DepthDive: Enhanced Underwater Depth Estimation using Monocular Images

Abhimanyu Bhowmik
SeaTech - Ecole d'ingénieurs
Université de Toulon
La Garde, France
abhimanyu-bhowmik@etud.univ-tln.fr

Madhushree Sannigrahi
SeaTech - Ecole d'ingénieurs
Université de Toulon
La Garde, France
madhushree-sannigrahi@etud.univ-tln.fr

Krittapat Onthuam
SeaTech - Ecole d'ingénieurs
Université de Toulon
La Garde, France
krittapat-onthuam@etud.univ-tln.fr

***Abstract*—Accurate underwater depth estimation is vital for applications such as autonomous underwater vehicles, marine biology, and underwater archaeology. Traditional methods often rely on expensive and complex equipment, whereas monocular depth estimation offers a more cost-effective alternative. Despite significant advancements in terrestrial monocular depth estimation driven by deep learning, these models are inefficient in underwater environments due to challenges such as light attenuation, water turbidity, and data scarcity. This paper introduces DepthDive, a novel approach that adapts the Depth Anything Model (DAM) for underwater depth estimation using monocular images. The model is fine tuned via the parameter effcient fine tuning (PEFT), specifically low rank adaptation (LoRA). In addition, this work proposed a data sample filtering method to improve the quality of underwater depth dataset. Experimental results demonstrate that DepthDive significantly improves depth estimation accuracy in underwater environments, even with limited datasets, showcasing the potential of fine-tuning foundation models for specialized applications. All the code and supplementary documents can be found in the GitHub repository[1].**

***Index Terms*—underwater depth estimation, fine-tuning**

I. Introduction

Making up more than two-thirds of Earth's surface, the ocean is essential to human exploration and study. For many applications, such as autonomous underwater vehicles (AUVs), underwater robotics, marine biology, ecology, and archaeology, precise underwater depth estimation is crucial. Compared to expensive and operationally complex pieces of equipment, monocular depth estimation provides a more cost-effective and convenient alternative for measuring underwater depth.

Monocular depth estimation has made significant progress for terrestrial images, primarily driven by advancements in deep learning. However, its application to underwater settings remains inadequate, mainly due to the scarcity of relevant data. The challenges posed by light attenuation, water turbidity, and backscattering make it difficult to acquire clear underwater images and accurate depth information. Notably, the zero-shot performance of underwater depth estimations lags significantly when tested on foundation models trained for terrestrial environments, such as Depth Anything [2], Marigold [3], and UniMatch [4]. Foundation models are AI networks trained on vast amounts of diverse, typically unlabeled data, making them suitable for various downstream tasks. However, the fragmented nature and a lack of properly annotated underwater depth maps hinder the training of a vision transformer model entirely from scratch, which requires substantial computing resources and datasets (about 65 million in the case of Depth Anything).

Given the challenges and limitations of current models and data availability, there is a need for effective methods to adapt existing large-scale transformer models for underwater depth estimation. This adaptation should minimize the need for extensive new datasets and computational resources while improving the performance of these models in underwater environments.

A. Key Contributions

In response to these challenges, we developed DepthDive, a method to customize the Depth Anything Model (DAM) for underwater monocular depth estimation. Our contribution includes:

1) **Dataset Aggregation:** We conducted an extensive survey to compile multiple small underwater datasets with reliable depth annotations.
2) **Data Quality Enhancement:** We proposed a universal filter to eliminate poorly annotated data samples, preventing the model from learning inaccurate information.
3) **Model Customization:** We utilized a low-rank adaptation strategy (LoRA) [5] to fine-tune the DAM. By freezing the trained weights and updating only a small fraction of the parameters (approximately 2% of the original model size), we significantly improved the model's performance with fewer datasets in underwater scenarios.
4) **Analysis with synthetic data:** We utilize different percentages of synthetic and real data to understand at least how much real dataset we need to achieve similar performance as the model trained on a purely real dataset.

Our experimental results demonstrate that finetuning a foundation model is a viable approach to achieving higher accuracy in underwater depth estimation, even with limited datasets.

