UGAN: Improving Underwater Image Quality using Generative Adversarial Networks

Cameron Fabbri[1], Md Jahidul Islam[2], and Junaed Sattar[3]

***Abstract*— Autonomous underwater vehicles (AUVs) rely on a variety of sensors – acoustic, inertial and visual – for intelligent decision making. Due to its non-intrusive, passive nature, and high information content, vision is an attractive sensing modality, particularly at shallower depths. However, factors such as light refraction and absorption, suspended particles in the water, and color distortion affect the quality of visual data, resulting in noisy and distorted images. AUVs that rely on visual sensing thus face difficult challenges, and consequently exhibit poor performance on vision-driven tasks. This paper proposes a method to improve the quality of visual underwater scenes using Generative Adversarial Networks (GANs), with the goal of improving input to vision-driven behaviors further down the autonomy pipeline. Furthermore, we show how recently proposed methods are able to generate a dataset for the purpose of such underwater image reconstruction[Reconstruction meaning what we are doing here, right? So this is a contribution too, to show how CycleGAN is used to generate the dataset, correct?]. For any visually-guided underwater robots, this improvement can result in increased safety and reliability through robust visual perception. To that effect, we present quantitative and qualitative data which demonstrates that images corrected through the proposed approach generate more visually appealing images, and also provide increased accuracy for an underwater diver tracking algorithm.**

I. Introduction

Underwater robotics has been a steadily growing subfield of autonomous field robotics, assisted by the advent of novel platforms, sensors and propulsion mechanisms. While autonomous underwater vehicles are often equipped with a variety of sensors, visual sensing is an attractive option because of its non-intrusive, passive, and energy efficient nature. The monitoring of coral reefs [1], deep ocean exploration [2], and mapping of the seabed [3] are a number of tasks where visually-guided AUVs and ROVs (Remotely Operated Vehicles) have seen widespread use. Use of these robots ensures humans are not exposed to the hazards of underwater exploration, as they no longer need to venture to the depths (which was how such tasks were carried out in the past). Despite the advantages of using vision, underwater environments pose unique challenges to visual sensing, as phenomena such as light refraction, absorption and scattering from suspended particles can greatly affect the optics of light. As an example, because red wavelengths are quickly absorbed by water, images tend to have a green or blue hue to them. As one goes deeper, this effect worsens, as more and more red hue is absorbed. This distortion is extremely non-linear in nature, and is affected by a large number of factors, such as the amount of light present (overcast versus sunny, operational depth), amount of particles in the water, time of day, and the camera being used. This may cause difficulty in tasks such as segmentation, tracking, or classification due to their indirect or direct use of color.

The authors are with the Department of Computer Science and Engineering, University of Minnesota-Twin Cities, MN 55455, USA.
{[1]fabbr013, [2]islam034, [3]junaed}@umn.edu

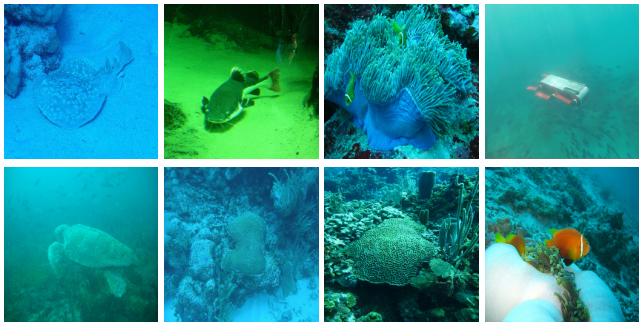


Fig. 1. Sample underwater images with natural and man-made artifacts (which in this case is our underwater robot) displaying the diversity of distortion that can occur. With the varying camera-to-object distances in the images, the distortion and loss of color varies between the different images.

As color and illumination begin to change with the depth, vision-based algorithms must need to be very generalizable in order to work within the depth ranges a robot may operate in. Because of the high cost and difficulty of acquiring a variety of underwater data to train a visual system on, as well as the high amount of noise introduced, algorithms may (and do) perform poorly in these different domains. Figure I shows the high variability in visual scenes that may occur in underwater environments. A step towards a solution to this issue is to be able to restore the images such that they appear to be above water, *i.e.*, with colors corrected and suspended particles removed from the scene. By performing a many-to-one mapping of these domains from underwater to not underwater (what the image would look like above water), algorithms that have difficulty performing across multiple forms of noise may be able to focus only one clean domain.

