DIP Project: Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data

Chen Pery , Roi Papo

1 Overview

The "Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data" paper by Lihe Yang *et al* (2024) introduces a groundbreaking approach to monocular depth estimation (MDE) by leveraging a vast dataset of 62 million unlabeled images. Traditional MDE methods rely heavily on labeled data, limiting their scalability and generalization to diverse environments. This paper addresses these limitations by developing a data engine that automatically collects and annotates unlabeled images, using a pre-trained MDE model to generate pseudo-labels. The training process involves an initial teacher model trained on labeled data and a student model trained on a combination of labeled and pseudo-labeled data, enhanced by data augmentation and feature alignment with DINOv2. The model demonstrates impressive zero-shot capabilities across six public datasets and sets new state-of-the-art performance with fine-tuning. The paper highlights the potential of using large-scale unlabeled data and advanced feature alignment to improve the robustness and generalization of depth estimation models.

2 Introduction to the problem

MDE is a fundamental task in computer vision, where the goal is to estimate the depth of each pixel in a single image. This task is crucial for various applications, including robotics, autonomous driving, and virtual reality. Traditional methods for depth estimation rely heavily on labeled datasets, which are created mainly by acquiring depth data from sensors, stereo matching or SfM which is costly, time-consuming, or even intractable in particular situations. This limitation hinders the scalability and generalization of MDE models to diverse and unseen environments.

2.1 Depth Sensing Techniques

2.1.1 Depth Sensors

- **Types**: Includes time-of-flight sensors, structured light sensors, and LiDAR. (Lidar - Light Detection and Ranging is a remote sensing method used to examine the surface of the earth)
- **Function**: Projects light (often infrared) onto the scene and measures the time it takes for the light to return or the pattern it forms.
- **Advantages**: Effective for surfaces without distinguishing features.
- **Challenges**: Can be expensive and have their own limitations, such as sensitivity to environmental conditions.

2.1.2 Stereo Cameras

- **Inspiration**: Mimics human binocular vision.
- **Setup**: Requires two cameras placed side by side, calibrated to know the exact distance between them.
- **Function**: Triangulates points that match in each image to estimate depth.
- **Challenges**: Calibration and matching points in featureless areas can be difficult.

2.1.3 Structure from Motion (SfM)

- **Function**: Uses multiple images taken from different viewpoints to estimate both camera positions and 3D structure.
- **Challenges**: Computationally expensive and struggles with scenes lacking distinct features (e.g., a plain white wall).

3 Objective

The objective of this work is to develop a foundation model for MDE that can produce high-quality depth information for any image under any circumstances. The approach focuses on scaling up the dataset using large-scale unlabeled data, which offers several advantages over traditional methods (like a depth sensors, stereo matching, or SfM). Specifically, monocular unlabeled images are:

- simple and inexpensive to acquire
- diverse enough to cover a wide range of scenes
- easy to annotate using a pre-trained MDE model (only takes a feedforward step)

This process not only ensures efficient data collection but also generates dense depth maps (than LiDAR) without the need for specialized equipment or computationally intensive processes.

Figure 1: the model exhibits impressive generalization ability across extensive unseen scenes, works robustly in low-light environments (1st and 3rd column), complex scenes (2nd and 5th column), foggy weather (5th column), and ultra-remote distance (5th and 6th column).

4 Related work

4.1 Monocular depth estimation (MDE)

4.1.1 Early Methods:

Traditional Computer Vision Techniques: Initial approaches relied on handcrafted features and explicit depth cues. These methods struggled with complex scenes involving occlusions and textureless regions.

4.1.2 Deep Learning-Based Methods :

Deep learning significantly advanced MDE by learning depth representations from carefully annotated datasets. Eigen et al.("Depth map prediction from a single image using a multi-scale deep network") introduced a multi-scale fusion network to regress depth, setting a new direction for the field. Subsequent research consistently improved depth estimation accuracy by transforming the regression task into a classification problem, introducing various priors, and developing better objective functions. Despite the promising performance, they are hard to generalize to unseen domains.

4.2 Zero-shot depth estimation

train a MDE model using a diverse training set to predict depth for any image. Pioneering efforts in this area collected more training images but provided sparse supervision. MiDaS made a significant contribution by using an affine-invariant loss(Affine Transformations include scaling, translation, and rotation of the image) to handle different depth scales and shifts across datasets, offering relative depth information. Recent methods have attempted to estimate metric depth but showed poorer generalization than MiDaS.

