Bridging the Sim-to-Real GAP for Underwater Image Segmentation

Luiza Ribeiro Marnet[†,‡], Stella Grasshof[‡], Yury Brodskiy[†], and Andrzej Wąsowski[‡]
[†]EIVA a/s, Denmark, [‡]Computer Science Department, IT University of Copenhagen, Denmark

***Abstract*— Labeling images for every new task or data pattern a model needs to learn is a significant time bottleneck in real-world applications. Moreover, acquiring the necessary data for training the models can be challenging. Ideally, one would train the models with simulated images and adapt them for the desired real tasks using the least possible amount of data. Active learning can be used to solve this problem with minimal effort. In this work, we train SegFormer for pipeline segmentation with synthetic images from an underwater simulated environment and fine-tune the model with real underwater pipeline images recorded in a marina. The evaluation shows that selecting real data with active learning for fine-tuning the model gives better results than randomly selecting the images. As part of the work, we release the dataset recorded in the marina, MarinaPipe, which will be publicly available.**

***Index Terms*— active learning, computer vision, sim-to-real, underwater image segmentation**

I. Introduction

Training deep learning models requires good-quality datasets. Acquiring and labeling data is costly and time-consuming, making synthetic data an attractive option. Although modern simulators are highly advanced and close to real scenarios, a gap still exists between their patterns.

One way to address this gap is to pre-train a model using synthetic data and then fine-tune it with real data. Yet, this means acquiring and annotating the dataset, which typically requires hours of labeling by specialists. Nevertheless, the datasets usually contain many repetitive patterns that pre-trained models already recognize. Active learning can be applied to select the minimal subset of samples that needs to be used for fine-tuning, based on the model's lack of knowledge [1].

In this paper, we study this sim-to-real gap and how to overcome it in the underwater vision domain, cf. Fig. 1. Real underwater images pose many challenges, including non-uniform illumination, low contrast, color degradation, and motion blur [2], [3], [4], [5]. These are properties that are challenging to mimic realistically in synthetic images, making this study relevant. To the best of our knowledge, this is the first paper studying the sim-to-real gap for RGB underwater images using active learning. Our contributions include:

- The use of active learning to fine-tune a model trained with synthetic data using underwater real data;
- An evaluation of the visual transformer SegFormer with the active learning technique;
- An underwater pipeline dataset, MarinaPipe, publicly released.

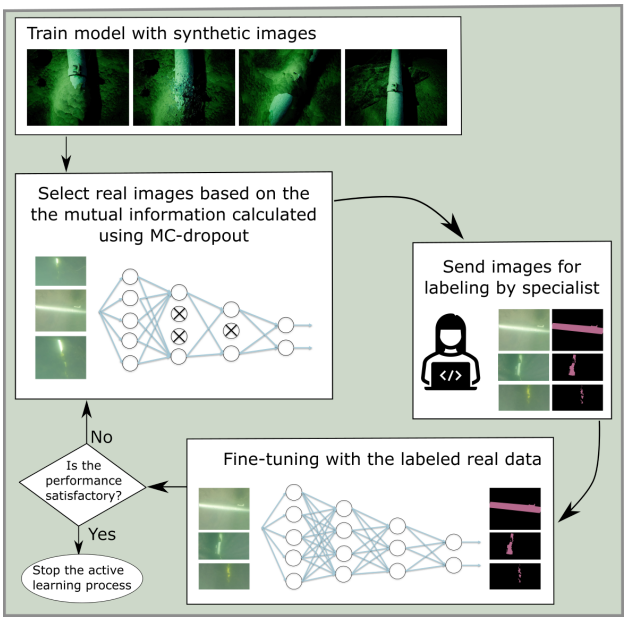


Fig. 1: Sim-to-Real with Active Learning.

Our experiments show that active learning gives better results than random selection for overcoming the sim-to-real gap. Moreover, the model pre-trained with synthetic data and fine-tuned with real data presented better performance than the model trained only with real data in almost all image sequences tested. It is worth using active learning when having a limited budget for labeling images, since it ensures the best outcomes.