[1]https://github.com/abhimanyubhowmik/Underwater_Depth_Estimation

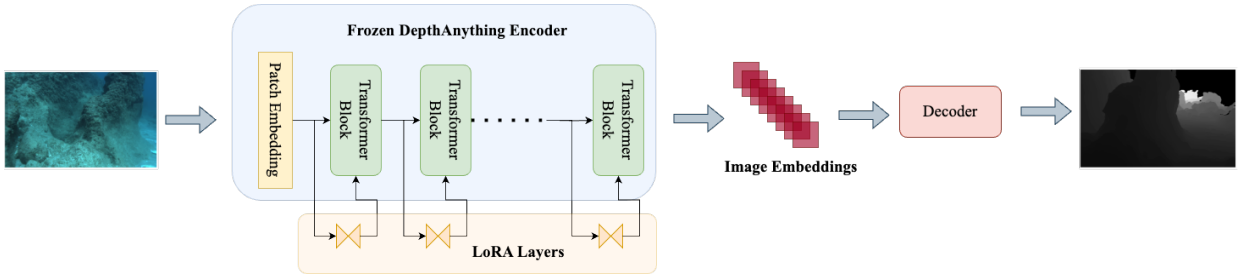

Fig. 1. DepthDive Approach (diagram inspired from [1])

II. Related Works

Early efforts in underwater depth estimation from monocular images focused on improving the quality of underwater images [6]. These methods aimed to address issues such as poor lighting and attenuation effects, transforming underwater images to more closely resemble those taken in terrestrial environments [7]. Additionally, some research has utilized stereo-pair images for underwater depth estimation using geometric models [8]. However, these models struggle to accurately compute the depth of objects that are far and blurred due to underwater lighting challenges.

Depth estimation from terrestrial images using supervised DL-based methods has achieved good success, leveraging extensive training datasets with ground truth depth information [3]. However, this approach is not feasible for underwater environments due to the absence of a large-scale underwater depth dataset that could support supervised training. To overcome this limitation, Gupta and Mitra [9] proposed an unsupervised network for single-image underwater depth estimation. Their method involves learning a mapping between unpaired RGBD hazy terrestrial images and arbitrary underwater images, though the depth information from terrestrial images may not directly correspond to underwater image characteristics. Similarly, Hambarde et al. [10] introduced a GAN-based network using synthetic underwater images generated from a synthetic dataset, but these synthetic images failed to fully capture the complexities of real underwater environments. Recent advancements include self-supervised learning techniques [8] [11] [12], which establish pixel correspondences based on predicted depth maps and minimize photometric reconstruction loss of paired pixels.

Supervised Learning was also attempted to devise lightweight models by either using models like MobileNet [1] or by fine-tuning or merging existing simpler models [13]. These methods worked great for real-time underwater depth estimations but in general, their accuracy and robustness to change are highly uncertain. Despite the progress in underwater depth estimation, existing methods often fall short in addressing the unique challenges posed by underwater environments. These challenges include data scarcity, poor image quality, and the complexity of accurately mapping underwater scenes. Our approach, DepthDive, addresses these gaps by customizing the existing Depth Anything Model [2] for underwater depth estimation, utilizing a low-rank adaptation strategy [5] to fine-tune the existing foundation model with minimal data. Recently, fine-tuning approaches have seen greater success and in our case, rival specialist models on benchmarks. This method not only improves accuracy but also reduces the computational resources required, making it a practical solution for real-world underwater applications.

III. Proposed Methodology

A. Foundation Model Comparison

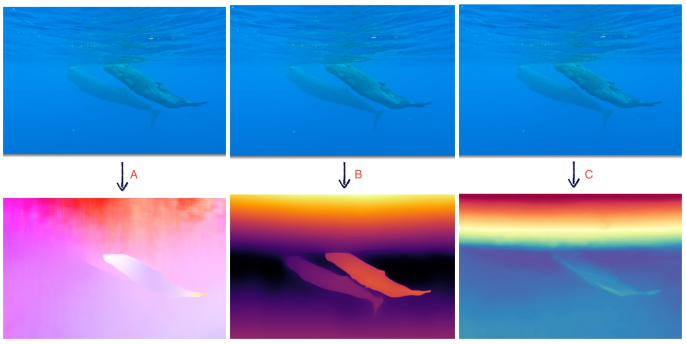

Fig. 2. Comparing different baseline models (A) UniMatch, (B) DepthAnything Model (C) Marigold

In our study, we compared three recent state-of-the-art baseline models trained on terrestrial depth estimation: UniMatch [4], Marigold, and Depth Anything [2] as in Figure 2. UniMatch [3] employs a hybrid approach combining CNN and Transformer architectures with cross-attention mechanisms. It processes stereo pair images to generate disparity and depth maps. Marigold, in contrast, is a diffusion model fine-tuned specifically for monocular depth estimation. Similarly, Depth Anything is a transformer-based model, designed for depth estimation in different scenarios.

