SeaDroneSim: Simulation of Aerial Images for Detection of Objects Above Water

Xiaomin Lin[1], Cheng Liu[1], Allen Pattillo[2], Miao Yu[3], Yiannis Aloimonous[1]
[1]Perception and Robotics Group, University of Maryland - College Park, MD
[2] University of Maryland Extension, MD
[3] Mechanical Engineering, University of Maryland - College Park, MD
Email:{xlin01, cliu2021, dapatt, mmyu, jyaloimo}@umd.edu

Abstract

Unmanned Aerial Vehicles (UAVs) are known for their speed and versatility in collecting aerial images and remote sensing for land use surveys and precision agriculture. With UAVs' growth in availability and accessibility, they are now of vital importance as technological support in marine-based applications such as vessel monitoring and search-and-rescue (SAR) operations. High-resolution cameras and Graphic processing units (GPUs) can be equipped on the UAVs to effectively and efficiently aid in locating objects of interest, lending themselves to emergency rescue operations or, in our case, precision aquaculture applications. Modern computer vision algorithms allow us to detect objects of interest in a dynamic environment; however, these algorithms are dependent on large training datasets collected from UAVs, which are currently time-consuming and labor-intensive to collect for maritime environments.

To this end, we present a new benchmark suite, ***SeaDroneSim****, that can be used to create photo-realistic aerial image datasets with ground truth for segmentation masks of any given object. Utilizing only the synthetic data generated from* ***SeaDroneSim****, we obtained 71 a mean Average Precision (mAP) on real aerial images for detecting our object of interest, a popular, open source, remotely operated underwater vehicle (BlueROV) in this feasibility study. The results of this new simulation suit serve as a baseline for the detection of the BlueROV, which can be used in underwater surveys of oyster reefs and other marine applications.*

Index Terms—***simulation, aerial images, automation, BlueROV, detection***.

1. Introduction

Unmanned Aerial Vehicles (UAVs) equipped with high-resolution cameras and Graphics Processing Units (GPUs) are increasingly used in a variety of tasks, ranging from land-use classification [7, 4], precision agriculture [22, 26, 7], coastal management [3], crowd surveillance [10, 20], and recently in assisting with search and rescue (SAR) missions [5, 6, 12, 13, 19] and post-disaster exploration missions by providing a bird's eye view over the scene. Specifically, in marine scenarios, where a large ocean area needs to be searched quickly, UAVs provide vital aid to emergency search and rescue operations [31]. For SAR missions, a ro-

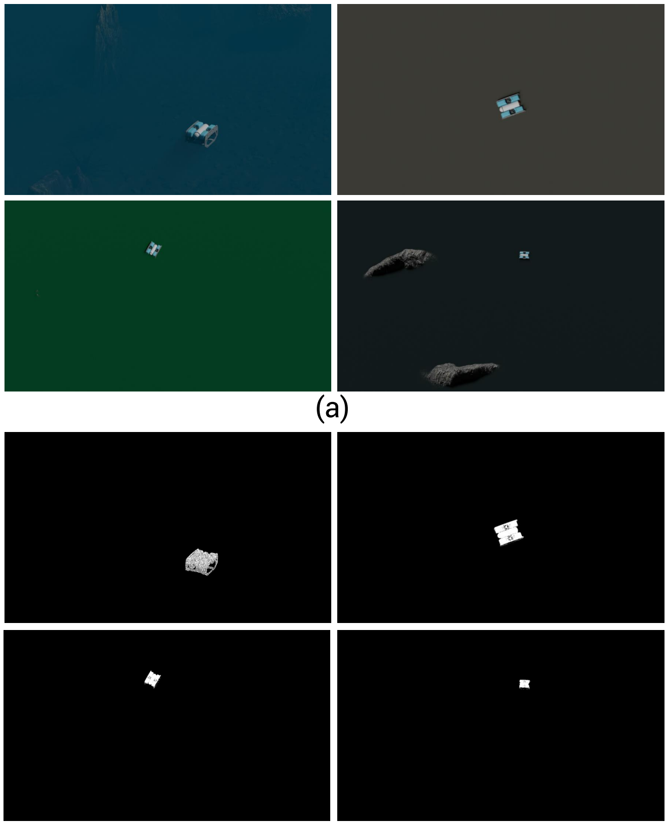

Figure 1. (a) Typical image examples with varying altitudes, angles of view and background water color. (b) Corresponding ground truth masks for the object of interest (BlueROV).