II. Background and Related Work

Active Learning methods train machine learning models in an iterative way, beginning with training the model with an initial subset of samples, typically randomly selected, and then retraining it with strategically chosen new samples. To cleverly select these new samples, the uncertainties of the predictions made by the trained model are computed. These uncertainties reflect the knowledge, or lack thereof, that the model has about the input samples. Samples with high uncertainty are those that the model has less knowledge about and, therefore, are more beneficial for using while retraining the model, contrary to the samples with low uncertainty, about which the model already possesses enough knowledge.

Active learning has been extensively studied with the goal of training models with the minimal amount of data necessary to achieve results comparable to the models trained with the entire dataset. In medical images, for

instance, the softmax confidence was used as a measure of uncertainty when segmenting pulmonary nodules [6] and membrane images [7]. However, softmax can be overconfident and present high confidence values during the inference phase for samples that are out-of-distribution in relation to the training dataset [8]. Other studies using medical images applied Monte Carlo Dropout (MC-Dropout) [9] for calculating metrics such as max-entropy and Bayesian active learning by disagreement (BALD) as a measure of uncertainty [10], [11]. In the context of autonomous driving cars, which is closer to underwater inspection, entropy-based metrics were used for selecting images for training segmentation models [12], [13], [14].

Since active learning methods retrain the models with the least amount of data by detecting the samples that the models do not have sufficient knowledge about, they can be applied to help overcome the sim-to-real gap with minimal effort [1]. In this work, we train an underwater pipeline segmentation model using the synthetic dataset MIMIR [15]. Subsequently, we utilize active learning to select the most relevant images in a real underwater dataset for fine-tuning the previously trained model, therefore adapting it to the specific chosen dataset. To calculate the uncertainty used for querying new images with active learning, we employ MC-Dropout, a method largely studied in the deep learning community for accessing the epistemic uncertainty, which arises from the model's lack of knowledge [16], [17]. This method consists of allowing the dropout layers [18] during the inference time. Dropout layers temporarily remove neurons in a specific layer with a chosen probability. It means that when these layers are allowed, the same input can have different outputs if forward passed through the model more than once. These layers are originally used during training to prevent overfitting [18], but can be used for accessing the epistemic uncertainty during inference [9]. If the outputs for several forward passes of the same input are similar, it means that the model has enough knowledge about that input pattern; if the outputs are very different from each other, it indicates a lack of knowledge.

III. Methodology

In this section, we present the methodology and details for our experiments with pipeline image segmentation: Sec. III-A presents the pre-training phase using the synthetic dataset MIMIR [15], Sec. III-B the fine-tuning phase for overcoming the sim-to-real gap, Sec. III-C the details about the model structure and the training, and Sec. III-D the datasets used.

A. Pre-Training with Synthetic Data

The first step of the experiments was to train a segmentation model on the synthetic dataset MIMIR [15]. MIMIR has several environments, and we used the one called SandPipe, which contains a single pipeline on the ocean floor. The images were captured by a camera placed at the bottom of the autonomous underwater vehicle (AUV) in the simulated environment, similar to the position of the camera that collected the real dataset.

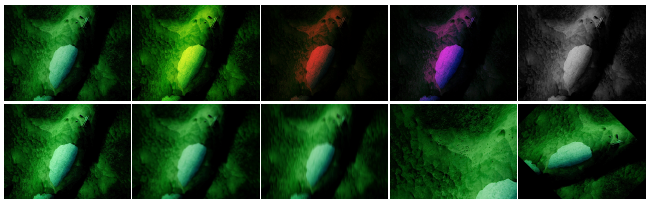


Fig. 2: Examples of augmentations applied to MIMIR. Top: The most left image is the original; the next three are examples of RGB channels perturbation; and the most right is a conversion to grayscale. Bottom (from left-to-right): Value and saturation perturbation; Gaussian blur addition; motion blur addition; resize with cropping; and rotation. In the images with blur addition, notice how the object contours get weaker and how the screw in the pipeline joint "disappears" (Best viewed online in color with zoom-in.)

We chose to use a visual transformer SegFormer [19] for segmentation. We modified it to include dropout layers. The final goal was to use the trained model with real underwater images. To reduce the overfitting to MIMIR [15], smoothing the transition from simulation to reality, several randomized augmentations were performed, Fig. 2:

- Perturbation of the RGB channels and the value and saturation in the HSV color space;
- Resizing, cropping and flipping;
- Conversion to grayscale;
- Addition of motion and Gaussian blur.