4.3 Leveraging unlabeled data

Leveraging unlabeled data through semi-supervised learning involves training models using both labeled and unlabeled data. While this approach is popular in various applications, it typically assumes a limited availability of labeled images.

4.4 DINOv2

DINOv2 is a self-supervised learning model known for its excellence in capturing rich semantic information without requiring labeled data. It provides nuanced, continuous features that offer a comprehensive understanding of scenes. When applied to monocular depth estimation (MDE), DINOv2's detailed semantic representations enhance the model's ability to estimate depth accurately. Unlike traditional auxiliary tasks like semantic segmentation, which convert images into discrete classes and may lose detailed information, DINOv2 maintains the rich, continuous semantic priors, leading to improved scene understanding and depth prediction. DINOv2 generalizes to a lot of tasks, including semantic segmentation, depth estimation, instance retrieval, video understanding, and fine-grained classification.

5 Description

5.1 Strategies in the Paper

5.1.1 Combining Labeled and unlabeled data

The authors design a data engine to collect and automatically annotate large-scale unlabeled data (62M images) from public datasets like SA-1B, Open Images, and BDD100K. An initial MDE model is trained on 1.5M labeled images(labeled with sfm, depth sensors or stereo), and then unlabeled images are annotated and jointly learned with the labeled images in a self-training manner. As obtaining large volumes of labeled data is challenging, the abundant unlabeled data can be utilized to enhance and supplement the labeled data.

5.1.2 Student model overcome Teacher model

The authors proposed challenging the student model with more difficult optimization targets during pseudo-label learning. They introduce strong perturbations to the unlabeled images during training, compelling the student model to actively seek extra visual knowledge and develop invariant representations. Two forms of perturbations are applied:

1. **Strong Color Distortions:** includes techniques like color jittering and Gaussian blurring.
2. **Strong Spatial Distortion:** Utilizes a method called CutMix, which involves spatially mixing different parts of images.

5.1.3 Using Semantic Segmentation benefits:

The author believes that arming the depth estimation model with such high-level semantic-related information is beneficial. an initial attempt by carefully assigning semantic segmentation labels to unlabeled images with a combination of models, then the authors observed that when an MDE model is already powerful, these tasks bring limited gains, therefore, instead of using auxiliary tasks, the authors propose maintaining rich semantic priors from DINOv2 using a feature alignment loss.

5.1.4 Feature Alignment:

a technique used in machine learning and deep learning to ensure that the features (representations) learned by one model are consistent with or aligned to the features learned by another model. This alignment process helps to transfer useful information between models and enhance the performance of the target task.

5.2 Training

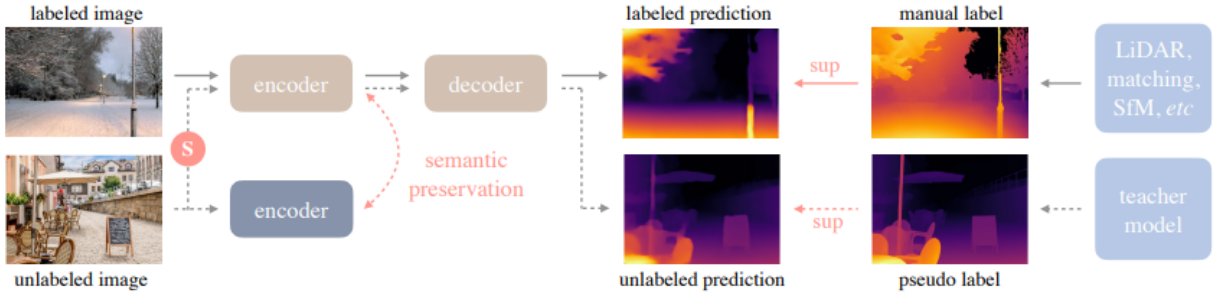

Figure 2: The pipeline. Solid line: flow of labeled images, dotted line: unlabeled images. We especially highlight the value of large-scale unlabeled images. The S denotes adding strong perturbations To equip our depth estimation model with rich semantic priors, we enforce an auxiliary constraint between the online student model and a frozen encoder to preserve the semantic capability

5.3 Learning labeled images:

denote the labeled and unlabeled sets as $D^l = \{(x_i, d_i)\}_{i=1}^M$ *and* $D^u = \{u_i\}_{i=1}^N$. A teacher model T is trained on the labeled dataset D^l and then used to assign pseudo-depth labels to the unlabeled dataset D^u. Finally, a student model S is trained on the combination of the labeled and pseudo-labeled datasets.