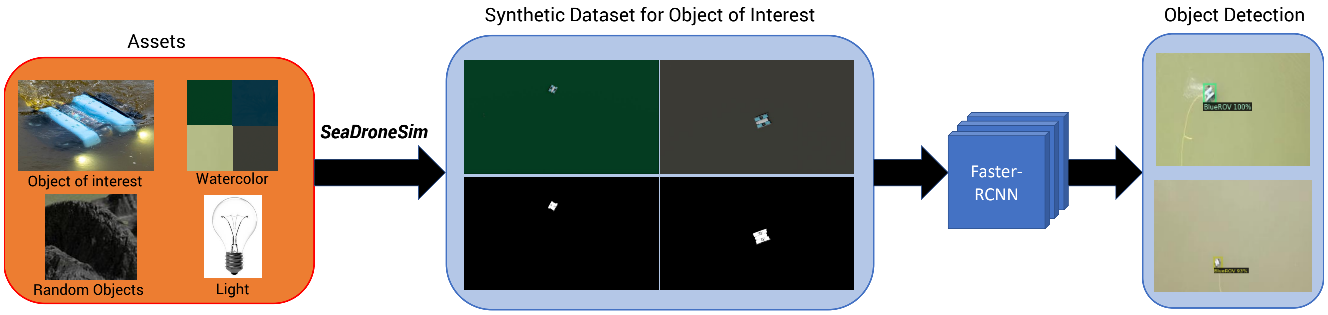

Figure 2. An overview of our approach: 3D models for the object of interest are used in *SeaDroneSim* to generate Synthetic datasets. Noted the synthetic dataset would include its ground truth mask for the object of interest. The synthetic dataset generated is then fed into a Neural network to obtain the object detection result. Note: Object Detection images are cropped and enlarged for better visualization.

bust vision-based system is needed to locate/track objects of interest in maritime environments with different lighting conditions, image altitudes, viewing angles, and water colors. Currently, most of these are vision-based, data-driven systems employing deep neural networks. These systems are dependent on large datasets to correctly recognize objects under diverse environmental conditions. However, there are only a few datasets available, which are limited in their size and variety of objects of interest. The process of collecting large-scale datasets for specific object detection tasks is slow, time-consuming, and has poor scalability. Using a manual approach, only a few objects of interest can only be located/tracked within a specific dataset for that task. Varga [27] proposed a relatively large dataset for the Maritime environment, yet with a nearly infinite number of potential objects of interest there is a need for methods to rapidly generate large-scale datasets in maritime environments for a variety of objects, including our object of interest, the remotely operated underwater vehicle - BlueROV Fig. 3.

Similar to Kiefer et al. [14], this work aims to help fill the gap of large-scale maritime-based datasets captured from UAVs. We introduce a simulation suit for generating simulated images of maritime-based aerial images, called ***SeaDronesSim***. We are capable of generating videos and images of any object of interest with a 3D model Fig. 3 in open water with various lighting conditions, image altitudes, viewing angles, and background water colors (see Fig. 1(a)). To fully utilize the *SeaDronesSim*, the ground truth for the object of interest is generated at the same time (see Fig. 1(b)). Most SAR missions require a vision-based system to be able to detect and track from a great distance. *SeaDronesSim* is able to vary its resolution for the Red, Green, and Blue (RGB) footage captured. To study the feasibility of object recognition purely from synthetic images generated from *SeaDronesSim*, we used our object of interest, the BlueROV, for which we have a detailed 3D computer model on hand for simulating various image poses from Patrick[21]. Moreover, the simulation provides all the ground truth information of the aerial camera (altitude taken, rotation) and the BlueROV (rotation), allowing us to output all the metadata for the aerial camera and the objects of interest. We believe all this metadata could be used in the future to develop multi-modal systems to improve accuracy and speed. Finally, we conducted extensive experiments to compare the datasets that we generated using a state-of-the-art model and established baseline models for BlueROV detection. These results serve as a starting point for our *SeaDronesSim* simulation suit. To streamline the process of object detection, our goal is to utilize advancements in robotics and artificial intelligence that enable us to gather UAV-based aerial images from the maritime environment to automate the process of object detection and tracking. *SeaDronesSim* is among the first simulations for UAV-based aerial images from the maritime environment [14]. Our main contributions are as follows:

- We propose a novel simulation suit for generating UAV-based aerial images for the maritime environment.
- To the best of our knowledge, we established a baseline for detecting BlueROV in open water.
- We open-source [1]our *SeaDronesSim* and dataset associated with this work to accelerate further research.
- We proposed a pipeline for autonomously generating UAV-based maritime images for objects of interest and detecting them.

The rest of this paper is organized as follows: We first place this work in the context of related works in Sec. 2. Then, we describe the proposed simulation which is used to create realistic images in Sec. 3. We then present some quantitative and qualitative evaluations of our approach in Sec. 4. We conclude our work in Sec. 5 with parting thoughts on future work.

