+ tries to challenge the limitations of conventional datasets for Depth Perception with the introduction of raw, unlabeled images. To follow DepthAnything's methodology, we curated a small custom unlabeled dataset from LSUN and BDD100K that represents indoor and outdoor scenery. Using the swapped encoder from ZoeDAM, we create a teacher model T which is solely trained with labeled depths from NYUv2. Details regarding the size of these subset datasets are listed in Table 1. We then utilize model T to assign pseudo-depth labels for the small custom unlabeled dataset. Finally, we wanted to train a student model S on the combination of labeled and pseudo-labeled sets to witness any value in unlabeled images in enhancing data coverage. As mentioned in DepthAnything's paper, there is no improvement solely from unlabeled images because the extra knowledge acquired from unlabeled data is limited due to the shared architecture and structure used.

We then utilized the idea of strong perturbations to unlabeled images during training so that the student model S gains invariant representations of the world. The first form of perturbation was applied to all images with strong color jittering and Gaussian blurring. We then utilized CutMix – a technique that patches together segments of different images to stimulate and encourage the student model to learn depth cues in a wide range of contexts. We provide a 50 percent occurrence for each unlabeled image to perform a CutMix on its perturbed image.

3.3.1 Additional Framework Adjustments

Details provided about modifications to the model are shown in Figure 4. Any unlabeled images fed into the teacher model remain clean without any distortions or modifications. The prediction will then be used as a ground truth when fed with the unlabeled image into the student model S.

4. Experiments and Results

4.1. Dataset

For all model training and evaluation a subset 1449 image from the NYU Depth Dataset V2 [14] was used. From this dataset, 400 images were heldout for testing and model evaluation, while 1049 images were used for any model training. This dataset consisted of a series of indoor RGB images of bedrooms, dining rooms, kitchens, offices, classrooms, and bathrooms. Each paired with a groundtruth depth map generated using the Microsoft Kinect.

4.2. Baseline Reproduction

4.2.1 Training

To train the full encoder-decoder architecture, we used an Nvidia 1660 ti GPU which trained for 20 epochs over 3 hours. Our loss function was the one included within the DenseDepth paper, which was a combination of L1 loss between depth and ground-truth, L1 loss of directional gradients, and the structural similarity (SSIM) metric [19]. The full equation is rewritten in equation (1) directly taken from the DenseDepth paper which we reimplemented in our train function.

$$\mathcal{L}(y,\hat{y}) = \lambda L_{\text{depth}}(y,\hat{y}) + L_{\text{grad}}(y,\hat{y}) + L_{\text{SSIM}}(y,\hat{y}) \quad (1)$$

Our custom training function utilizes our custom dataloader class and saves checkpointed model files for ease of ac-

cess in testing. Figure 6 shows several sets of images that demonstrate the depth estimation of our trained model.

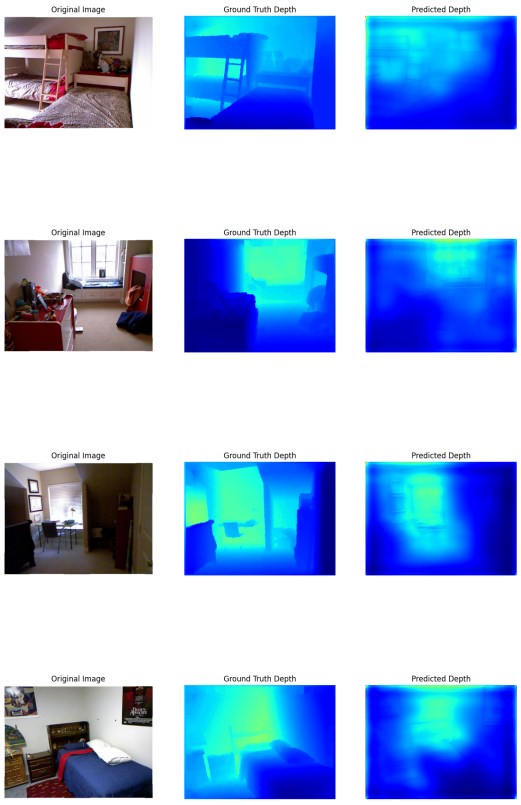

Figure 6. Custom trained DenseDepth model results (from left to right: original RGB image, ground-truth from NYU Depth, predicted depth from our model)

4.2.2 Result

Our baseline DenseDepth outperformed the average depth map as shown in table 2. However, the depth predictions are not as defined and consistent compared to the performance from the DenseDepth paper [1]. As seen in figure 6, the general depth of areas within each image are correct, however the features are cloudy and sometimes overlap compared to the ground-truth depths. This is likely due to the limited amount of training used to create the custom model and the fact that we were starting from scratch without pretrained weights for the decoder. However, this model still outperforms our baseline average depth map by a large margin and can serve as a baseline comparison of conventional methods to ZoeDAM.

