Optimizing Depth Anything V2 for Real-Time Depth Estimation via Quantization and Pruning

Vorrapard Kumthongdee Qianru Zhang Jay Roy Anuj Attri
New York University
{vk2584, qz2432, jgr7704, aa11527}@nyu.edu

Abstract

This study explores optimizing Depth Anything V2 [29], a state-of-the-art MDE model, for resource-constrained environments. While the model excels in relative depth prediction, adapting it to metric depth using the NYU Depth V2 dataset [21] revealed challenges due to pseudo-labeled training data and labeling inconsistencies. Dynamic `INT16` quantization reduced the model size by half while maintaining global depth prediction accuracy, with minor losses in detail. Additionally, unstructured pruning was applied at different sparsity levels, revealing that lower sparsity (10%–20%) preserved prediction quality, while higher sparsity (above 40%) significantly degraded both visual and quantitative performance. These experiments highlight the trade-offs between model compression and accuracy. Future work will focus on integrating quantization and pruning to develop a compact, efficient MDE model suitable for real-world deployment. Project Repository: https://github.com/RubyQianru/Depth-Anything-V2-Mini/

1. Introduction

Monocular depth estimation (MDE) is an important area in computer vision that focuses on obtaining depth information from a single image. Unlike stereo depth estimation, which requires images from multiple cameras along with precise calibration and synchronization—leading to increased system complexity and cost [14]—MDE works without any additional setup. MDE models have a variety of applications, including robotics [12], autonomous driving [30], augmented reality (AR) [19], and 3D reconstruction [17]. Many of these applications require real-time performance, especially on devices with limited resources, such as smartphones and embedded systems. However, while state-of-the-art models like Depth Anything V2 [29] provide highly accurate depth predictions, their high computational requirements make them unsuitable for real-time use in resource-constrained environments.

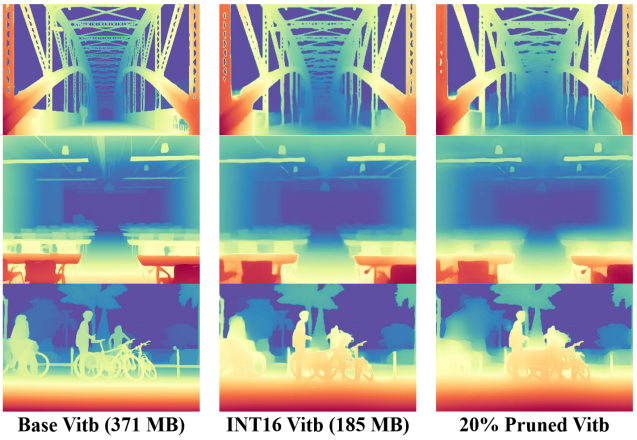

Figure 1. The quantized model and the pruned model lose only a small amount of fine-detail capability compared to the base model.

Early approaches to monocular depth estimation relied on traditional computer vision techniques such as structure-from-motion (SfM) [24] and shape-from-shading [13]. With the rise of deep learning, convolutional neural networks (CNNs) have greatly improved depth estimation by learning from large datasets of paired RGB and depth images [3, 17, 20]. State-of-the-art CNN models have shown strong performance in both supervised [5] and self-supervised [1] learning. More recently, Transformers [27] and Vision Transformers (ViT) [2] have pushed the field even further. Transformers are effective at capturing long-range dependencies, which is particularly useful in complex environments. Many of the latest state-of-the-art monocular depth estimation models are transformer-based, including Dense Prediction Transformer (DPT) [23], Marigold [11], and Depth Anything V2 [29].

While transformers offer excellent scalability and high performance [10], they also have drawbacks, including high complexity, significant computational requirements, and large model sizes [25, 26], which make them challenging to use on resource-limited devices such as embedded systems.

This project focuses on optimizing the transformer-based monocular depth estimation model, Depth Anything V2, to improve its efficiency and make it suitable for deployment on such devices. We will use techniques like static and dynamic quantization, as well as unstructured pruning with varying levels of sparsity, to reduce the model's size and computational needs.