[1] https://github.com/prgumd/SeaDroneSim

2. Related work

In this Section, we first review some of the major datasets for aerial images and maritime environments. Then we describe some of the simulations for the maritime/aerial domain.

2.1. Datasets for Maritime Environments

Large-scale datasets are necessary when developing modern computer vision algorithms. However, many datasets for maritime environments are satellite-based synthetic aperture radar imagery and are used for remote sensing tasks. In 2018, Airbus [1] released 40k satellite images (with instance segmentation labels) from synthetic aperture radars for ship detection. Gallego[11] also proposed a method for autonomous ship detection with aerial images. Li et al. [15] proposed another dataset for ships consisting mainly of Google Earth and some UAV-based images. In 2019, Lygouras et al. [18] worked on human detection with UAV-based images. However, Lygouras' dataset was limited, as the dataset is collected either near-shore or in a swimming pool. Keifer et al. [14] analyzed real and synthetic maritime and terrestrial images from both satellite and UAV sources for detecting boats and people. Another UAV-based image dataset, released by Varga [27], is tailored toward recognizing objects in the water and collecting large-scale data for the maritime environment. However, Varga's dataset only provides six classes of objects. There are many more objects of interest in the SAR tasks such as hovercraft, floating planes, humans waving for help, etc. There is no dataset that can represent all objects of interest for every conceivable detection task, thus the need for an automated process to generate this data synthetically as needed.

2.2. Simulation

Instead of collecting large-scale datasets for detection tasks, we follow the conceptual approach of detecting oysters [16][17] and propellers [24] which utilizes a 3D model of the object to generate an enormous amount of data synthetically. In particular, we used a 3D model of the underwater BlueROV robot to create a maritime dataset for BlueROV detection.

When there is a lack of large-scale datasets for some robotics tasks, different research groups have developed various simulations to meet their needs. Most simulators for the aerial and maritime domains focus on controlling the drone for safety operations [9] and rapid control [28]. Abujob [2] used simulation to verify his algorithm of landing the drone on the ship with the prediction of ship motion. The most similar simulation to *SeaDroneSim* is from Matlab UAV Toolbox [25] which provides control of the drone and also images from a camera attached to the drone. However, this work focuses on controlling the drone for terrestrial applications and does not provide segmentation ground truth for the object of interest.

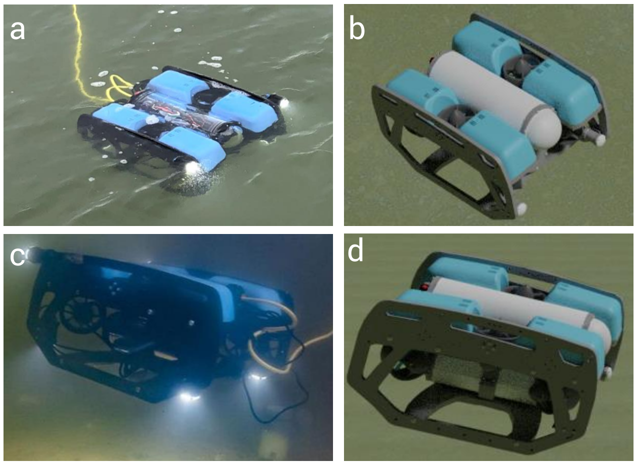

Figure 3. (a) BlueROV in water near Horn Point Lab, (b) BlueROV in Blender^{TM}, (c) BlueROV in water in Horn Point Lab, (d) BlueROV in Blender^{TM}.

With the ultimate goal of developing an autonomous UAV-based surveillance system, we acknowledge the need for object detection methods. Moreover due to the lack of large-scale datasets for the maritime environment and the lack of literature addressing this problem, we are seeking an alternative path for generating maritime datasets through synthetic image generation. To the best of our knowledge, we are among the first to propose a simulator (*SeaDroneSim*) for maritime environment datasets and use that data to recognize objects of interest. *SeaDroneSim* will be described in the next Section.

3. Proposed Approach

In this work, we propose a novel simulation built from Blender^{TM} [8] game engine. Then we used the simulation for creating UAV-based maritime images which are then used to train a Neural Network for object detection. As we can see in Fig. 2), we first choose our object of interest and different assets(i.e. water color, light, random objects) and input them into the simulation engine that we build. Moreover, utilizing the data produced from the simulation engine, we can come up with a detection network for detecting objects of interest. In this section, we will go through some of the details of the implementation of the *SeaDronesSim*.