Deep neural networks have been shown to be powerful non-linear function approximators, especially in the field of vision [4]. Often times, these networks require large amounts of data, either labeled or paired with ground truth. For the problem of automatically colorizing grayscale images [5], paired training data is readily available due to the fact that any color image can be converted to black and white. However, underwater images distorted by either color or some other phenomenon lack ground truth, which is a major hindrance towards adopting a similar approach for correction. This paper proposes a technique based on Generative

Adversarial Networks (GANs) to improve the quality of visual underwater scenes with the goal of improving the performance of vision-driven behaviors for autonomous underwater robots. We use the recently proposed CycleGAN [6] approach, which learns to translate an image from any arbitrary domain X to another arbitrary domain Y *without* image pairs, as a way to generate a paired dataset. By letting ★ X be a set of undistorted underwater images**[need to say where we find the undistorted ones]**, and Y be a set of distorted underwater images, we can generate an image that appears to be underwater while retaining ground truth.

II. Related Work

While there have been a number of successful recent approaches towards automatic colorization [5], [7], most are focused on the task of converting grayscale images to color. Quite a few approaches use a physics-based technique to directly model light refraction [8]. Specifically for restoring color in underwater images, the work of [9] uses an energy minimization formulation using a Markov Random Field. Most similar to the work proposed in this paper is the recently proposed WaterGAN [10], which uses an adversarial approach towards generating realistic underwater images. Their generator model can be broken down into three stages: 1) Attenuation, which accounts for range-dependent attenuation of light. 2) Scattering, which models the haze effect caused by photons scattering back towards the image sensor and 3) Vignetting, which produces a shading effect on the image corners that can be caused by certain camera lenses. Differentiating from our work, they use a GAN for generating the underwater images and use strictly Euclidean loss for color correction, whereas we use a GAN for both. Furthermore, they require depth information during the training of WaterGAN, which can be often difficult to attain particularly for underwater autonomous robotic applications. Our work only requires images of objects in two separate domains (*e.g.*, underwater and terrestrial) throughout the entire process.

Recent work in generative models, specifically GANs, have shown great success in areas such as inpainting [11], style transfer [12], and image-to-image translation [13], [6]. This is primarily due to their ability to provide a more meaningful loss than simply the Euclidean distance, which has been shown to produce blurry results. In our work, we structure the problem of estimating the true appearance of underwater imagery as a paired image-to-image translation problem, using Generative Adversarial Networks (GANs) ★ as our generative model**[this needs a little bit more expansion]**. Much like the work of [13], we use image pairs from two domains as input and ground truth. **[Somewhere I** ★ **think we should mention that unpaired image to image translation is more difficult, and that cyclegan is a good way to generate a dataset in such a way to do this with image pairs.]**

III. Methodology

Underwater images distorted by color or circumstance lack ground truth, which is a necessity for previous colorization approaches. Furthermore, the distortion present in an underwater image is highly nonlinear; simple methods such as adding a hue to an image do not capture all of the dependencies. We propose to use CycleGAN as a distortion model in order to generate paired images for training. Given a domain of underwater images with no distortion, and a domain of underwater images with distortion, CycleGAN is able to perform style transfer. Given an undistorted image, CycleGAN distorts it such that it appears to have come from the domain of distorted images. These pairs are then used in our algorithm for image reconstruction.

A. Dataset Generation

Depth, lighting conditions, camera model, and physical location in the underwater environment are all factors that affect the amount of distortion an image will be subjected to. Under certain conditions, it is possible that an underwater image may have very little distortion, or none at all. We let I^C be an underwater image with no distortion, and I^D be the same image with distortion. Our goal is to learn the function $f : I^D \rightarrow I^C$. Becasue of the difficulty of collecting underwater data, more often than not only I^D or I^C exist, but never both.

To circumvent the problem of insufficient image pairs, we use CycleGAN to generate I^D from I^C, which gives us a paired dataset of images. Given two datasets X and Y, where $I^C \in X$ and $I^D \in Y$, CycleGAN learns a mapping $F : X \rightarrow Y$. Figure 2 shows paired samples generated from CycleGAN. From this paired dataset we train a generator G to learn the function $f : I^D \rightarrow I^C$. It should be noted that during our data generation process, CycleGAN simultaneously learns a mapping $G : Y \rightarrow X$, which is similar to f. In Section IV, we compare images generated by CycleGAN with images generated through our approach.