The benchmark for this study is the NYU Depth Dataset V2 [21], a well-known dataset with ground truth depth labels. We will measure the performance and accuracy of each model variation using metrics such as MAE, RMSE, and threshold accuracy. Our ultimate goal is to create a lightweight version of Depth Anything V2 that can run in real-time on resource-constrained platforms with minimal loss of accuracy.

2. Related Work

2.1. Monocular Depth Estimation

In monocular depth estimation (MDE), models are generally categorized as either discriminative or generative. Discriminative models use supervised learning methods, often relying on convolutional neural networks (CNNs) or Vision Transformers (ViT), to directly map input RGB images to depth maps by minimizing prediction errors compared to ground-truth data [16, 23]. However, these models require large annotated datasets for training and may struggle to generalize well in diverse or highly dynamic environments [18]. On the other hand, generative models use techniques like generative adversarial networks (GANs) or variational autoencoders (VAEs) to capture the underlying distribution of depth data, allowing them to create detailed and realistic depth maps. Generative approaches are better at capturing fine details, handling transparent objects, and dealing with reflections, which are often challenging for discriminative models [15]. However, they are computationally demanding and involve more complex training processes to ensure stability. Emerging hybrid models [4, 6] combine the strengths of both discriminative and generative methods, improving robustness and accuracy in depth estimation. By integrating direct depth prediction with probabilistic depth generation, these models provide more reliable and comprehensive depth estimation for various applications.

2.2. Depth Anything models

Depth Anything v2 [29] represents one of the most recent state-of-the-art monocular depth estimation models, building upon the foundational capabilities of its predecessor, Depth Anything v1 [28]. The model combines state-of-the-art convolutional neural networks (CNNs) and transformer-based architectures to achieve higher precision in depth prediction. Both V1 and V2 employ a knowledge distillation technique [8], where a teacher model generates pseudo-labels to train the student model. However, Depth Anything V2 introduces significant enhancements by training the teacher model exclusively on synthetic images, replacing the use of labeled real images that were utilized in V1. Additionally, V2 features a larger teacher model and incorporates a bridge of large-scale pseudo-labeled real images, which further bolster its training process and performance. These improvements enable Depth Anything V2 to achieve approximately 10x faster processing speeds and higher accuracy compared to other Stable Diffusion models [11].

The proposed method utilizes a pseudo-labeling approach to advance monocular depth estimation by leveraging both synthetic and real-world data. Initially, a teacher model based on DINOv2-G [22] is trained exclusively on high-quality synthetic images from five precise synthetic datasets, totaling 595,000 images. This teacher model then generates accurate pseudo-depth labels for eight large-scale, unlabeled real-world datasets comprising 62 million images. These pseudo-labeled real images are subsequently used to train student models, enabling robust generalization without the need for extensive manual annotations. By producing precise pseudo-depth maps on large amounts of unlabeled real images, the method ensures that the final student models achieve high accuracy and versatility. Depth Anything V2 effectively combines the strengths of generative models like Marigold [11], which excel in capturing fine details, transparent objects, and reflections, with the discriminative capabilities of Depth Anything V1 [28], which adeptly handles complex scenes, efficiency, and transferability. This hybrid approach allows Depth Anything V2 to address and overcome the limitations of each individual model, resulting in superior performance, enhanced detail capture, and greater efficiency across a wide range of depth estimation tasks as shown on Figure 2.

Existing benchmarks for monocular depth estimation (MDE) face challenges such as noisy labels, limited diversity, and low resolution, reducing their reliability for modern high-resolution applications. To address these issues, the DA-2K benchmark [29] was introduced alongside the model, featuring sparse but precise depth annotations across diverse high-resolution scenes. By leveraging automated pipelines with expert model voting and human validation, DA-2K ensures accuracy and diversity, encompassing eight key application scenarios with 2,000 annotated pixel pairs from 1,000 images. Yang et al. report achieving 97% accuracy on DA-2K with their large model and 95% with their small model, significantly outperforming Marigold's [11] 87%. However, its performance on real-world datasets like NYU Depth V2, which focuses on dense indoor depth annotations, remains an open question. This study also focuses on evaluating the performance of Depth Anything V2 on the NYU Depth V2 benchmark to confirm the practical applications of the model.