3.1. Object of Interest

It is critically important to have an accurate 3D model of the object for this work. Only with an accurate 3D model of the object in the simulation, can we correctly represent the visual appearance from the UAV-based maritime images. In this work, we select BlueROV as our object of interest because of the availability of its 3D model[21] and actual ob-

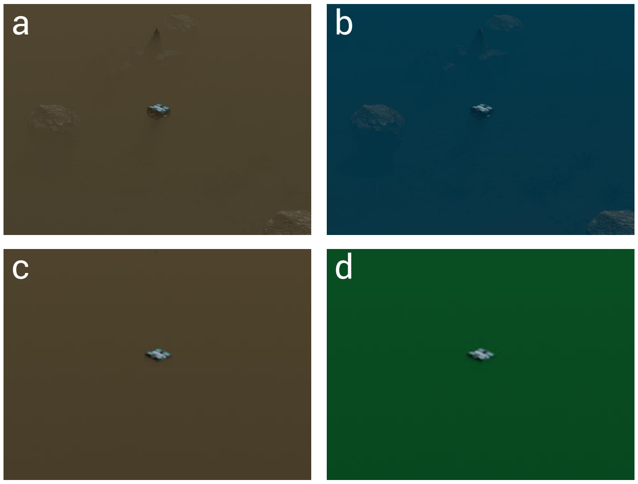

Figure 4. Different water colors and turbidity levels.

ject for collecting real datasets. BlueROV can also be used for many marine tasks, i.e., oyster monitoring[17] and navigation. Our group potentially wants to do marine-aerial navigation like Wang[29] has done for ground-aerial navigation. Detection of the BlueROV becomes necessary for such tasks.

As we can see in Fig. 3, the 3D model in the simulation (Fig. 3(b)(d)) looks very similar to real BlueROV in the water (Fig. 3(a)(c)). Although there might be some visual difference between the images from the real data and the synthetic data, the actual object size in the UAV-based image is relatively small, so we can neglect the visual difference here. The relatively small size of the object in UAV-based images makes it possible for rendering the datasets in *SeaDronesSim*, and directly use the datasets for training a network to detect the real object.

3.2. Water Volume

The major distinction between an on-land simulation and a maritime simulation is the water volume. In the Blender^{TM} [8]environment, the render engine ”CYCLE” uses path-tracing of the lights to generate images. In other words, it passes light with a certain number of bounces and receives light to generate images of different brightness, colors, and shades. By customizing particles in the water volume, we get a photo-realistic maritime image for different water colors and turbidity levels. The color of the water volume can also be modified to mimic a specific maritime environment's color scheme. As we can see in Fig. 4(a) and Fig. 4(b), these are UAV-based maritime images of different water colors in low turbidity. As we increase the turbidity level for the water, the object we added at the bottom (visible in Fig. 4(a)) is no longer visible in Fig. 4(c). Another image with different water colors in high turbidity is shown in Fig. 4(d).

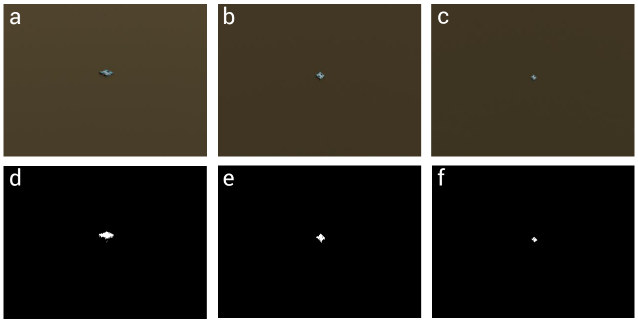

Figure 5. Images from different altitudes. (a) Image from an altitude equal to 20 m, (b) image from an altitude equal to 30 m, (c) image from an altitude equal to 40 m, (d)(e)(f) are the corresponding ground truth masks for (a)(b)(c).

3.3. Aerial Image

We want to simulate the aerial Images in *SeaDroneSim*. However, we do not care about the UAV's appearance and its physical model. The only thing that we care about is the camera attached to it. So we basically modeled the drone as a point object with a camera attached to it (we will just refer to the point object as the drone). Instead of moving the drone or the object in the simulation, we choose to move the object and fixate the camera on the object. In this case, when the object is closer to the drone, the object will look larger so that we can obtain multiple sizes within one run. However, this will not cover all possible sizes that we may want, so we adjust the altitude of the drone to obtain different images. As we can see in Fig. 5, the object and its corresponding ground truth mask in the image become smaller when we increase the altitude when taking the images. The resolution of the image could go up to 30,000 x 30,000 pixels depending on the computational power of the hardware that is running the simulation.

3.4. Objects of Non-Interest

To simulate some of the scenarios in that some objects not of interest are included, we add a few objects at the bottom of the seafloor and some hills above the sea level as we can see in Fig. 1(a).