B. Adversarial Networks

In machine learning literature, Generative Adversarial Networks (GANs) [14] represent a class of generative models based on game theory in which a generator network competes against an adversary. From a classification perspective, the generator network produces instances which actively attempt to 'fool' the discriminator network. The goal is for the discriminator network to be able to distinguish between 'true' instances coming from the dataset and 'false' instances produced by the generator network. In our case, conditioned on an image I^D, the generator is trained to produce an image to try and fool the discriminator, which is trained to distinguish between distorted and non-distorted underwater images. In the original GAN formulation, our goal is to solve the minimax problem:

$$\min_G \max_D \mathbb{E}_{I^C \sim p_{train}(I^C)}[logD(I^C)] + \mathbb{E}_{I^D \sim p_{gen}(I^D)}[log(1 - D(G(I^D)))] \quad (1)$$

Fig. 2. Paired samples of ground truth and distorted images generated by CycleGAN. Top row: Ground truth. Bottom row: Generated samples.

Note for simplicity in notation, we will further omit $I^C \sim p_{train}$ and $I^D \sim p_{gen}$. In this formulation, the discriminator is hypothesized as a classifier with a sigmoid cross-entropy loss function, which in practice may lead to issues such as the vanish gradient and mode collapse. As shown by [15], as the discriminator improves, the gradient of the generator vanishes, making it difficult or impossible to train. Mode collapse occurs when the generator "collapses" onto a single point, fooling the discriminator with only one instance **[Such as when you are training a GAN to generate digits from MNIST, but it only generates a 7, when in reality you'd want it to generate a diverse amount of all the digits. Not sure if we should expand more.]**. There have been a number of recent methods which hypothesize a different loss function for the discriminator [16], [17], [18], [19]. We focus on the Wasserstein GAN (WGAN) [17] formulation, which proposes to use the Earth-Mover or *Wasserstein-1* distance W by constructing a value function using the Kantorovich-Rubinstein duality [20]. In this formulation, W is approximated given a set of k-Lipschitz functions f modeled as neural networks. To ensure f is k-Lipschitz, the weights of the discriminator are clipped to some range $[-c, c]$. In our work, we adopt the Wasserstein GAN with gradient penalty (WGAN-GP) [18], which instead of clipping network weights like in [17], ensures the Lipschitz constraint by enforcing a soft constraint on the gradient norm of the discriminator's output with respect to its input. Following [18], our new objective then becomes

$$\mathcal{L}_{WGAN}(G, D) = \mathbb{E}[D(I^C)] - \mathbb{E}[D(G(I^D))] + \lambda_{GP}\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(||\nabla_{\hat{x}}D(\hat{x})||_2 - 1)^2], \tag{2}$$

where $\mathbb{P}_{\hat{x}}$ is defined as samples along straight lines between pairs of points coming from the true data distribution and the generator distribution, and λ_{GP} is a weighing factor. In order to give G some sense of ground truth, as well as capture low level frequencies in the image, we also consider the $L1$ loss

$$\mathcal{L}_{L1} = \mathbb{E}[||I^C - G(I^D)||_1]. \tag{3}$$

Combining these, we get our final objective function for our network, which we call Underwater GAN (UGAN),

$$\mathcal{L}^*_{UGAN} = \min_G \max_D \mathcal{L}_{WGAN}(G, D) + \lambda_1 \mathcal{L}_{L1}(G). \tag{4}$$

C. Image Gradient Difference Loss

Often times generative models produce blurry images. We explore a strategy to sharpen these predictions by directly penalizing the differences of image gradient predictions in the generator, as proposed by [21]. Given a ground truth image I^C, predicted image $I^P = G(I^D)$, and $\alpha \geq 1$, the Gradient Difference Loss (GDL) is given by

$$\mathcal{L}_{GDL}(I^C, I^P) = \sum_{i,j} ||I^C_{i,j} - I^C_{i-1,j}| - |I^P_{i,j} - I^P_{i-1,j}||^\alpha + ||I^C_{i,j-1} - I^C_{i,j}| - |I^P_{i,j-1} - I^P_{i,j}||^\alpha. \tag{5}$$