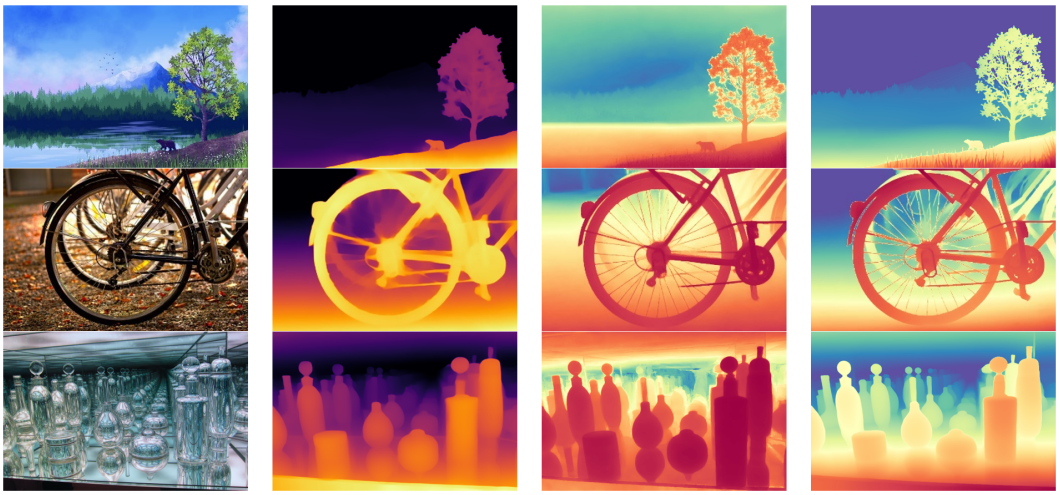

Figure 2. Performance comparison between Depth Anything V1 [28], Marigold [11], and Depth Anything V2 [29].The V2 model shows a robust fine detail, transparent, and reflection property in a complex scene, improving from the V1 model and overcomes the Marigold model.

2.3. Model Optimization Techniques for Efficiency

Pruning is a method for reducing the size of deep learning models by carefully removing unnecessary weights and neurons. By identifying and removing parts of the network that contribute little to its performance, pruning makes neural networks smaller and less complex without greatly affecting their accuracy [7]. This process also reduces memory usage and speeds up inference, making pruned models more practical for use on devices with limited resources, such as smartphones and embedded systems. For monocular depth estimation, pruning helps advanced models like Depth Anything V2 [29] run in real-time by lowering computational requirements while keeping their accuracy high.

Quantization is another important technique for reducing the size and computational needs of deep learning models. It works by converting the model's weights and activations from high-precision floating-point numbers to simpler formats like `int16` or `int8`. This reduces memory usage and speeds up computations, which helps save energy and make models faster to use [9]. Quantization is especially useful for running deep learning models on edge devices with limited resources. By balancing accuracy and efficiency, it enables real-time use of advanced monocular depth estimation models in environments where low latency is critical.

3. Methodology

To evaluate the Depth Anything V2 [29] model for use on resource-limited devices, we applied quantization and unstructured pruning to its three pretrained variants: 25M (vits), 98M (vitb), and 335M (vitl) parameters. These methods were tested systematically using the NYU v2 [21] dataset to examine the balance between model size, computational efficiency, and performance. During optimization, `.pth` checkpoint files were saved at different stages, enabling comparisons of model size, processing speed, and prediction accuracy.

3.1. Fine-tuning

One challenge faced during this study was the scale of the model's predicted depth values. Because Depth Anything V2 was trained on pseudo-labeled real images [29], it predicts relative depth values rather than the metric depths provided by the NYU Depth V2 dataset masks. To enable accurate comparisons and evaluations, we fine-tuned all model variants using the NYU Depth V2 dataset, selecting a subset of 500 training samples and 100 validation samples. Each image was resized to `224×224` pixels to ensure faster training, with a `batch size` of 4.

During fine-tuning, the model's backbone was frozen to maintain its generalization ability, while only the depth estimation head was trained. All model variants were fine-tuned for 10 epochs, with this duration optimized through cross-validation. After fine-tuning, the models were saved for further quantization and pruning steps.