B. Fine-Tuning with Real Data

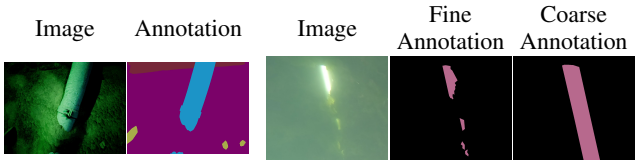
After training the model with MIMIR [15], the active learning method was applied to select the most relevant images from the real underwater dataset for fine-tuning the model. The mean epistemic uncertainty over all pixels of each image was used as the acquisition function for querying new images with the active learning method. For calculating the mean value for each image, the epistemic uncertainty of each pixel was first calculated. This work uses the mutual information, $\mathcal{I}$, calculated with MC-Dropout, as the epistemic uncertainty. The MC-Dropout is applied during the inference phase and consists of forward passing each input sample T times. For a dataset with C classes, at each forward pass t, the model generates for each pixel a softmax output equal to $p_t = (p_{1t}, ..., p_{Ct})$. Using the outputs p_t, the mutual information of each pixel is:

$$\mathcal{I} = \mathcal{H} + \frac{1}{T \log_2(C)} \sum_{c=1}^{C} \sum_{t=1}^{T} p_{ct} \log_2(p_{ct}), \quad (1)$$

where $\mathcal{H}$ is the entropy, calculated as:

$$\mathcal{H} = \frac{-1}{\log_2(C)} \sum_{c=1}^{C} p_c^* \log_2(p_c^*), \quad (2)$$

where $p^* = (p_1^*, ..., p_C^*)$ is the average of the predictions p_t over the T forward passes. Equation (1) and Eq. (2) where divided by $log_2(C)$ to normalize the entropy between 0 and 1.



(a) MIMIR (b) MarinaPipe

Fig. 3: MIMIR and MarinaPipe samples.

Finally, the image uncertainty, EU_{img}, was calculated as:

$$EU_{\text{img}} = \frac{1}{N}\sum_{i=1}^{N} EU_i, \tag{3}$$

where EU_i is the mutual information $\mathcal{I}$ defined in Eq. (1) for the pixel i of an image with N pixels. Images with high EU_{img}, above a chosen threshold TR, were selected for fine-tuning the SegFormer model that was pre-trained with MIMIR [15]. The threshold TR was defined as:

$$\text{TR} = \overline{EU}_{\text{img}} + S\sigma, \tag{4}$$

where $\overline{EU}_{\text{img}}$ and σ are the mean and standard deviation of the values of EU_{img} computed for the MIMIR images using the pre-trained SegFormer. The variable S is a scalar value defined by the user. Smaller values of S ensure a more rigid tolerance with the uncertainty but mean querying more images for being labeled. Notice that two values of TR are calculated, one for the training and the other for the validation dataset. More details on how to query new images with the active learning framework are in our previous work [20].

C. Model Structure and Training Details

The segmentation model used in this study was the visual transformer SegFormer [19] implemented in PyTorch.[1] We modified the structure for including dropout layers in the encoder.

The model was pre-trained from scratch, for 600 epochs, using cross-entropy loss, Adam optimizer, and an initial learning rate of 10^{-4}. During the fine-tuning phase with real data, the decoder and encoder of SegFormer were frozen, and only the head of the model was allowed to train. At this phase, the model was trained for 100 epochs, using cross-entropy loss, Adam optimizer, and an initial learning rate of 10^{-5}. During both pre-training and fine-tuning, the classes used were background and pipeline.

D. Datasets

Two datasets were used in this paper, cf. Fig. 3.

1) MIMIR: a synthetic multipurpose dataset originating in a prior study [15], tailored for pipeline tracking, created in a simulation environment with automatic pixel-wise labeling for many classes, including pipeline. MIMIR has several environments, with SandPipe being one of them. SandPipe has images of a single pipeline, positioned on the ocean floor. This environment has images recorded from three cameras in a simulated AUV. In this study, we use the images from SandPipe recorded from the bottom of the AUV facing down the ocean floor. These images were selected because the images on the MarinaPipe dataset are visually closer to them than to the images recorded by the other cameras and on the other environments of MIMIR.