In our experiments, we denote our network as UGAN-P when considering the GDL, which can be expressed as

$$\mathcal{L}^*_{UGAN\text{-}P} = \min_G \max_D \mathcal{L}_{WGAN}(G, D) + \lambda_1 \mathcal{L}_{L1}(G) + \lambda_2 \mathcal{L}_{GDL} \tag{6}$$

D. Network Architecture and Training Details

Our generator network is a fully convolutional encoder-decoder, similar to the work of [13], which is designed as a "U-Net" [22] due to the structural similarity between input and output. Encoder-decoder networks downsample (encode) the input via convolutions to a lower dimensional embedding, in which this embedding is then upsampled (decode) via transpose convolutions to reconstruct an image. The advantage of using a "U-Net" comes from explicitly preserving spatial dependencies produced by the encoder, as opposed to relying on the embedding to contain all of the information. This is done by the addition of "skip connections", which concatenate the activations produced from a convolution layer i in the encoder to the input of a transpose convolution layer $n - i + 1$ in the decoder, where n is the total number of layers in the network. Each convolutional layer in our generator uses kernel size 4×4 with stride 2. Convolutions in the encoder portion of the network are followed by batch normalization [23] and a leaky ReLU activation with slope 0.2, while transpose convolutions in the decoder are followed by a ReLU activation [24] (no batch norm in the decoder). Exempt from this is the last layer of the decoder, which uses a TanH nonlinearity to match the input distribution of $[-1, 1]$. Recent work has proposed Instance Normalization [25] to improve quality in image-to-image translation tasks, however we observed no added benefit.

Our fully convolutional discriminator is modeled after that of [26], except no batch normalization is used. This is due to the fact that WGAN-GP penalizes the norm of the discriminator's gradient with respect to each input individually, which batch normalization would invalidate. The authors of [18] recommend layer normalization [27], but we found no significant improvements. Our discriminator is modeled as

a PatchGAN [13], [28], which discriminates at the level of image patches. As opposed to a regular discriminator, which outputs a scalar value corresponding to real or fake, our PatchGAN discriminator outputs a $32 \times 32 \times 1$ feature matrix, which provides a metric for high level frequencies.

IV. Experiments

A. Datasets

We used several subsets of Imagenet [29] for training and evaluation of our methods. We also evaluate a tracking algorithm on a video of scuba divers taken from Youtube [1]. Subsets of Imagenet containing underwater images were selected for the training of CycleGAN, and manually separated into two classes based on visual inspection. We let X be the set of underwater images with no distortion, and Y be the set of underwater images with distortion. X contained 6143 images, and Y contained 1817 images. We then trained CycleGAN to learn the mapping $F : X \rightarrow Y$, such that images from X appeared to have come from Y. Finally, our image pairs for training data were generated by distorting all images in X with F. Figure 2 shows sample training pairs. When comparing with CycleGAN, we used a test set of 56 images acquired from Flickr™.

B. Evaluation

We train UGAN and UGAN-P on the image pairs generated by CycleGAN, and evaluate on the images from the test set, Y. Note that these images do not contain any ground truth, as they are original distorted images from Imagenet. Images for training and testing are of size $256 \times 256 \times 3$ and normalized between $[-1, 1]$. Figure 3 shows samples from the test set. Notably, these images contain varying amounts of noise. Both UGAN and UGAN-P are able to recover lost color information, as well as correct any color information this is present.

While many of the distorted images contain a blue or green hue over the entire image space, that is not always the case. In certain environments, it is possible that objects close to the camera are undistorted with correct colors, while the background of the image contains distortion. In these cases, we would like the network to only correct parts of the image that appear distorted. The last row in Figure 3 shows a sample of such an image. The orange of the clownfish is left unchanged while the distorted sea anemone in the background has its color corrected.

We use three metrics for a quantitative evaluation. First, we compare to CycleGAN, as it inherently learns an inverse mapping during the training of $F : X \rightarrow Y$, using the Canny edge detector [30]. This provides a color agnostic evaluation of the images in comparison to ground truth. Second, we compare local image patches to provide sharpness metrics on our images. Lastly, we show how an existing tracking algorithm for an underwater robot improves performance with generated images.