4. Experiments and Results

We perform a comprehensive comparison with various training data by utilizing Faster R-CNN [23]. By rendering the training data in *SeaDroneSim*, it can provide any number of images under different sizes, altitudes, angles, and colors. We resized the images to 416 x 416 pixels. For each masked image, the bounding boxes can be determined by the highest and the lowest x and y coordinates of BlueROV in the image. With these x and y coordinates, we added a BlueROV category and the corresponding bounding boxes. Therefore, each image receives a COCO-dataset format label. Instead of implementing a Faster R-CNN model from

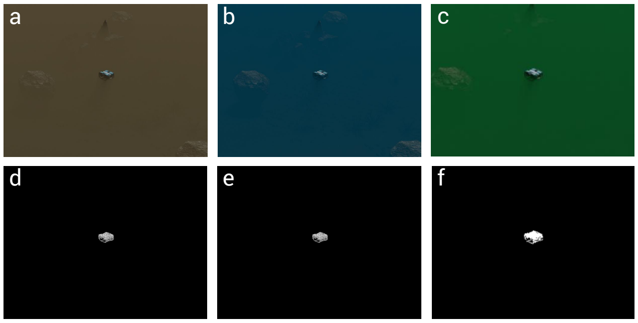

Figure 6. Examples of original and ground truth mask images. These examples are based on different background water colors brown (a, d), blue (b, e), and green (c, f), respectively.

scratch, we used Detectron2 [30] which allows us to accelerate the developing process. Detectron2 is a novel system created by Facebook AI Research (FAIR) for computer vision research such as object detection and segmentation.

4.1. Synthetic Dataset

There are 626 images for each dataset generated with the same parameters/constraints unless specified differently. For all the images from the simulation, we took 80% for training and 20% for validation. Fig. 5 shows the altitude images in the training dataset. To avoid the influence of noise, we removed the environment in the simulation. Therefore, it was not affected by other factors except for BlueROV. In the second part, we created a few datasets with different water colors in the simulation environment. There were 626 images for each color with multiple angles and altitudes in the maritime environment. Fig. 6 shows the example images of different water colors in the training dataset. We could mimic the real UAV-based maritime images by modifying the water colors and adjusting the altitude for the image taken.

4.2. Base model

In this experiment, we used Faster R-CNN with X101-FPN as the backbone to train our dataset. Though X101-FPN takes more time and memory during training, it can produce higher accuracy results compared to other models in Faster R-CNN. In the model, we set up the learning rate of the model as 0.001. To prevent overfitting of the model to the synthetic data, the maximum number of iterations was 1,500, which can be adjusted according to the performance of validation. Since we only focused on detecting BlueROV in the image, there is only one class in this recognition task.

With dataset and model, it is convenient to train custom models in Detectron2. Registering the coco instance function, we uploaded training, validation, and testing data to the Detectron2 platform. Then we created a class and imported Faster R-CNN as our training model. A function exists that allows us to access the parameter of the model. From this function, we can assign the value to the parameter. The checkpoint and weight were saved in the output file. The checkpoint was provided to us to continue training the data after interrupting the training process.

4.3. Evaluation

To prevent the convolutional neural network from overfitting to the synthetic dataset for a relatively easy task, we did not provide a large quantity of training data. However, these training data performed at least 95% Average Precision *(AP)* in the validation. Therefore, the next step is to evaluate the testing data. Testing data was collected from Sandy Point Park in Annapolis, MD. Fig. 7 shows some of the real UAV-based maritime images. We used DJI pro2 [cite] and took a few video clips. We then carefully labeled 2,000 image frames selected from the video as the testing data to produce convincing results.

Table 1 displays the output of the testing data with different training data obtained from various altitudes. Samples of images taken from altitudes 20 m, 30 m, and 40 m can be seen in Fig. 5(a), Fig. 5(b), and Fig. 5(c), respectively. The highest mean Average Precision (mAP) is 58.41 at an altitude of 10 meters. Higher altitude results in a smaller object. Therefore, 10 meters altitude images contribute to the higher accuracy to match the testing data compared with others.

Table 2 presents the evaluation results for different water color training. As shown in Fig. 6, we applied three common water colors: brown, blue, and green. Each color of training data contains different sizes and angles of the BlueROV, as mentioned in Sec. 3.3, to evaluate the result. Some of the results are displayed in the second row of Fig. 7. The brown training data achieved the highest mAP in Table 2, followed by blue, whereas green was much lower. This may be because the water color of our field data collected from Sandy Point Park was more brown, which may have made the training more robust to detect with a brown

Table 1. The Average Precision *(AP)* of the training data with various BlueROV image altitudes.