TABLE I: Details of MarinaPipe. (*Both* refers to fine and coarse labeling.)

Video	Selected frames	Frames with pipes	Annotation	Occlusions
1	236	43	Both	Yes
2	237	70	Both	No
3	260	2	Both	No
4	268	11	Both	Yes
5	266	45	Coarse	Yes
6	270	11	Coarse	Yes
7	186	17	Both	Yes

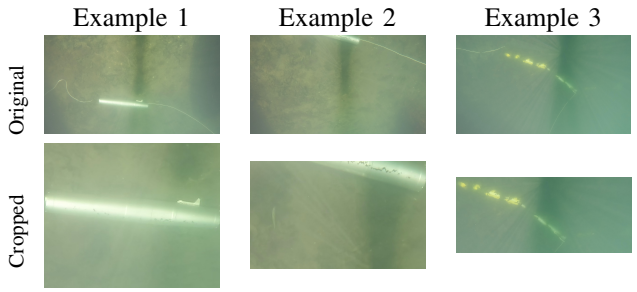


Fig. 4: Examples of how the video frames were cropped before being labeled. Top: Original frames. Bottom: Correspondent cropped frames.

2) MarinaPipe: a real underwater dataset, recorded in a marina close to the north of Portugal, by our partner OceanScan-MST. The dataset contains pieces of pipes placed on the marina floor, filmed using a GoPro camera attached to a lightweight autonomous underwater vehicle (LAUV). Seven videos were recorded at 240 frames per second (FPS), from which we extracted five frames per second to create the dataset. In some videos, the pipes are partly occluded by algae. For performing the experiments, which the results are described in Sec. IV-B, 10% of the frames of each video were labeled for the task of pipeline segmentation. Table I provides an overview of the MarinaPipe dataset. This dataset was originally idealized for training a model that would later be tested for tracking long pipelines. Because of this, the extracted frames were cropped before being labeled, so that the pipe goes through the image, as Fig. 4 shows. The link for downloading MarinaPipe can be found in the REMARO GitHub.[2] We are releasing the original videos, the extracted frames, Tbl. I, and the frames' pixel-wise fine and coarse annotation for the pipeline class in the format of masks.

[1]Based on the implementation from https://github.com/FrancescoSaverioZuppichini/SegFormer/

[2]https://github.com/remaro-network/MarinaPipe-dataset

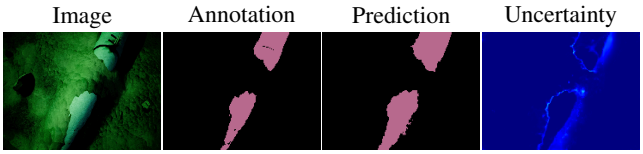


Fig. 5: Example of prediction on the test dataset of MIMIR using the model trained on MIMIR. The pipeline and background color are different from Fig. 3 because MIMIR has annotation for many classes and we are only using the pipeline class. Everything that is not pipeline is defined as background in this paper. For the uncertainty plot, calculated as the mutual information, the warmer colors represent higher values.

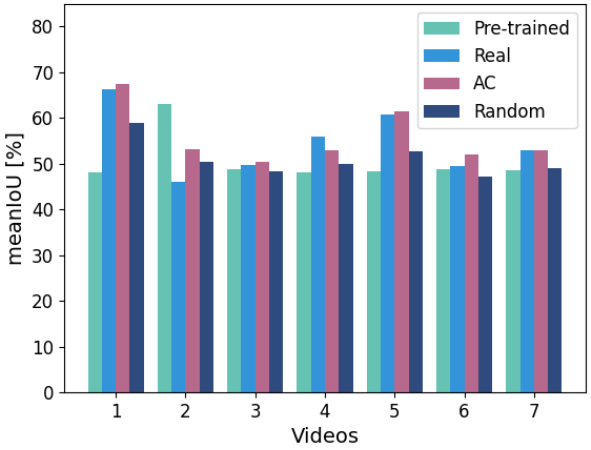


Fig. 6: Experimental results: meanIoU for the classes pipeline and background. Results were calculated using the coarse annotation as ground truth.