All these models represent a significant advancement in depth estimation, leveraging large-scale transformer models. However, when evaluated in zero-shot on underwater images with favourable lighting conditions, all models exhibited a significant performance drop. UniMatch performed the worst, failing to detect any meaningful depth information in the images. Marigold and Depth Anything demonstrated comparable performance, but quantitative analysis revealed that Depth

TABLE I
ANALYSING DIFFERENT DATASETS

Name	Camera	Size	Image Type	Depth type	Lighting	Depth (m)	Estimation method
SQUID [14]	Stereo	57(Video)	Natural	Real (Metric)	Clear	3-30	AprilTags with size reference
Eiffel Tower [15]	Mono	18082	Natural	Real (Relative)	Dark	1700	Structure-From-Motion (SFM)
NAREON [16]	Mono	7000	Natural	Real (Relative)	Varying	0.01 - 2.5	Hybrid imaging system
FLSea VI [17]	Mono	22451	Natural	Real (Metric)	Varying	0-12	AprilTags with size reference
SeaThru [6]	Mono	1157	Natural	Real (Metric)	Clear	4-10	Structure-From-Motion (SFM)
VAROS [18]	Mono	4713	Synthetic (Blender)	Real (Metric)	Dark	-	Information from blender
ATLANTIS [1]	Mono	3200	Synthetic (Generated)	Real (Relative)	Varying	-	Using MiDas
DRUVA [12]	Mono	20 (30 fps)	Natural	Generated (Relative)	Clear	3-6	Using USe-ReDI-Net
USOD 10k [19]	Mono	10255	Natural	Generated (Relative)	Varying	5-60	Using DPT

Anything outperformed Marigold in terms of accuracy and reliability.

B. Model Overview: Depth Anything Architecture

Utilising the same approach as MiDas, DepthAnything is a state-of-the-art monocular depth estimation model generally developed for general scene depth estimation. The model utilises both labelled and unlabelled datasets by adapting the teacher-student method. The teacher model learns the labelled dataset and predicts the pseudo-label of the unlabeled dataset. The student model is then able to learn from both datasets. The model excels in zero-shot depth estimation and is a potential baseline for underwater depth estimation.

C. PEFT (Parameter Efficient Fine-Tuning)

In order to reduce computational demands, we utilised PEFT. It reduces the amount of fine-tuning parameters without compromising the model performance, resulting in a more efficient model training pipeline.

1) LoRA (Low-Rank Adaptation): Most PEFT methods are designed to optimize large language models. LoRA, however, is also proven to be successful in optimizing image-based models. LoRA allows us to train some dense layers in a neural network indirectly by optimizing rank decomposition matrices of the dense layers' change during adaptation instead of keeping the pre-trained weights frozen. With LoRA, we have fine-tuned 2% of the total parameters and preserved the model performance, significantly reducing the resources and time required for the experiments.

D. Evaluation Metrics

1) Absolute Relative Error (AbsRel): The Absolute Relative Error measures the difference between the estimated depth $\hat{d}$ and the ground truth depth d. It is defined as:

$$\text{AbsRel} = \frac{1}{n}\sum_{i=1}^{n}\frac{|d_i - \hat{d}_i|}{d_i}$$

This metric highlights the proportional difference between the predicted and actual depths.

2) Squared Relative Error (SqRel): The Squared Relative Error focuses on the square of the difference between the estimated and ground truth depths, normalized by the ground truth depth. It is given by:

$$\text{SqRel} = \frac{1}{n}\sum_{i=1}^{n}\frac{(d_i - \hat{d}_i)^2}{d_i}$$

This metric penalizes larger errors more heavily.

3) Root Mean Squared Error (RMSE): The Root Mean Squared Error measures the square root of the average of the squared differences between estimated and ground truth depths. It is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - \hat{d}_i)^2}$$

This metric provides a measure of the overall error magnitude.

4) Logarithmic RMSE: The Logarithmic RMSE measures the error in the log space, which can help in evaluating depth estimations with a wide range of values. It is given by:

$$\text{RMSElog} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log d_i - \log \hat{d}_i)^2}$$

This metric reduces the impact of large absolute differences.