- In order to get a strong teacher model T, DINOv2 pre-trained weights are used to initialize the encoder, leveraging its strong semantic understanding (The encoder is the part of the model that processes the input data and extracts meaningful features from it.)
- Following prior works, instead of fine-tuning S from T, S re-initializes for better performance. This means starting S with new initial weights rather than the fine-tuned weights from T.

5.4 Unleashing the Power of Unlabeled Images:

$\mathcal{L}_l$	$\mathcal{L}_u$	$\mathcal{S}$	$\mathcal{L}_{feat}$	KI	NY	SI	DD	ET	DI
✓				0.085	0.053	0.492	0.245	0.134	0.070
✓	✓			0.085	0.054	0.481	0.242	0.138	0.073
✓	✓	✓		0.081	0.048	0.469	0.235	0.134	0.068
✓	✓	✓	✓	**0.076**	**0.043**	**0.458**	**0.230**	**0.127**	**0.066**

Table 9. Ablation studies of: 1) challenging the student with strong perturbations ($\mathcal{S}$) when learning unlabeled images, and 2) semantic constraint ($\mathcal{L}_{feat}$). Limited by space, we only report the AbsRel (↓) metric, and shorten the dataset name with its first two letters.

Initial attempts to simply combine labeled and pseudo-labeled images did not improve performance (see Figure 5.4), suggesting that the knowledge gained from this naive self-teaching method was limited.

the teacher model and student model share the same pre-training and architecture and they tend to make similar correct or false predictions. To address this issue, the authors proposed additional optimization challenges to the student during pseudo-label learning:

5.4.1 Strong Perturbations

The researchers introduced strong perturbations to the unlabeled images during the training process. These perturbations included color distortions (such as color jittering), Gaussian blurring, and other augmentation techniques such as CutMix. This approach forces the model to seek extra visual knowledge and develop robust representations (see Figure 3) .

Figure 3: example of the strong perturbations presented in the paper

5.4.2 Semantic Preservation

Considering that segmentation models excel at grouping semantically related pixels and that elements of the same object typically lie at similar distances from the camera lens, the authors devised a feature alignment loss function. This loss function penalizes predictions where the estimated depth representation deviates in angle from the segmentation representation produced by the pre-trained Dinov2 model- maintaining rich semantic priors.

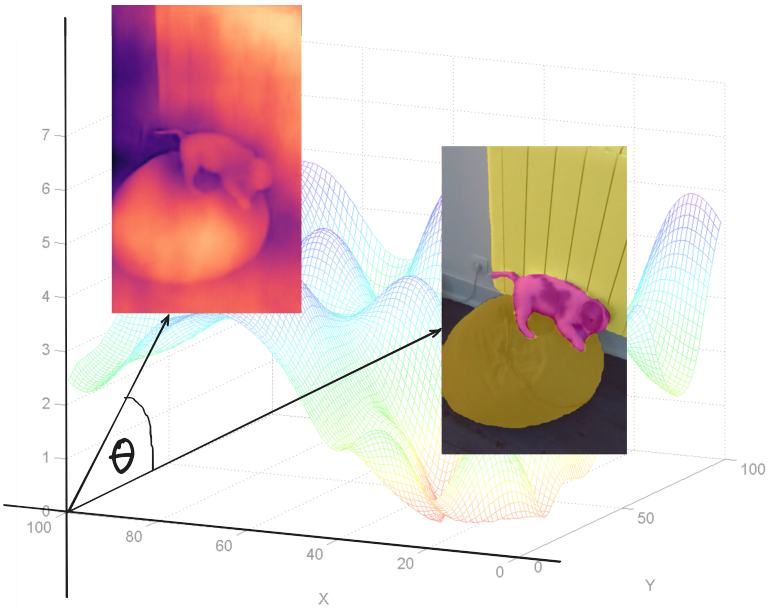

Figure 4: the distance in angle between depth vector representation and semantic representation shall stay small

the loss is given by:

$$\mathcal{L}_{\text{feat}} = 1 - \frac{1}{HW} \sum_{i=1}^{HW} \cos(f_i, f_i') \tag{1}$$

where $\cos(\cdot, \cdot)$ measures the cosine similarity between two feature vectors. f is the feature extracted by the depth model S, while f' is the feature from a frozen DINOv2 encoder. (see Figure 4)

6 Demonstration of the results

In this part, we aimed to evaluate and compare the results of a depth estimation model by replicating the experimental setup used by the original authors. To achieve this, we first needed to obtain the dataset used by the authors and preprocess it accordingly.