Altitude	AP	AP_{50}	AP_{75}	AP_s	AP_m
10 m	58.41	90.90	72.62	57.82	84.72
20 m	47.46	95.03	31.28	47.27	57.41
30 m	32.70	91.04	1.41	32.63	40.79
40 m	16.81	78.68	0.07	16.78	30.76
50 m	6.87	48.32	0.01	6.82	22.50

Table 2. The Average Precision *(AP)* of the training data with different water colors at an image altitude of 10 m.

Water Color	AP	AP_{50}	AP_{75}	AP_s	AP_m
Brown	58.41	90.90	72.62	57.82	84.72
Blue	54.72	89.12	65.86	54.70	73.88
Green	57.71	89.24	71.33	57.51	78.73

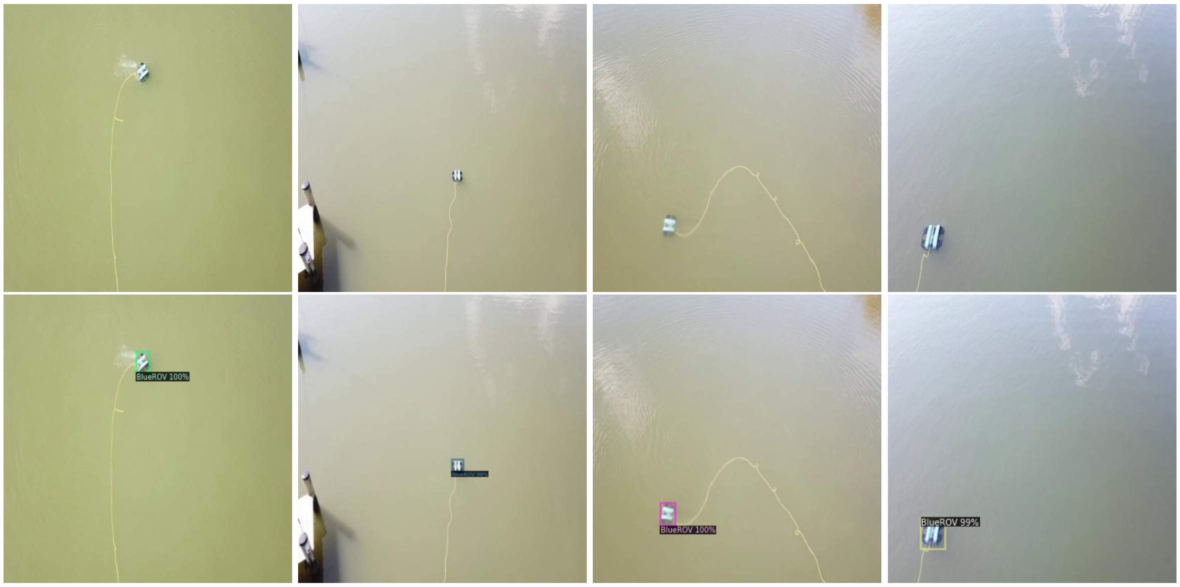

Figure 7. The first row is the examples of testing images. The second row is the testing results for the above images. These examples are collected from Sandy Point Park in Annapolis, MD.

background. Alternatively, brown may simply provide a more distinguishable contrast color to that of the BlueROV, making is easier to detect. The reduced performance using the green dataset could be an issue with the model's ability to distinguish between the color of the BlueROV and the color of the maritime environment.

We also perform a few experiments with different data sizes. We suspect that if the dataset is too large for our detection task, the network is likely overfitting to recognize objects in the synthetic domain. We use image capture for brown water color with an altitude equal to 10 meters. Then, we use the same network with the same number of iterations for training and only vary the images used in the training. As we can see, more data does not necessarily produce better results. A dataset of size 2500 is overfitting the network for the detection of the BlueROV. We also use 137 images generated from green water color, namely "G" in Table 3. Data from a different domain(water color) could improve the neural network's robustness and versatility. The result we obtained with a data size of 308 added G has 71 mAP which has a 20.5% improvement over the result without adding different water color data. We have also performed training with only real data which is added in the appendix Table 6. Without actually training on any real data, we are able to get very close to the training with real dataset(77.56% mAP). The same experiments for Table 1 and Table 2 are performed with more training data. These results are shown in the appendix.