5) Scale Invariant MSE in Log Scale: This metric evaluates the mean squared error in log space while being invariant to the scale of the depth values. It is defined as:

$$\text{SiLog} = \sqrt{\mathbb{E}\left[(\log(\hat{d}) - \log(d))^2\right] - \left(\mathbb{E}\left[\log(\hat{d}) - \log(d)\right]\right)^2}$$

$E[\cdot]$ denotes the expected value (mean) over the dataset.

6) Peak Signal-to-Noise Ratio (PSNR): PSNR measures the ratio between the maximum possible value of a signal and the power of distorting noise. It is given by:

$$\text{PSNR} = 10\log_{10}\left(\frac{\text{MAX}_d^2}{\sqrt{\text{MSE}}}\right)$$

where MAX_d is the maximum possible depth value.

7) Structural Similarity Index (SSIM): SSIM evaluates the perceptual quality of the depth estimation by comparing structural information. It is defined as:

$$\text{SSIM}(d, \hat{d}) = \frac{(2\mu_d\mu_{\hat{d}} + C_1)(2\sigma_{d\hat{d}} + C_2)}{(\mu_d^2 + \mu_{\hat{d}}^2 + C_1)(\sigma_d^2 + \sigma_{\hat{d}}^2 + C_2)}$$

where μ and σ represent the mean and variance, and C_1 and C_2 are constants to stabilize the division.

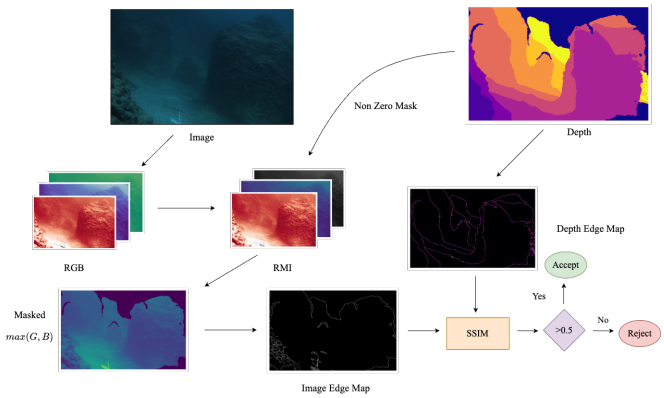

Fig. 3. Data Sample Filtering procedure

8) Pearson Correlation: Pearson Correlation measures the linear correlation between the estimated depth and the ground truth depth. It is defined as:

$$r = \frac{\sum_{i=1}^{n}(d_i - \bar{d})(\hat{d}_i - \bar{\hat{d}})}{\sqrt{\sum_{i=1}^{n}(d_i - \bar{d})^2 \sum_{i=1}^{n}(\hat{d}_i - \bar{\hat{d}})^2}}$$

where $\bar{d}$ and $\bar{\hat{d}}$ are the means of d and $\hat{d}$, respectively.

9) δ_i: The δ_i metric measures the percentage of pixels for which the estimated depth is within a certain factor of the ground truth depth. For δ_1, it is defined as:

$$\delta_i = \text{percentage of } \left(\max\left(\frac{d}{\hat{d}}, \frac{\hat{d}}{d} \right) < 1.25^i \right)$$

This metric provides insight into the accuracy of depth estimation relative to the true depth.

IV. Experiments and Results

A. Datasets

The main hurdle in underwater depth estimation is the lack of comprehensive datasets. Unlike terrestrial depth estimation, there is no benchmark dataset for underwater environments. Available underwater datasets are quite small compared to terrestrial ones and vary greatly in terms of depth, processing methods, and camera setups. Additionally, the depth maps in these datasets are generated using different methods and can represent either relative or metric depth.

To address this issue, we analyzed almost all available underwater datasets to combine them for training our model. However, as shown in Table I, many of these datasets were of poor quality. Benchmark datasets in the literature, such as SQUID [14] and SeaThru [6], have unreliable depth maps with missing objects. These maps are typically generated using the Structure-from-Motion (SFM) technique, which often blurs distant objects and fails to capture the depth of moving objects. The most accurate ground truths are found in synthetically generated datasets like VAROS [18] and ATLANTIS [1]. However, these synthetic datasets do not fully mimic real-world conditions, as they lack the presence of moving objects and the varying light and turbidity conditions found in actual underwater environments.