IV. Results

The sections below present the results obtained from our experiment for overcoming the sim-to-real gap. It also includes a study about the influence of data augmentation, freezing and unfreezing the decoder layers, the learning rate values, and the type of annotation used (fine or coarse).

A. Pre-training with MIMIR

From the SandPipe images selected for this study, 90% were reserved for training, 5% for validating, and the other 5% for testing SegFormer. From the data reserved for training, part of the images containing only background were eliminated to diminish the imbalance between this class and pipeline. After training, the model obtained 88.80% mean intersection over union (meanIoU) on the test dataset, with 81.05% intersection over union (IoU) for the pipeline class and 96.56% for the background. Figure 5 showcases an example of prediction using the pre-trained SegFormer on the MIMIR test subset.

B. Fine-tuning with real data

The experimental results are presented in Fig. 6, where the legends refer to the following set-ups:

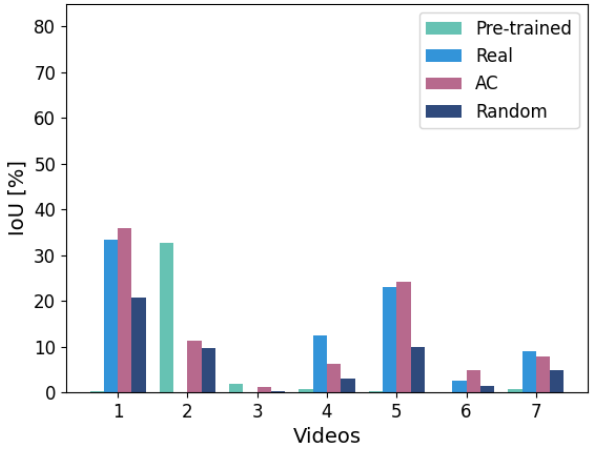


Fig. 7: Experimental results: IoU of pipeline. Results were calculated using the coarse annotation as ground truth.

- *Pre-trained* the model trained only on the MIMIR dataset at Sec. IV-A.
- *Real* the model was trained with the data from video 1, Tbl. I, and validated with video 7. Only part of the frames not containing pipelines were used, for diminishing the class imbalance. As augmentation, it was only applied resizing, cropping and flipping.
- *Random* the model was pre-trained with MIMIR, model from Sec. IV-A, and then fine-tuned with ca. 45% of images randomly selected from videos 1 and 7 for training and validation, respectively. The fine annotation was used as ground truth.
- *AC* the model training and fine-tuning was done as in *Random*, however the images were chosen with active learning, instead of at random.

The meanIoU results in Figure 6 were calculated using the coarse annotation as ground truth. For both *random* and *AC*, the same random augmentation techniques from Sec. III-A were applied to the training and validation images used to fine-tune the model.

For selecting new images with active learning, *AC*, we set $S = 3.0$ in Eq. (4). This parameter choice resulted in 110 images selected from MarinaPipe for training and 79 for validation. The same number of images were selected for training and validating the *random* model.

We found that fine-tuning the model with real images always reduces the sim-2-real gap. Figure 6 and Figure 7 show that active learning (AC) consistently outperforms random selection (Random). The results of the pipeline class in Figure 7 leave room for improvement. The pipeline's low IoU may be due to the dataset's complex patterns, which may require more annotated data to improve the model's performance. Even though the IoU for the pipeline is low, notice that the model fine-tuned with active learning (AC) can recognize this object, Fig. 8.

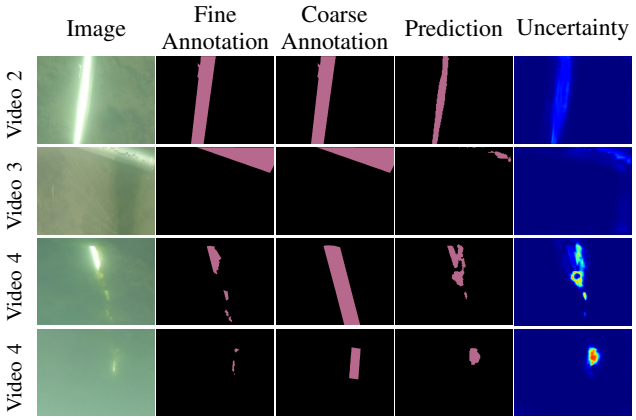


Fig. 8: Examples of predictions using the model fine-tuned with active learning, *AC*, and the respective mutual information uncertainties. Warmer colors represent higher uncertainty and reflect the models' difficulty when predicting pipelines.