6.1 Dataset Acquisition and Preprocessing

To replicate the results, we acquired the NYU-V2 dataset.

6.1.1 About the NYU-V2 Dataset

The NYU-Depth V2 (NYU-V2) dataset is a widely used benchmark dataset for depth estimation tasks. It consists of RGB and depth image pairs captured using a Microsoft Kinect sensor. The dataset includes 1,449 densely labeled pairs of aligned RGB and depth images, along with 464 new scenes taken from a variety of indoor environments. The dataset is particularly valuable for training and evaluating depth estimation models due to its diversity and high-quality annotations.

6.1.2 Data Preparation

The dataset was provided as a MATLAB (.mat) file. Using a license from my professor's lab, I was able to open and extract the data. The next step involved preprocessing the data into image-depth (ground truth label) pairs.
The authors' code required the images to be separated into "inside" and "outside" folders. To achieve this, I found a file online that mapped each image to its respective class. For example, image `2221.png` is labeled as a bathroom, indicating it is an inside image, while image `4321.png` is labeled as a garden, indicating it belongs to the outside folder. Using this information, I wrote a script to parse and organize the data as required by the authors' code.

6.2 Model and Pretrained Weights

We obtained the pretrained weights of the model, which were trained on hardware and data volume that we could not replicate. These weights were integrated into our project, and the NYU-V2 data was passed through the model.

6.3 Evaluation Metrics

To measure the performance of the model, we used the following depth metrics:

6.3.1 Depth Metrics

- $\boldsymbol{\delta_1}$: The percentage of predicted depths where the ratio between the predicted and ground truth depth is within a threshold of 1.25.
- $\boldsymbol{\delta_2}$: The percentage of predicted depths where the ratio between the predicted and ground truth depth is within a threshold of 1.25^2.
- $\boldsymbol{\delta_3}$: The percentage of predicted depths where the ratio between the predicted and ground truth depth is within a threshold of 1.25^3.
- **abs_rel**: The mean absolute relative error between the predicted and ground truth depths.

- **rmse**: The root mean square error between the predicted and ground truth depths.
- **log_10**: The mean log10 error between the predicted and ground truth depths.
- **rmse_log**: The root mean square error of the logarithm of the predicted and ground truth depths.
- **silog**: The scale-invariant logarithmic error, which measures the difference between the predicted and ground truth depths while being invariant to scale.
- **sq_rel**: The mean squared relative error between the predicted and ground truth depths.

These metrics were presented in the original paper to facilitate a direct comparison of results.

6.4 Results and Discussion

We managed to achieve results that were very close to those reported by the authors (see Figure 5). The slight differences observed might be attributed to potential errors in the separation of inside and outside images or differences in the Pytorch version used.