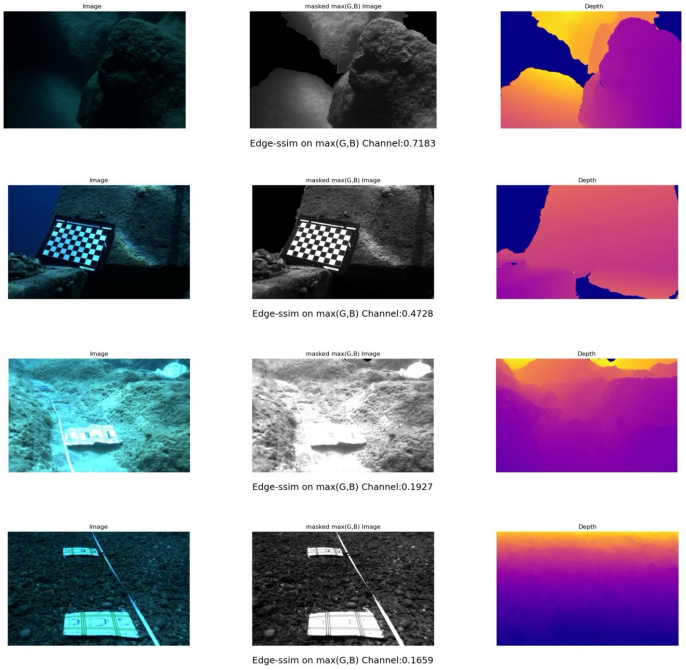

Fig. 4. Results after using the filter method

To overcome all these issues, we trained our model with the FLSeaVI [17] (real) and VAROS (synthetic) datasets. We experimented with different percentages of synthetic and original datasets to train the model in order to minimise the dependence on real-world datasets while still maintaining the model's performance in the real world. After various experiments as shown in Figure 5. The trained model is tested and compared with other state-of-the-art models using the D3 and D5 subsets of the SeaThru dataset and the FLSeaVI dataset.

B. Data Sample Filtering

For real-world datasets, we often get inaccurate ground truth values. If we finetune our model on those data points, the model might learn biased distributions of depth maps. To avoid this, we developed a method, which can eliminate inaccurate ground truths, providing a better dataset for model fine-tuning. Our method includes converting RGB images to RMI input space [8], which takes into account underwater light characteristics of propagation. The red wavelength suffers more aggressive attenuation underwater, so the relative differences between R channel and G, B channel values can provide useful depth information for a given pixel. We take the maximum value of B and G channels and mask the pixels, which have zero depth values in the ground truth. The resultant images are given in figure 4.

Then, we performed edge-based SSIM (ESSIM) [22]as shown in figure 3. We first perform canny-edge detection on the masked image and depth map. Then we calculate SSIM

TABLE II
PERFORMANCE COMPARISON OF DEPTH ANYTHING MODEL WITH VAROS DATASET

	AbsRel ↓	SqRel ↓	RMSE ↓	RMSElog ↓	SIlog ↓	log10 ↓	PSNR↑	SSIM↑	Person corr↑	δ_1 ↑	δ_2 ↑	δ_3 ↑
Without Training	5.9228	52.6330	5.8734	1.8618	0.6001	0.7679	8.4031	0.7501	0.7093	0.0189	0.0404	0.0669
5 Epochs Training	0.2336	0.2696	0.1581	0.2753	0.2195	0.0780	18.8912	0.9367	0.7885	0.7878	0.9220	0.9567

TABLE III
PERFORMANCE COMPARISON OF DEPTH ANYTHING MODEL WITH FLSEAVI DATASET

	AbsRecl	SqRel ↓	RMSE ↓	RMSElog ↓	SIlog ↓	log10 ↓	PSNR↑	SSIM↑	Person corr↑	δ_1 ↑	δ_2 ↑	δ_3 ↑
Without Training	3.8750	36.1175	7.5801	1.4900	0.9353	0.5726	11.7313	0.7005	-0.8213	0.0796	0.1608	0.2440
5 Epochs Training	0.0762	0.4483	0.7114	0.3690	0.3629	0.0404	24.3683	0.9488	0.8658	0.9633	0.9753	0.9794

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT MODELS ON FLSEAVI AND SEATHRU DATASETS