[1]https://www.youtube.com/watch?v=QmRFmhILd5o

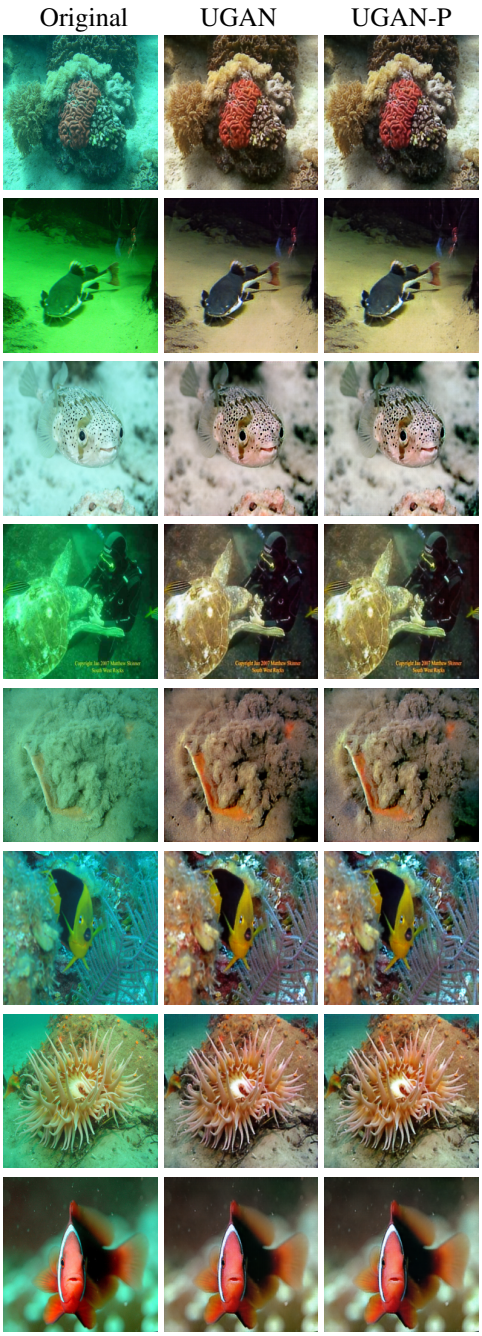


Fig. 3. Samples from our Imagenet testing set. The network can both recover color and also correct color if a small amount is present.

C. Training and Inference Performance

In all of our experiments, we use $\lambda_1 = 100$, $\lambda_{GP} = 10$, batch size of 32, and the Adam Optimizer [31] with learning rate $1e-4$. Following WGAN-GP, the discriminator is updated n times for every update of the generator, where $n = 5$. For UGAN-P, we set $\lambda_2 = 1.0$ and $\alpha = 1$. Our implementation was done using the Tensorflow library [32]. [2] All networks were trained from scratch on a GTX 1080 for 100 epochs. Inference on the GPU takes on average $0.0138s$,

[2]Code is available at `https://github.com/cameronfabbri/Underwater-Color-Correction`

which is about 72 Frames Per Second (FPS). On a CPU (Intel Core i7-5930K), inference takes on average $0.1244s$, which is about 8 FPS. We find both of these measures acceptable for underwater tasks.

D. Comparison to CycleGAN

It is important to note that during the process of learning a mapping $F : X \rightarrow Y$, CycleGAN also learns a mapping $G : Y \rightarrow X$. Here we give a comparison to our methods. We use the Canny edge detector [30] to provide a color agnostic evaluation of the images, as the original contain distorted colors and cannot be compared back to as ground truth. Due to the fact that restoring color information should not alter the overall structure of the image, we measure the distance in the image space between the edges found in the original and generated images. Figure 4 shows the original images and results from edge detection. Table 1 provides the measurements from Figure 4, as well as the average over our entire Flickr™ dataset. Both UGAN and UGAN-P are consistently closer in the image space to the original than that of CycleGAN, suggesting noise due to blur. Next, we evaluate this noise explicitly.

We explore the artifacts of content loss, as seen in Figure 5. In particular, we compare local statistics of the highlighted image patches, where each image patch is resized to 64×64. We use the GDL [21] from (5) as a sharpness measure. A lower GDL measure implies a smoother transition between pixels, as a noisy image would have large jumps in the image's gradient, leading to a ★ higher score. **[I THINK. I would double check, bottom of page 4: https://arxiv.org/pdf/1511.05440.pdf]**. As seen in Table II, the GDL is lower for both UGAN and UGAN-P. Interestingly, UGAN consistently has a lower score than UGAN-P, despite UGAN-P explicitly accounting for this metric in the objective function. Reasoning for this is left for our future work.