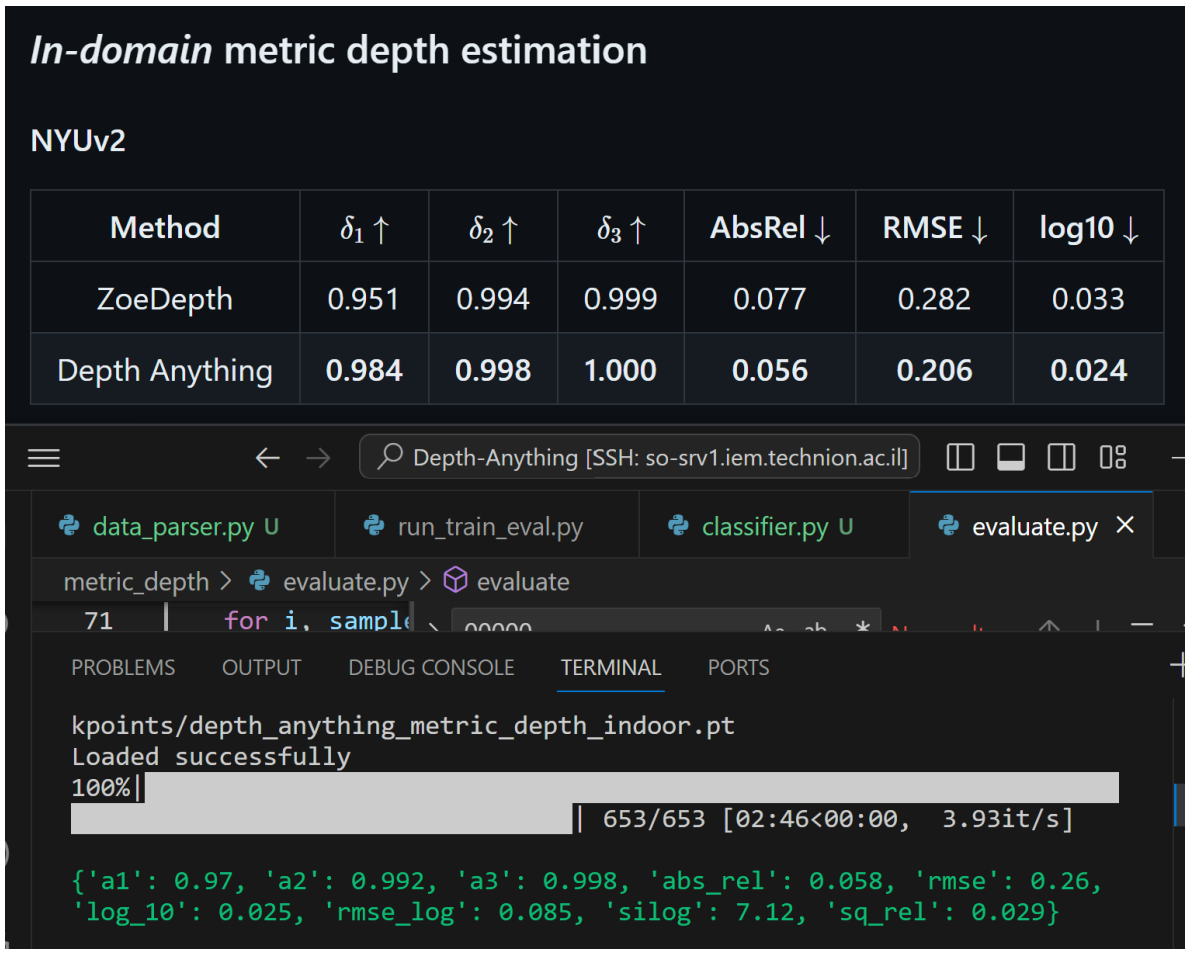

Figure 5: The above panel shows the results from the paper, the bottom panel shows my vs code with the reproduction of the results

In conclusion, we successfully replicated the experimental setup and achieved comparable results to those reported by the authors.

7 Critical review of the paper

7.1 Methodology Pros and cons

7.1.1 strengths:

1. Scalability: the approach effectively scales up the dataset using large-scale unlabeled images, improving generalization.
2. Robustness: leveraging pseudo-labeling and feature alignment enhances model robustness and depth estimation accuracy.
3. Efficiency: the method simplifies the data collection and annotation process, making it more efficient and cost-effective.

7.1.2 weaknesses:

1. Complexity: the training pipeline is complex, involving multiple stages and careful tuning of feature alignment.
2. Resource Intensive: handling and processing a large-scale dataset requires significant computational resources.

7.2 Structure and Writing

The structure follows a logical progression, starting with an introduction that sets the context and significance of the study, followed by a detailed review of related work. The methodology section is comprehensive, explaining the novel approach of leveraging large-scale unlabeled data through a teacher-student model and feature alignment loss. For instance, the paper describes how the teacher model is trained on 1.5 million labeled images and then used to generate pseudo labels for 62 million unlabeled images, which are subsequently used to train the student model. This approach is illustrated in Figure 2 of the paper, which shows the flow of labeled and unlabeled images through the training pipeline. The experiments section is robust, providing extensive quantitative and qualitative results, supported by well-organized tables and figures that enhance the reader's understanding. For example, Table 2 compares the zero-shot relative depth estimation performance of the proposed model with MiDaS v3.1 across multiple datasets, demonstrating significant improvements in metrics such as AbsRel and δ_1. Ablation Studies section provides insights into the contributions of different components of the model, which is crucial for understanding the effectiveness of the proposed approach.
The writing of the paper is clear and concise, making complex technical content accessible to readers. Technical terms are well-defined, and the explanations are detailed, ensuring that the reader can follow the methodology and results without confusion. The flow of ideas is logical, with smooth transitions between sections. The use of figures and tables is appropriate and enhances the readability of the paper, providing visual support to the textual content. The formal style is suitable for an academic audience. Overall, the paper is easy to understand, and the ideas are presented in a coherent and logical manner.