Table 3. The Average Precision *(AP)* of the training data with different data size at an image altitude of 10 m of Brown water color

Dataset Size	AP	AP_{50}	AP_{75}	AP_s	AP_m
308	58.90	94.75	72.32	58.73	80.59
626	58.41	90.90	72.62	57.82	84.72
2500	45.38	95.52	25.13	45.06	76.88
308+G	71.00	96.42	87.24	70.94	80.08
626+G	64.87	98.27	77.75	65.04	70.69

5. Conclusion and future work

In this work, we utilized the capability of a game engine and built a simulation for rendering UAV-based maritime images. We discussed some of the implementation details for using Blender$^{\text{TM}}$ for generating synthetic datasets for object detection. One of the 3D models for the object of interest, BlueROV, is used as a feasibility study for the systematic automation approach for rendering UAV-based maritime images and then detection. We then compared our detection result with the usage of different synthetic datasets generated from the simulation. With only the usage of the generated synthetic dataset, we then set the benchmark for detecting BlueROV in maritime UAV-based images as 71.00 mAP. These results highlight that for data-critical applications when collecting real images is challenging, it is possible to use 3D model of the object to create photorealistic images that will successfully detect objects in a specific domain. This work is among the first research to build a UAV-based maritime image simulation focusing on object detection. In this new field, there are many directions and possible improvements that can be made to our *SeaDroneSim* framework. One possible future work would be obtaining the various 3D models for SeaDroneSee [27] datasets and then rendering the datasets in *SeaDroneSim*. It shall further increase *SeaDroneSim's* utilization scenarios and prove the practicality of our simulation. As for the detection phase, we adopted the existing network. New network architectures can be developed to tackle the UAV-based maritime domain problem. Finally, more object noise can be introduced into the simulation for more robust detection. Further, detection of objects from a single UAV could be applied to a swarm of UAVs units simultaneously, improving search capability and efficiency. Finally, detection angles from multiple UAVs could make the precise Geo-location of objects possible when traditional GPS methods are not possible.

6. ACKNOWLEDGMENT

This work is supported by ”Transforming Shellfish Farming with Smart Technology and Management Practices for Sustainable Production” grant no. 2020-68012-31805/project accession no. 1023149 from the USDA National Institute of Food and Agriculture.

References

[1] Airbus ship detection challenge, Oct 2018.

[2] Shadi Abujoub, Johanna Mcphee, Cassidy Westin, and Rishad A Irani. Unmanned aerial vehicle landing on maritime vessels using signal prediction of the ship motion. In *OCEANS 2018 MTS/IEEE Charleston*, pages 1–9. IEEE, 2018.

[3] Richard Adade, Abiodun Musa Aibinu, Bernard Ekumah, and Jerry Asaana. Unmanned aerial vehicle (uav) applications in coastal zone management—a review. *Environmental Monitoring and Assessment*, 193(3):1–12, 2021.

[4] Özlem Akar. Mapping land use with using rotation forest algorithm from uav images. *European Journal of Remote Sensing*, 50(1):269–279, 2017.

[5] Antonio Albanese, Vincenzo Sciancalepore, and Xavier Costa-Pérez. Sardo: An automated search-and-rescue drone-based solution for victims localization. *IEEE Transactions on Mobile Computing*, 21(9):3312–3325, 2021.

[6] Jonathan Cacace, Alberto Finzi, and Vincenzo Lippiello. Multimodal interaction with multiple co-located drones in search and rescue missions. *arXiv preprint arXiv:1605.07316*, 2016.

[7] Pei-Chun Chen, Yen-Cheng Chiang, and Pei-Yi Weng. Imaging using unmanned aerial vehicles for agriculture land use classification. *Agriculture*, 10(9):416, 2020.

[8] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[9] Ian Cox, Jeremy Howitt, and John Duncan. The use of simulation to de-risk maritime uav operations. Technical report, SYSTEM ENGINEERING AND ASSESSMENT LTD BRISTOL (UNITED KINGDOM), 2007.

[10] Rodrigo Saar De Moraes and Edison Pignaton De Freitas. Multi-uav based crowd monitoring system. *IEEE Transactions on Aerospace and Electronic Systems*, 56(2):1332–1345, 2019.

[11] Antonio-Javier Gallego, Antonio Pertusa, and Pablo Gil. Automatic ship classification from optical aerial images with convolutional neural networks. *Remote Sensing*, 10(4):511, 2018.

[12] Antonio-Javier Gallego, Antonio Pertusa, Pablo Gil, and Robert B Fisher. Detection of bodies in maritime rescue operations using unmanned aerial vehicles with multispectral cameras. *Journal of Field Robotics*, 36(4):782–796, 2019.

[13] Yunus Karaca, Mustafa Cicek, Ozgur Tatli, Aynur Sahin, Sinan Pasli, Muhammed Fatih Beser, and Suleyman Turedi. The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations. *The American journal of emergency medicine*, 36(4):583–588, 2018.

[14] Benjamin Kiefer, David Ott, and Andreas Zell. Leveraging synthetic data in object detection on unmanned aerial vehicles. *arXiv preprint arXiv:2112.12252*, 2021.