Dataset	Model	AbsRel ↓	SqRel ↓	RMSE ↓	RMSElog ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
FLSeaVI	UW-Net [9]	0.527	1.765	1.725	1.961	0.337	0.565	0.699
	Amitai et al [11]	0.203	1.955	1.546	**0.245**	0.768	0.923	0.966
	Ours	**0.0762**	**0.4483**	**0.7114**	0.3690	**0.9633**	**0.9753**	**0.9794**
SeaThru (D3 and D5)	IDisc-KITTI [20]	4.702	4.4288	5.891	1.192	0.093	0.241	0.359
	IDisc-Atlantis [1]	1.630	1.4279	**1.371**	**0.354**	**0.553**	**0.850**	**0.955**
	NewCRFs-KITTI [21]	2.874	1.5768	3.251	0.934	0.213	0.375	0.465
	NewCRFs-Atlantis [1]	1.683	1.4764	1.435	0.378	0.476	0.837	0.952
	Ours	**0.7925**	**0.9480**	1.6575	0.8268	0.1797	0.4052	0.6128

between the pair of edgemaps. We tried multiple thresholds and finally settled on $\alpha = 0.5$. If an image pair has SSIM of less than $alpha$, we discard them from the training set and label them as poor data points.

C. Loss Function

For the Loss function, we tested the model initially with Mean Absolute Error ($L1\_Loss$). We later incorporated Scale-Invariant Mean Squared Error($SiLog$), which penalizes errors at close range and becomes more forgiving at greater distances. We experimented with the weighted sum of $L1\_Loss$ and $SiLog$, which allowed us to balance the learning focus for estimating accurate relative depth.

$$\text{Loss} = \lambda_1 \cdot \mathcal{L}_{SiLog} + \lambda_2 \cdot \mathcal{L}_{L1\_Loss}$$

Where: - λ_1 and λ_2 are the weights for the SiLog and L1 loss components respectively. We experimented with different values of λ_1 and λ_2 as seen in Figure 6. We got similar results with all the combinations and decided to go with $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$.

The SI-Log and L1-loss components are defined as:

$$\mathcal{L}_{SILog}(\hat{d}, d) =$$

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(\hat{d}) - \log(d))^2 - \frac{\lambda}{N^2}(\sum_{i=1}^{N}\log(\hat{d}) - \log(d))^2}$$

$$\mathcal{L}_{L1\_Loss} = \frac{1}{N}\sum_{i=1}^{N}|\hat{d}_i - d_i|$$

In these equations:
- $\hat{d}$ represents the predictions,
- d represents the targets, over the dataset.

For the background error, the training almost mitigates all the issues related to backgrounds. Thus, there is not enough motivation to incorporate this error in the loss function which might increase computation load while training.

We also experimented with Chamfer Distance [23] as the loss function. It provided us similar results as now but we dropped it since it was computationally very expensive for a minimal improvement.

D. Implementation Details

We used a cosine annealing learning rate scheduler with different warmup periods (10, 20, 30, 40, and 50) and adjusted the learning rate from 10^{-5} to 10^{-2}. We used the AdamW optimizer and tested the model for 5, 10, and 15 epochs with the combined dataset, and for 2, 4, and 8 epochs with other datasets. For the LoRA configuration, we set alpha to 32 and rank to 16, which allowed us to fine-tune the model by training only 1.18% of the total parameters (294,912 out of 25,080,001). The best results were achieved with a warmup period of 40, a learning rate of 10^{-3}, and 5 epochs.

E. Discussion

In order to assess the performance of the model training, we visualized the 3d point-cloud in figure 7. The image shows the before and after training distribution of depth map in 3D. The 3D map is being generated using camera parameters from the FlSea VI dataset given in the calibration folder. The intrinsic camera parameters, consist of the calibration matrix