7.3 Limitations of the Paper

after reading the paper, three points are troubling us:

7.3.1 Model Size

The largest model trained in this project was ViT-Large. It would be beneficial to explore the performance of a more powerful teacher model, such as ViT-Giant, to potentially achieve better results.

7.3.2 Image Resolution

All models were trained on 512x512 images. This resolution is relatively low and may not provide practical depth maps for real-world applications. Future work should consider retraining on higher

resolutions, such as 700x700 or 1000^{+}x1000^{+}, to improve the quality and applicability of the depth maps.

7.3.3 Camera Parameters

The datasets used in this project have varying camera parameters, which means the scale of the depth is different across datasets. This variation poses a challenge in ensuring that the normalized depth maps (scaled from 0.0 to 1.0) generalize well to new cameras. A clear formula or method for converting these normalized values to real-world 3D coordinates is necessary to address this issue.

8 Our research work

8.1 Main idea

The primary objective of this research is to determine whether incorporating depth information can enhance the performance of image classification tasks. This study leverages the "Depth Everything" model to augment image data, aiming to improve the classification accuracy of a convolutional neural network (CNN). The dataset used for this research is "Animals in the Wild," which presents a challenging scenario due to the complex backgrounds and textures that often obscure the animals.

8.2 Dataset

The "Animals in the Wild" is a balanced dataset consists of 5400 images spanning 90 different classes of animals. The images are captured in natural settings, where the animals are sometimes camouflaged by their surroundings. This complexity makes it difficult for standard CNNs to classify the images accurately. Another issue arises from the fact that some animals come in various colors, which can cause a model to incorrectly use color as a distinguishing feature, leading to prediction errors.

Figure 6: sample from animals in the wild dataset some colors can hide the object or confuse the model

8.3 Methodology

8.3.1 Data Augmentation with Depth Information

Using the code provided by the authors of the "Depth Everything" model, we modified it to generate depth maps for the "Animals in the Wild" dataset. This new dataset, referred to as "Depth-Animals," retains the original structure and train-val split but includes depth maps of the original images.

8.3.2 Feature Extraction

A pre-trained ResNet50 model was employed as a feature extractor. The fully connected head of the ResNet50 was removed, resulting in a feature vector of size 1000 for each image. This process was applied to both the original and Depth-Animal datasets. These vectors capture essential characteristics of the images, which are gathered from convolutions with learned kernels that can be considered as sophisticated "edge" or "shape" detectors as we saw in the course. we believe that using those convolutions on the depth map can lead to stronger distinct-able shape characteristics.