3.2. Model Compression Techniques

Quantization was applied to the model's state dictionary, using both static and dynamic quantization techniques as follows:

- Static Quantization: In this method, all weights were permanently stored in the `torch.int16` format to reduce

memory usage. During model execution, the weights were converted back to the `float32` format for computations, ensuring the model's functionality was preserved while improving storage efficiency.

- Dynamic Quantization: This method performed on-the-fly conversion of weights during runtime to balance memory usage and computational speed. For `INT16`, weights were converted to `float16` at runtime and stored in the `float16` format. Similarly, for `INT8`, weights were converted to `qint8` and stored in the `torch.qint8` format, significantly reducing memory requirements. Dynamic quantization was applied selectively to layers that benefit most from it, such as `torch.nn.Linear` and `torch.nn.Conv2d`, ensuring efficient processing without substantial loss of accuracy.

Unstructed Pruning was applied to the `Linear` and `Conv2d` layers of the Depth Anything V2 model to reduce its size and computational complexity by removing weights with the smallest absolute values. Using the `torch.nn.utils.prune` module, pruning was performed at different sparsity levels from 10% to 90%, allowing us to analyze trade-offs between model efficiency and accuracy.

3.3. Dataset and Evaluation Metrics

The NYU v2 [21] dataset was used as the benchmark to evaluate the optimized models. This dataset includes high-quality depth estimation data with reliable ground truth labels, making it suitable for analyzing the impact of model compression techniques. The models' performance was measured using the following metrics:

1. **Mean Absolute Error (MAE):** Measures the average absolute difference between the predicted and actual depth values, indicating prediction accuracy
2. **Root Mean Squared Error (RMSE):** Evaluates prediction precision by penalizing larger errors more heavily, offering a comprehensive measure of quality.
3. **Threshold Accuracy (δ):** Calculates the percentage of predictions that fall within a set range of the ground truth depth values ($\tau = 1.25$).
4. **File Size (MB):** Compares the storage efficiency of the models by analyzing the size of the `.pth` checkpoint files at different optimization stages.

4. Result

4.1. Quantization

Table 1 presents the performance of the original and quantized versions of the Depth Anything V2 model on the NYU Depth V2 dataset. Static quantization produced poor results across all model variants, with outputs failing to capture meaningful depth information. These models generated blank images and were excluded from further analysis.

Table 1. Performance Metrics of Original and Optimized Depth Anything V2 Models on NYU Depth V2 Dataset

Model Variant	RMSE	δ (1.25)	Size
vits	2.18	15.38%	95 MB
vits_static_INT16	2.89	0%	47 MB
vits_dynamic_INT16	2.18	15.41%	47 MB
vits_dynamic_INT8	2.15	15.08%	34 MB
vitb	2.58	14.95%	371 MB
vitb_static_INT16	2.89	0%	185 MB
vitb_dynamic_INT16	2.58	14.95%	186 MB
vitb_dynamic_INT8	1.60	26.25%	128 MB
vitl	7.42	2.13%	1.24 GB
vitl_static_INT16	2.89	0%	639 MB
vitl_dynamic_INT16	7.43	2.13%	639 MB
vitl_dynamic_INT8	7.44	2.04%	415 MB

Dynamic quantization, on the other hand, showed mixed results. The dynamic `INT16` quantization method effectively preserved performance while reducing model size by nearly half. For example, the `vits_dynamic_INT16` variant maintained a similar RMSE and threshold accuracy compared to the original model but required significantly less storage. However, the dynamic `INT8` quantization displayed inconsistent performance. While some variants, such as `vitb_dynamic_INT8`, demonstrated improved accuracy, others, like `vitl_dynamic_INT8`, failed to maintain acceptable prediction quality. Figure 3 highlights the performance of the `dynamic_INT16` model, which achieves reliable depth predictions with global accuracy but lacks some fine details.