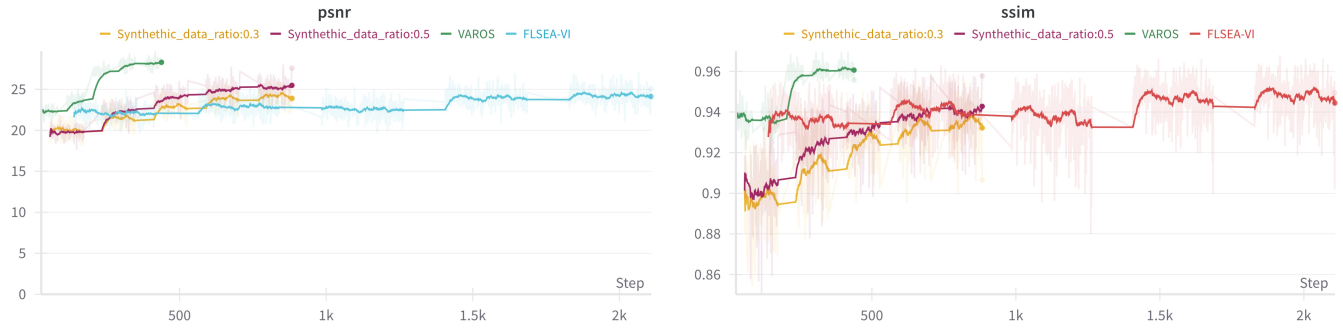

Fig. 5. PSNR and SSIM plot for various Synthetic Data Ratio

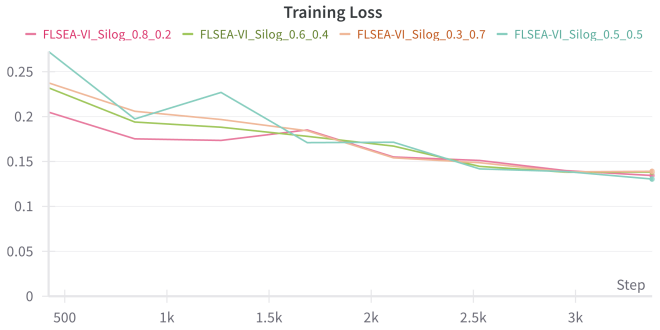

Fig. 6. Training Loss for various λ_1 and λ_2

$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$, are essential for this transformation. For each pixel (u, v) in the depth map with depth d, the corresponding 3D point (X, Y, Z) can be computed using the equations:

$$X = \frac{(u - c_x) \cdot d}{f_x}, \quad Y = \frac{(v - c_y) \cdot d}{f_y}, \quad Z = d$$

This transformation translates pixel coordinates and depth values into real-world coordinates. As shown in Figure 7, the impact of training on the model's depth distribution representation is evident. The model learns the range of depth distribution according to the ground truth and effectively reduces the noise and deviation from the actual data points.

To test the hypothesis of incorporating synthetic datasets, we experimented with two configurations: one with 70% real and 30% synthetic data, and another with an equal split of 50% real and 50% synthetic data. FL-SEA VI and VAROS serve as benchmarks for pure real and synthetic datasets, respectively. The length of the curves in the graphs varies because changing the dataset ratio alters the number of data points accordingly. For comparison, we have focused on PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) to provide a clear understanding of performance metrics, though similar trends are observed across other metrics as well. According to figure 5 we can clearly understand that even though having a mix of synthetic data reduces the matrices initially, it's clearly evident that the training model can learn the distribution and generalise fairly quickly.

V. Conclusion

In this work, we introduce DepthDive, an innovative underwater depth estimation model derived from DepthAnything-small. Our research involved experimenting to obtain the most efficient baseline general image depth estimation model, which is subsequently adapted for underwater environments. The model was trained on various underwater depth image datasets, encompassing both original and synthetic images. Fine-tuning was performed using LoRA, which optimizes computational efficiency while preserving model performance. The model showed improvement on both real(FlseaVI) and synthethic(VAROS) dataset. Additionally, we proposed a pre-processing technique to improve the quality of the annotated dataset. We investigated the integration of real and synthetic datasets, which holds the potential to enhance model efficacy. Our model shows promising result compared to other state-of-the-art depth estimation models.

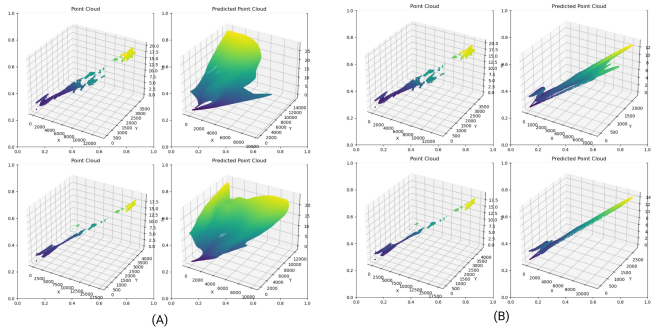

Fig. 7. (A) Image samples before training, (B) Image samples after training

VI. Future Works

1) More experiments could be done on optimizing the ratio between original and synthetic images in the training dataset.