Method	AbsRel	RMSE	log10
Average Depth	228.077	228.081	1.751
Custom DenseDepth	**1.079**	**1.235**	**0.103**

Table 2. Baseline model performance on NYU Depth subset

Method	AbsRel	RMSE	log10
ZoeDepth	0.075	0.270	0.032
ZoeDAM-L (Ours*)	**0.056**	**0.206**	**0.024**
ZoeDAM-B (Ours)	<u>0.061</u>	0.254	<u>0.029</u>
ZoeDAM+ (Ours)	0.287	<u>0.219</u>	0.510

Table 3. ZoeDAM Quantitative Results. **Bold** indicates the best result, <u>underline</u> indicates the second best result. The model marked with star (*) mean this is also implemented in DAM paper, but we re-implemented, trained, and evaluated with our code

4.3. ZoeDAM

4.3.1 Training

After the model was built and initilzed with DINOv2 pretrain checkpoints, it was further fine-tuned the encoder and decoder with NYU Depth V2 dataset with ground truth label on 2 NVIDIA RTX4090. For input and output setting, image size is $\mathbb{R}^{392\times518}$. For optimizer hyper-parameters, we set the batch size as 4, learning rate as 0.00016 with weight decay regularization of 0.01. The whole model is trained for 5 epochs using 7.5 hours.

During training of ZoeDAM, the scale-invariant pixel-level log loss is used, the equation can be written as:

$$\mathcal{L}(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}\left(\log\hat{y}_i - \log y_i - \frac{1}{n}\sum_{j=1}^{n}(\log\hat{y}_j - \log y_j)\right)^2 \tag{2}$$

Where, y represents the ground truth depth; $\hat{y}$ represents the predicted depth; n is the number of pixels in the image.

4.3.2 Results

We evaluate the trained model on NYU Depth V2 test set, and the quantitative result is given in Table. 3

Additionally, we provided some qualitative results in Fig. 7 and Fig. 1.

The quantitative evaluation shown in Table 3 demonstrates the a advancements of our ZoeDAM models. Relative to the DenseDepth model baseline, both ZoeDAM-L and ZoeDAM-B—demonstrate shows better performance across all evaluated metrics. The introduction of ZoeDAM-L, surpasses the improvements of ZoeDepth, achieving the

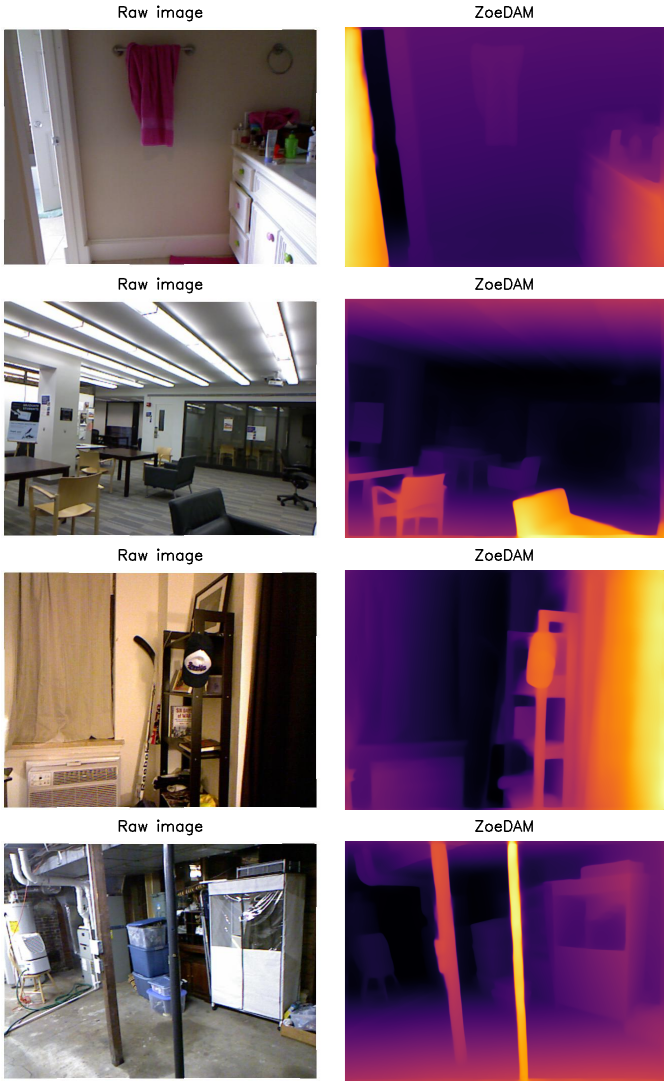

Figure 7. **Metric Depth Map Result of ZoeDAM**, input RGB images, output depth map. Tested on NYU Depth V2 for ZoeDAM.