[15] Qingpeng Li, Lichao Mou, Qingjie Liu, Yunhong Wang, and Xiao Xiang Zhu. Hsf-net: Multiscale deep feature embedding for ship detection in optical remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 56(12):7147–7161, 2018.

[16] Xiaomin Lin, Nitesh Jha, Mayank Joshi, Nare Karapetyan, Yiannis Aloimonos, and Miao Yu. Oystersim: Underwater simulation for enhancing oyster reef monitoring. *arXiv preprint arXiv:2209.09395*, 2022.

[17] Xiaomin Lin, Nitin J Sanket, Nare Karapetyan, and Yiannis Aloimonos. Oysternet: Enhanced oyster detection using simulation. *arXiv preprint arXiv:2209.08176*, 2022.

[18] Eleftherios Lygouras, Nicholas Santavas, Anastasios Taitzoglou, Konstantinos Tarchanidis, Athanasios Mitropoulos, and Antonios Gasteratos. Unsupervised human detection with an embedded vision system on a fully autonomous uav for search and rescue operations. *Sensors*, 19(16):3542, 2019.

[19] Balmukund Mishra, Deepak Garg, Pratik Narang, and Vipul Mishra. Drone-surveillance for search and rescue in natural disaster. *Computer Communications*, 156:1–10, 2020.

[20] Naser Hossein Motlagh, Miloud Bagaa, and Tarik Taleb. Uav-based iot platform: A crowd surveillance use case. *IEEE Communications Magazine*, 55(2):128–134, 2017.

[21] Patrick José Pereira. bluerov ros playground, 2018.

[22] Rahul Raj, Soumyashree Kar, Rohit Nandan, and Adinarayana Jagarlapudi. Precision agriculture and unmanned aerial vehicles (uavs). In *Unmanned Aerial Vehicle: Applications in Agriculture and Environment*, pages 7–23. Springer, 2020.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[24] Nitin J Sanket, Chahat Deep Singh, Cornelia Fermüller, and Yiannis Aloimonos. Prgflow: Unified swap-aware deep global optical flow for aerial robot navigation. *Electronics Letters*, 57(16):614–617, 2021.

[25] Inc. The MathWorks. *UAV Toolbox*. Natick, Massachusetts, United State, 2019.

[26] Dimosthenis C Tsouros, Stamatia Bibi, and Panagiotis G Sarigiannidis. A review on uav-based applications for precision agriculture. *Information*, 10(11):349, 2019.

[27] Leon Amadeus Varga, Benjamin Kiefer, Martin Messmer, and Andreas Zell. Seadronessee: A maritime benchmark for detecting humans in open water. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2260–2270, 2022.

[28] Omar Velasco, João Valente, Pablo J Alhama Blanco, and Mohammed Abderrahim. An open simulation strategy for rapid control design in aerial and maritime drone teams: A comprehensive tutorial. *Drones*, 4(3):37, 2020.

[29] Zi-Hao Wang, Te Zhang, Kai-Yu Qin, and Bo Zhu. A vision-aided navigation system by ground-aerial vehicle cooperation for uav in gnss-denied environments. In *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*, pages 1–6. IEEE, 2018.

[30] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[31] SP Yeong, LM King, and SS Dol. A review on marine search and rescue operations using unmanned aerial vehicles. *International Journal of Marine and Environmental Sciences*, 9(2):396–399, 2015.

A. Appendix A: More images and tables

We perform more training with more images and we have put the result here in Appendix A. More data does not guarantee more accurate, it could cause the network to overfit to the synthetic dataset which would lower the accuracy.

Table 4. The Average Precision *(AP)* of the training data(2500 images) with various BlueROV image altitudes

Altitude	AP	AP_{50}	AP_{75}	AP_s	AP_m
10 m	45.38	95.52	25.13	45.06	76.88
20 m	44.11	93.28	24.64	44.01	50.48
30 m	34.61	91.41	4.17	34.67	38.75
40 m	17.43	76.85	0.07	17.41	31.52
50 m	9.47	56.27	0.01	9.40	31.86

Table 5. The Average Precision *(AP)* of the training data(2500 images) with different water colors at an image altitude of 10 m.

Water Color	AP	AP_{50}	AP_{75}	AP_s	AP_m
Brown	45.38	95.52	25.13	45.06	76.88
Blue	47.70	95.49	32.74	47.50	70.79
Green	45.67	95.93	25.68	45.39	74.11

Table 6. Training with only real data with various training dataset sizes. We test with the remaining real dataset

Train Size	AP	AP_{50}	AP_{75}	AP_s	AP_m
500	77.09	98.97	94.03	76.91	89.15
1000	77.56	98.88	95.44	77.41	89.81
1500	77.44	98.84	93.87	77.13	91.66