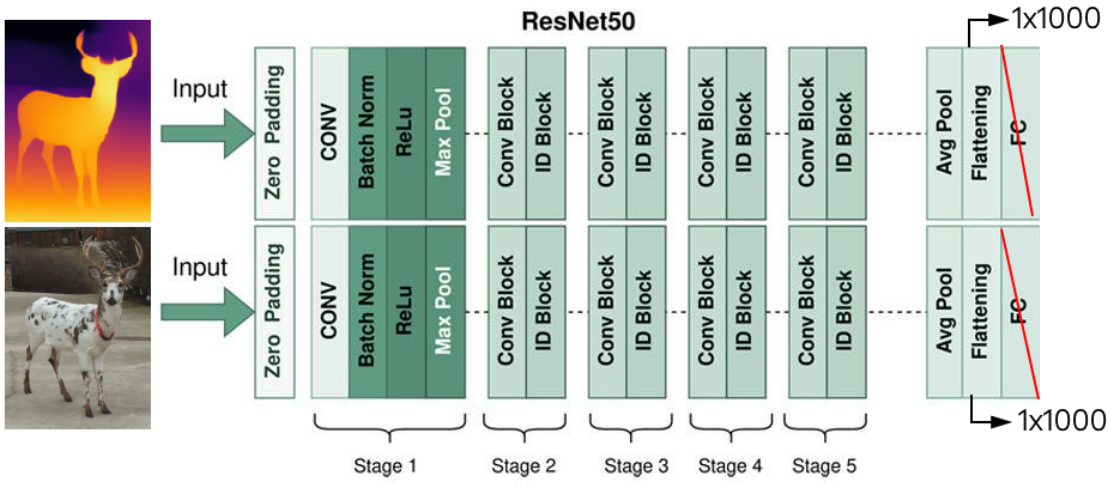

Figure 7: A Headless Resnet 50 is used as feature extractor in both animals and depth-animals dataset

8.3.3 Multi-Layer Perceptron as a vector classifier

We build a simple MLP that consists of 3 layers and 2 relu functions. In our experiments, the MLP was used to classify the feature vectors extracted from the RGB images and the Depth maps.

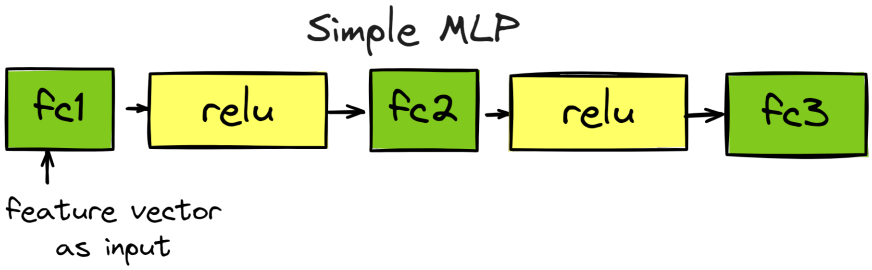

Figure 8: A simple MLP were trained on the vector classification method

8.3.4 Experimental Setup

Three experiments were conducted to evaluate the impact of depth information on image classification:

1. **Experiment 1: Depth-Only Classification**
   - **Training:** Depth-Animals Train-set feature vectors (of size: 1000x1)
   - **Testing:** Animals validation set
2. **Experiment 2: Standard Classification**
   - **Training:** Animals (RGB) Train-set feature vectors (of size: 1000x1)
   - **Testing:** Animals validation set
3. **Experiment 3: Combined Feature Classification**
   - **Training:** Concatenated feature vectors (RGB + depth) Train-set (of size: 2000x1)
   - **Testing:** Animals validation set

All experiments were trained for 30 epochs using the ADAM optimizer, on Nvidia A6000. It's also worth noting that we used the dropout technique during training to ensure that all neurons were engaged, preventing reliance solely on RGB features (see experiment 3).