lowest AbsRel at 0.056, an RMSE of 0.206, and a log10 of 0.024, indicating a substantial enhancement over it. The ZoeDAM-B variant, while not excelling past the L variant, still evidences considerable performance advancements over both the baseline and ZoeDepth. This is rather reasonable because ZoeDAM-B using the base version of DINOv2. These findings emphasizing the potential and robustness of our proposed methodology for metric depth estimation tasks.

4.4. ZoeDAM+

To further improve ZoeDAM, this model maintains DINOv2 pre-trained checkpoints as before. All parameters such as learning rate (0.00016) remain the same, but due to limitations, the model's batch size was decreased to 2. This model was trained for 1 epoch in approximately 45 minutes. Affine-invariant loss is used to ignore depth scales and shifts that may occur due to the different datasets used when creating the unlabeled dataset.

$$\mathcal{L}_u = \frac{\Sigma(M)}{HW}\mathcal{L}_u^M + \frac{\Sigma(1-M)}{HW}\mathcal{L}_u^{1-M} \tag{3}$$

The unlabeled dataset loss is a combination of unlabeled images with CutMix and without. M refers to a binary mask as a rectangular region of 1's.

$$L_u^M = p\left(S(u_{ab}) \odot M, T(u_a) \odot M\right), \tag{4}$$

$$L_u^{1-M} = p\left(S(u_{ab}) \odot (1-M), T(u_b) \odot (1-M)\right) \tag{5}$$

The equations above refer to the unlabeled loss obtained by computing affine-invariant losses p on valid regions defined by M and 1 - M.

4.4.1 Results

The quantitative result is given in Table 3. The results show a decrease compared to ZoeDAM. Deep learning models require larger training duration and epochs, and a single epoch was not sufficient for the model to effectively capture any necessary features from the training data. Another possibility for error is CutMix's reliability. Since there was not much fine-tuning involved when implementing the perturbation step, the unlabeled data may have introduced too much ambiguity. Since the model relies heavily on unlabeled data over labeled data, the ratio may have been too lob-sided which results in an inaccurate depth estimation.

5. Conclusion

In our project we explored various methods for monocular depth estimation from single RGB images. We utilized the NYU depth V2 dataset [14] to develop a baseline using DenseDepth's Unet model architecture. Which we then improved upon utilizing concepts from ZoeDepth [3] and DepthAnything [20]. From these experiments we found that our ZoeDAM combination model outperformed our baseline and improved upon ZoeDepth's base performance. We also investigated the use of raw unlabeled images to train depth prediction models with ZoeDAM+. A high log10 value indicates an issue at certain depths, and our CutMix integration was a high culprit for this issue. Given more time we would hope to train our ZoeDAM+ model longer and fine-tune perturbation values. We would also increase the amount of labeled data so the student model can learn proper values over solely challenging images.

References

[1] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *CoRR*, abs/1812.11941, 2018. 1, 2, 3, 5

[2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Localbins: Improving depth estimation by learning local distributions, 2022. 4

[3] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth, 2023. 1, 2, 3, 4, 6

[4] Reiner Birkl, Diana Wofk, and Matthias Müller. Midas v3.1 – a model zoo for robust monocular relative depth estimation, 2023. 1, 2, 3

[5] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild, 2017. 1

[6] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation, 2018. 1

[7] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation, 2019. 1

[8] Vitor Guizilini, Igor Vasiljevic, Dian Chen, Rares Ambrus, and Adrien Gaidon. Towards zero-shot scale-aware monocular depth estimation, 2023. 2

[9] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020. 3

[10] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. 3

[11] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks, 2016. 1

[12] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation, 2021. 1

[13] Alican Mertan, Damien Jade Duff, and Gozde Unal. Relative depth estimation as a ranking problem, 2020. 1

[14] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 3, 4, 6

[15] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. 2, 3

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. 3

[17] Saurabh Saxena, Junhwa Hur, Charles Herrmann, Deqing Sun, and David J. Fleet. Zero-shot metric depth with a field-of-view conditioned diffusion model, 2023. 2

[18] Joshua Luke Thompson, Son Lam Phung, and Abdesselam Bouzerdoum. D-net: A generalised and optimised deep network for monocular depth estimation. *IEEE Access*, 9:134543–134555, 2021. 1

[19] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 4

[20] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 1, 2, 3, 6