C. Study of the augmentation, training and structure choices

We fine-tune the model with very few data. To increase the model's performance, we used many augmentations and froze the entire encoder-decoder structure during the fine-tuning. Now, we present some comparisons between choices made during the fine-tuning phase. All the analyses were performed using the same frames used for training and validating the model *AC* in Sec. IV-B.

1) Data augmentation: Instead of using all the augmentations listed above, we now only apply resizing, cropping, and flipping to the data used for fine-tuning the model to test if so many augmentations were confusing the model.

2) Freeze vs. unfreeze the decoder: For analyzing the benefits of freezing the decoder during the fine-tuning phase, now only the encoder was frozen, allowing the decoder and the model's head to train.

3) Learning rate: For analyzing the initial learning rate choice during the fine-tuning, we test setting the initial value to 10^{-4} instead of 10^{-5}. Both the encoder and decoder were frozen, and only the head was allowed to train.

Figure 9 shows the results for the last three topics mentioned. As the figure shows, decreasing the amount of augmentation and unfreezing the decoder decreased the model's performance. Increasing the initial learning rate gave slightly better results. This was the only model fine-tuned with an initial learning rate equal to 10^{-4} in this study.

4) Test with fine annotation: As mentioned before, in Sec. IV-B, the models were fine-tuned using the fine annotation as ground truth; however, during the inference phase, the performance was analyzed using the coarse annotation. This choice has two reasons: (1) videos 5 and 6 only have the coarse annotation for evaluating the performance, and (2) apparently, the model learned how to extrapolate the fine annotation, and the results are better when compared to the coarse annotation.

5) Learning with coarse annotation: Since evaluating the performance in the coarse annotation gave better results, we wonder if training on coarse annotation would result in a better model. However, as Fig. 10 shows, the results of training using the coarse annotation were worse. We hypothesize that this is the case because the fine annotation gives a more "precise" label, and the model learns better to differentiate the pipeline from the rest of the image.

Figure 10 shows the results obtained for the tests in Sec. IV-C.4 and Sec. IV-C.5.

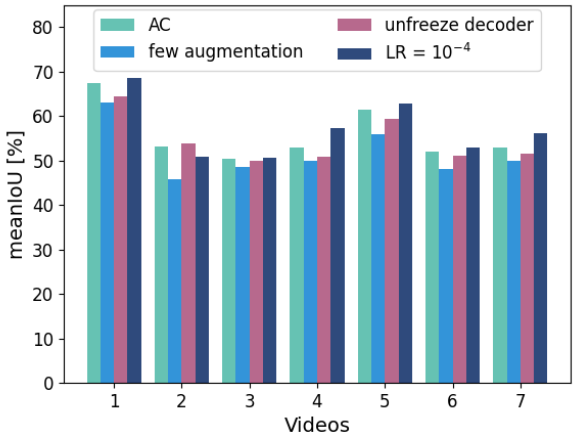


Fig. 9: These results analyze the influence of the choices for augmentation techniques, initial learning rate, and the option of freezing the model's decoder. *AC* refers to the model fine-tuned in Sec. IV-B.

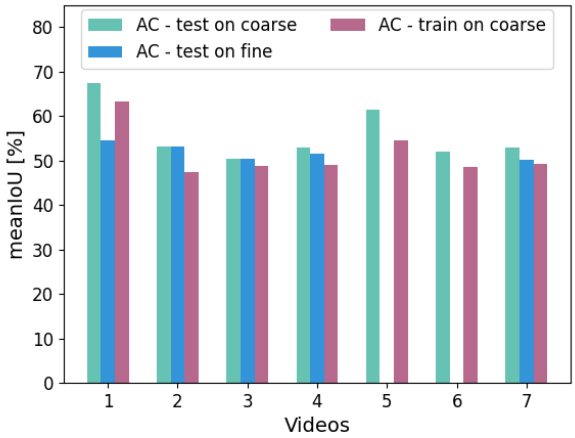


Fig. 10: *Test on coarse* and *test on fine* use the same model, trained in Sec. IV-B, but one is evaluated using the coarse annotation as in the referred section and the other using fine annotation. *Train on coarse* was trained and evaluated with coarse annotation. Notice that videos 5 and 6 do not have the fine annotation for performing the evaluation.