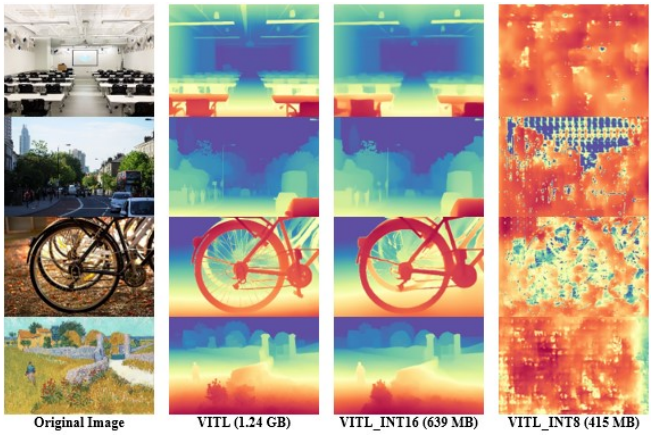

Figure 3. The dynamic `INT16` model, with half the size, shows a great overall performance. The model capture global depth, but missing a few fine-details.

4.2. Unstructured Pruning

The impact of unstructured pruning was evaluated across sparsity levels ranging from 10% to 90%. Figure 4 and Figure 5 illustrate the relationship between sparsity and the metrics MAE, RMSE, and threshold accuracy. Among all sparsity levels, 40% achieved the best trade-off in numerical evaluation, delivering a balance between reduced model size and maintained performance.

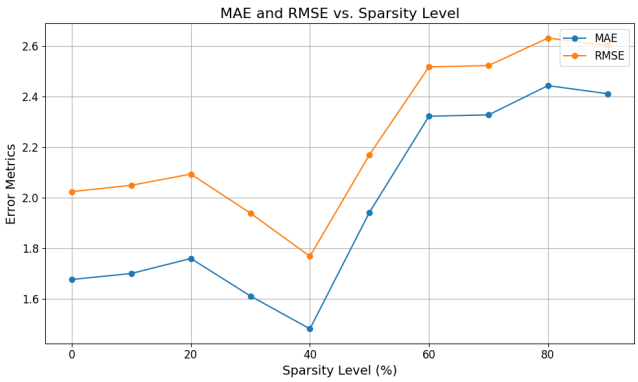

Figure 4. Relationship between sparsity levels (10%–90%) and error metrics (MAE and RMSE) for unstructured pruning. Lower sparsity levels demonstrate minimal error increase, while higher sparsity levels degrade performance.

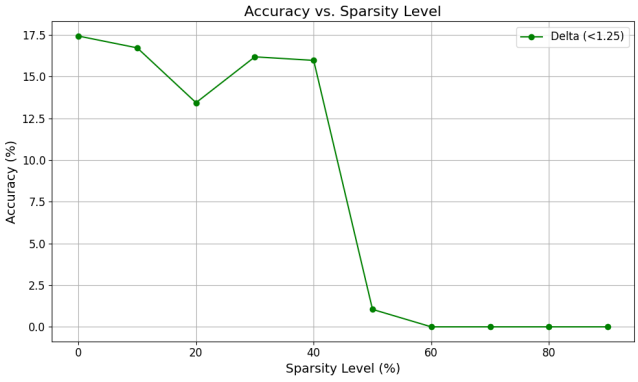

Figure 5. Threshold accuracy ($\tau = 1.25$) for different sparsity levels (10%–90%). Lower sparsity levels maintain higher accuracy, while aggressive pruning significantly reduces prediction reliability.

However, qualitative results revealed limitations at higher sparsity levels. As shown in Figure 6, the 40% pruned model produced blurry depth predictions, losing important details compared to the 10% and 20% pruned variants. This highlights the need for careful selection of pruning levels, as overly aggressive pruning can compromise visual quality despite favorable metric scores.

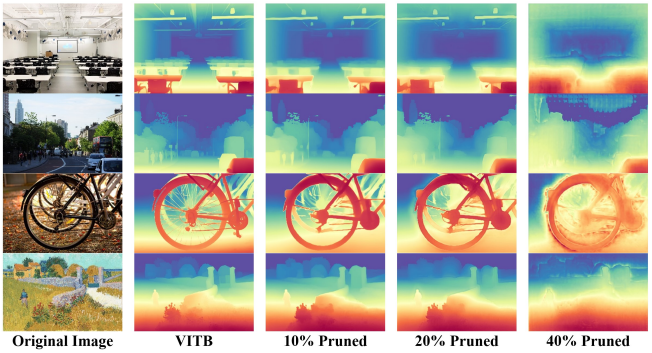

Figure 6. Qualitative comparison of depth predictions for pruned models at different sparsity levels. The 40% pruned model loses important details, resulting in blurry depth maps compared to the 10% and 20% pruned variants.