2) More experiments could be done on the different variations of the Depth Anything model.
3) We also need to incorporate more real-world datasets to increase the variations in the samples.
4) We need to experiment with different hyperparameters of LoRA. Other PEFT optimisation methods, such as ConvLoRA [24], could also be tested.

REFERENCES

[1] F. Zhang, S. You, Y. Li, and Y. Fu, "Atlantis: Enabling underwater depth estimation with stable diffusion," *arXiv preprint arXiv:2312.12471*, 2023.

[2] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," *arXiv preprint arXiv:2401.10891*, 2024.

[3] B. Ke, A. Obukhov, S. Huang, N. Metzger, R. C. Daudt, and K. Schindler, "Repurposing diffusion-based image generators for monocular depth estimation," *arXiv preprint arXiv:2312.02145*, 2023.

[4] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, F. Yu, D. Tao, and A. Geiger, "Unifying flow, stereo and depth estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[5] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[6] D. Akkaynak and T. Treibitz, "Sea-thru: A method for removing water from underwater images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1682–1691.

[7] C. Boittiaux, R. Marxer, C. Dune, A. Arnaubec, M. Ferrera, and V. Hugel, "Sucre: Leveraging scene structure for underwater color restoration," *arXiv preprint arXiv:2212.09129*, 2022.

[8] B. Yu, J. Wu, and M. J. Islam, "Udepth: Fast monocular depth estimation for visually-guided underwater robots," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3116–3123.

[9] H. Gupta and K. Mitra, "Unsupervised single image underwater depth estimation," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 624–628.

[10] P. Hambarde, S. Murala, and A. Dhall, "Uw-gan: Single-image depth estimation and image enhancement for underwater images," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[11] S. Amitai, I. Klein, and T. Treibitz, "Self-supervised monocular depth underwater," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1098–1104.

[12] N. Varghese, A. Kumar, and A. Rajagopalan, "Self-supervised monocular underwater depth recovery, image restoration, and a real-sea video dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 12 248–12 258.

[13] L. Ebner, G. Billings, and S. Williams, "Metrically scaled monocular depth estimation through sparse priors for underwater robots," *arXiv preprint arXiv:2310.16750*, 2023.

[14] D. Berman, D. Levy, S. Avidan, and T. Treibitz, "Underwater single image color restoration using haze-lines and a new quantitative dataset," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 8, pp. 2822–2837, 2020.

[15] C. Boittiaux, C. Dune, M. Ferrera, A. Arnaubec, R. Marxer, M. Matabos, L. Van Audenhaege, and V. Hugel, "Eiffel tower: A deep-sea underwater dataset for long-term visual localization," *The International Journal of Robotics Research*, vol. 42, no. 9, pp. 689–699, 2023.

[16] J. M. M. Dionísio, P. N. A. A. S. Pereira, P. N. Leite, F. S. Neves, J. M. R. S. Tavares, and A. M. Pinto, "Nereon - an underwater dataset for monocular depth estimation," in *OCEANS 2023 - Limerick*, 2023, pp. 1–7.

[17] Y. Randall, "Flsea: Underwater visual-inertial and stereo-vision forward-looking datasets," Ph.D. dissertation, University of Haifa (Israel), 2023.

[18] P. G. O. Zwilgmeyer, M. Yip, A. L. Teigen, R. Mester, and A. Stahl, "The varos synthetic underwater data set: Towards realistic multi-sensor underwater data with ground truth," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 3722–3730.

[19] L. Hong, X. Wang, G. Zhang, and M. Zhao, "Usod10k: a new benchmark dataset for underwater salient object detection," *IEEE transactions on image processing*, 2023.

[20] L. Piccinelli, C. Sakaridis, and F. Yu, "idisc: Internal discretization for monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 477–21 487.

[21] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, "Neural window fully-connected crfs for monocular depth estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3916–3925.

[22] G.-H. Chen, C.-L. Yang, L.-M. Po, and S.-L. Xie, "Edge-based structural similarity for image quality assessment," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 2, 2006, pp. II–II.

[23] M. A. Butt and P. Maragos, "Optimum design of chamfer distance transforms," *IEEE Transactions on Image Processing*, vol. 7, no. 10, pp. 1477–1484, 1998.

[24] Z. Zhong, Z. Tang, T. He, H. Fang, and C. Yuan, "Convolution meets lora: Parameter efficient finetuning for segment anything model," *arXiv preprint arXiv:2401.17868*, 2024.