V. Conclusion

Active learning is more efficient in reducing the sim-to-real gap than fine-tuning with random images. The

MarinaPipe dataset has a lot of motion blur and uneven illumination, which could be the reason for the pipeline class's low IoU. Thus, MarinaPipe could be considered an open challenge to the underwater computer vision research community. SegFormer was trained with MIMIR from scratch and then fine-tuned with MarinaPipe. An interesting next test is to pre-train the model with a larger dataset, such as COCO, before using MIMIR, and evaluate if it would result in better IoU for the pipeline class. Even though it was demonstrated that SegFormer can be used with active learning, more tests should be performed to study the best positions to insert the dropout layers in this structure. In future work, we plan to select more images for labeling, and rerun the experiments with the additional annotated data.

ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 956200. We thank OceanScan - Marine Systems & Technology Lda for recording the real pipeline dataset. We thank Olaya Álvarez-Tuñón, László Antal, Martin Aubard, Sergio Quijano, Maria Costa, João fonseca and Renato Campos for the fruitful discussions during the development of this work.

REFERENCES

[1] J. Feng, J. Lee, M. Durner, and R. Triebel, "Bayesian active learning for sim-to-real robotic perception," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10820–10827, 2022.

[2] R. Schettini and S. Corchs, "Underwater image processing: State of the art of restoration and image enhancement methods," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, p. 746052, 2010.

[3] A. Duarte, F. Codevilla, J. D. O. Gaya, and S. S. C. Botelho, "A dataset to evaluate underwater image restoration methods," in *OCEANS 2016 - Shanghai*, pp. 1–6, IEEE, 2016.

[4] J. Y. Chiang and Ying-Ching Chen, "Underwater image enhancement by wavelength compensation and dehazing," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1756–1769, 2012.

[5] T. Li, S. Rong, L. Chen, H. Zhou, and B. He, "Underwater motion deblurring based on cascaded attention mechanism," *IEEE Journal of Oceanic Engineering*, 2022.

[6] W. Wang, R. Feng, J. Chen, Y. Lu, T. Chen, H. Yu, D. Z. Chen, and J. Wu, "Nodule-plus R-CNN and deep self-paced active learning for 3D instance segmentation of pulmonary nodules," *Ieee Access*, vol. 7, pp. 128796–128805, 2019.

[7] U. Gaur, M. Kourakis, E. Newman-Smith, W. Smith, and B. Manjunath, "Membrane segmentation via active learning with deep networks," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 1943–1947, 2016.

[8] Y. Gal, *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[9] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[10] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," in *International conference on machine learning*, pp. 1183–1192, PMLR, 2017.

[11] I. C. Saidu and L. Csató, "Active learning with Bayesian UNet for efficient semantic image segmentation," *Journal of Imaging*, vol. 7, no. 2, p. 37, 2021.

[12] S. Xie, Z. Feng, Y. Chen, S. Sun, C. Ma, and M. Song, "Deal: Difficulty-aware active learning for semantic segmentation," in *Proceedings of the Asian conference on computer vision*, 2020.

[13] D. Sreenivasaiah, J. Otterbach, and T. Wollmann, "Meal: Manifold embedding-based active learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1029–1037, 2021.

[14] A. Rangnekar, C. Kanan, and M. Hoffman, "Semantic segmentation with active semi-supervised learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5966–5977, 2023.

[15] O. Álvarez Tuñón, H. Kanner, L. Ribeiro Marnet, H. Xuy Pham, J. le Fevre Sejersen, Y. Brodskiy, and E. Kayacan, "MIMIR-UW: A multipurpose synthetic dataset for underwater navigation and inspection," *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023 (In publication).

[16] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021.

[17] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, Jan. 2014.

[19] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12077–12090, 2021.

[20] L. Ribeiro Marnet, Y. Brodskiy, S. Grasshof, and A. Wąsowski, "Uncertainty driven active learning for image segmentation in underwater inspection," in *Proceedings of the 4th International Conference on Robotics, Computer Vision and Intelligent Systems*, Springer Nature, Feb 2024.