5. Conclusion

Depth Anything V2 [29] is a state-of-the-art monocular depth estimation model known for its excellent performance in predicting relative depth. However, fine-tuning the model to predict metric depth using the NYU v2 dataset [21] revealed several challenges. The model struggled to produce accurate metric depth predictions, resulting in high RMSE values and low accuracy. This limitation arises because Depth Anything V2 was primarily trained with pseudo-labeled data, making it difficult to scale its predictions to metric depth. Additionally, the NYU v2 dataset contains labeling errors along object boundaries, where some pixels incorrectly have zero ground truth depth. These inconsistencies further contributed to the errors when comparing Depth Anything V2 to state-of-the-art models, which predict depth for all pixels. This challenge highlights the importance of the DA-2K benchmark [29], created alongside Depth Anything V2 to better align with its design and capabilities.

Dynamic quantization using `INT16` demonstrated promising results in our experiments. By reducing the model size to half of its original, this method preserved strong global depth prediction performance, losing only minor fine details. These findings suggest that quantization can effectively optimize Depth Anything V2 for resource-constrained environments without significant accuracy loss.

Future work will focus on integrating quantization with pruning. Since pruning removes less significant weights by setting them to zero, it complements quantization and can further enhance compression efficiency. While model compression methods inevitably involve trade-offs between performance and accuracy, our goal is to identify the optimal balance point. This will allow us to develop a compact, robust monocular depth estimation model suitable for real-world applications, such as autonomous systems and embedded devices.

References

[1] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning, 2019. 1

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 1

[3] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. *Proceedings. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018:2002–2011, 2018. 1

[4] Ashkan Ganj, Hang Su, and Tian Guo. Hybriddepth: Robust metric depth fusion by leveraging depth from focus and single-image priors, 2024. 2

[5] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency, 2017. 1

[6] Ming Gui, Johannes S. Fischer, Ulrich Prestel, Pingchuan Ma, Dmytro Kotovenko, Olga Grebenkova, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. Depthfm: Fast monocular depth estimation with flow matching, 2024. 2

[7] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, 2016. 3

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 2

[9] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference, 2017. 3

[10] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. 1

[11] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation, 2024. 1, 2, 3

[12] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision?, 2017. 1

[13] Byungil Kim and Peter Burger. Depth and shape from shading using the photometric stereo method. *CVGIP: Image Understanding*, 54(3):416–427, 1991. 1

[14] Kai Yit Kok and Parvathy Rajendran. A review on stereo vision algorithm: Challenges and solutions, 2019. 1

[15] Aran CS Kumar, Suchendra M. Bhandarkar, and Mukta Prasad. Monocular depth prediction using generative adversarial networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 413–4138, 2018. 2

[16] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 239–248, 2016. 2

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

[18] Qing Li, Jiasong Zhu, Jun Liu, Rui Cao, Qingquan Li, Sen Jia, and Guoping Qiu. Deep learning based monocular depth prediction: Datasets, methods and applications, 2020. 2

[19] Aziz Makandar and Anita Patrot. An approach to analysis of malware using supervised learning classification. In *International Conference on Recent Trends in Engineering, Science Technology - (ICRTEST 2016)*, pages 1–5, 2016. 1

[20] Junhua Mao, Xu Wei, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan L. Yuille. Learning like a child: Fast novel visual concept learning from sentence descriptions of images. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2533–2541, 2015. 1

[21] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 1, 2, 3, 4, 5

[22] 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

[23] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction, 2021. 1, 2

[24] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846, 2006. 1

[25] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp, 2019. 1

[26] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey, 2022. 1

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 1

[28] 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. 2, 3

[29] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 1, 2, 3, 5

[30] Mert Burkay Çöteli, Orhun Olgun, and Hüseyin Hacıhabiboğlu. Multiple sound source localization with steered response power density and hierarchical grid refinement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(11):2215–2229, 2018. 1