8.4 Results and Discussion

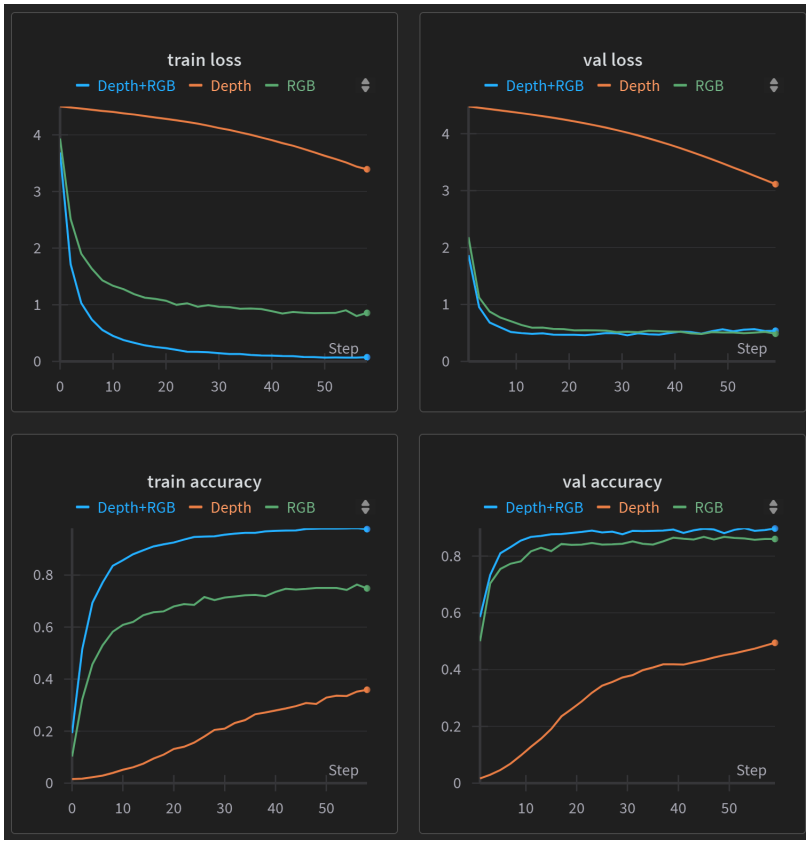

Figure 9: The training graphs

The results of the experiments are summarized as follows:

Experiment Name	Train Loss	Val Loss	Train Accuracy	Val Accuracy
Depth	3.392	3.114	0.359	0.494
RGB	0.857	0.486	0.779	0.860
Depth+RGB	0.074	0.539	**0.937**	**0.897**

Table 1: Results of the Experiments

- **Experiment 1:** Depth-Only Classification: Achieved 49.5% accuracy on the validation set using only depth information. This is a significant improvement over random guessing (11%) given the 90 classes, indicating that depth information alone can provide valuable cues for classification. However, the high train and validation losses (3.392 and 3.114, respectively) and low accuracies (train: 0.359, val: 0.494) suggest that depth information alone is insufficient for high accuracy.
- **Experiment 2:** Standard RGB Classification: The standard classification approach using the original images achieved 86% accuracy on the validation set, demonstrating the effectiveness of the ResNet50 feature extractor. The train and validation losses (0.857 and 0.486, respectively) are much lower than in Experiment 1, indicating better model performance. The train and validation accuracies (0.779 and 0.860, respectively) show that the model generalizes well to unseen data.
- **Experiment 3:** Combined Feature Classification: The combined feature approach, which concatenated the original and depth feature vectors, achieved the highest accuracy of 89.7%. This suggests that depth information, when used in conjunction with the original image features, can

enhance the model's ability to distinguish between classes. The train and validation losses (0.074 and 0.539, respectively) are the lowest among the three experiments, indicating a well-fitted model. The train and validation accuracies (0.937 and 0.897, respectively) are also the highest, showing that the combined features provide the most robust representation for classification and we believe that adding the depth map helped the model to ignore strong decision-making based on colors.

Evaluation Metrics: The performance of the classification models is evaluated using accuracy:

$$\text{Accuracy}(Set, Classifier) = \frac{\sum_{i=1}^{n} \mathbf{1}(y_i = \hat{y}_i)}{n} \tag{2}$$

where:

- n is the total number of samples in the Set.
- y_i is the true label of the i-th sample.
- $\hat{y}_i$ is the predicted label of the i-th sample.
- $\mathbf{1}$ is the indicator function, which returns 1 if the condition inside is true and 0 otherwise.

this metric is good for balanced datasets and measures the proportion of correctly classified images.

8.5 Conclusion

The experiments demonstrate that depth information can indeed improve image classification performance. The combined feature approach yielded the best results, indicating that depth maps provide complementary information that enhances the discriminative power of the model.

8.6 Future Work

Future research could explore the use of depth model inference as a layer in a CNN architecture during the training and inference process. Additionally, experimenting with different architectures and optimization strategies could further improve the classification performance.

9 works affected by this paper

This paper is relatively new (CVPR 2024) and has not yet had a significant impact. However, this month, the authors presented Depth Anything V2, which improves upon V1 by utilizing synthetic data instead of labeled real images, scaling up the teacher model, and employing large-scale pseudo-labeled real images for training student models.

References

[1] Huang, X., et al. (2023). Recognize Anything: A Strong Image Tagging Model. *arXiv preprint arXiv:2306.03514*.

[2] Liu, S., et al. (2023). Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. *arXiv preprint arXiv:2303.05499*.

[3] Ke, L., et al. (2023). Segment Anything in High Quality. *arXiv preprint arXiv:2306.01567*.

[4] Darcet, T., et al. (2024). DINOv2: Learning Robust Visual Features without Supervision. *arXiv preprint arXiv:2304.07193*.

[5] Srinivasan, P. P., Garg, R., Wadhwa, N., Ng, R., Barron, J. T. (2018). Aperture Supervision for Monocular Depth Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] He, K., Zhang, X., Ren, S., Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385*.