Another metric we can use to compare image patches are the mean and standard deviation of a patch. The standard deviation gives us a sense of blurriness because it defines how far the data deviates from the mean. A lower standard ★ deviation means that the data is closer to the mean **[not sure if we exactly need to say that because I assume the reviewers will know what it is]**. In the case of images, this would suggest a blurring effect due to the data being more clustered toward one pixel value. Table III shows the mean and standard deviations for the local image patches seen in Figure 5. Despite qualitative evaluation showing our methods are much sharper, quantitatively they show only slight improvement. Other metrics such as entropy are left as future work.

E. Diver Tracking using Frequency-Domain Detection

We investigate the frequency-domain characteristics of the restored images through a case-study of periodic motion tracking in sequence of images. Particularly, we compared the performance of Mixed Domain Periodic Motion (MDPM)- tracker [34] on a sequence of images of a diver

TABLE I

DISTANCES IN IMAGE SPACE

Row/Method	CycleGAN	**UGAN**	**UGAN-P**
A	116.45	85.71	86.15
B	114.49	97.92	101.01
C	120.84	96.53	97.57
D	129.27	108.90	110.50
Mean	111.60	94.91	96.51

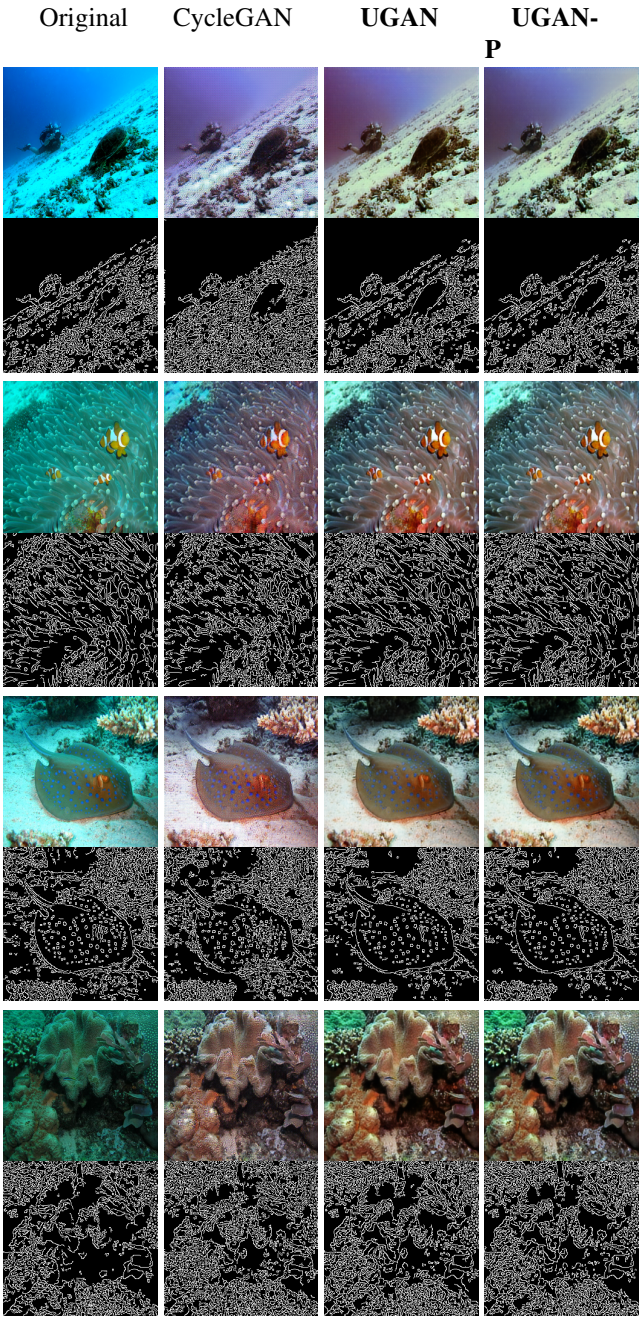


Fig. 4. Running the Canny Edge Detector on sample images. Both variants of UGAN contain less noise than CycleGAN, and are closer in the image space to the original. For each pair, the top row is the input image, and bottom row the result of the edge detector.

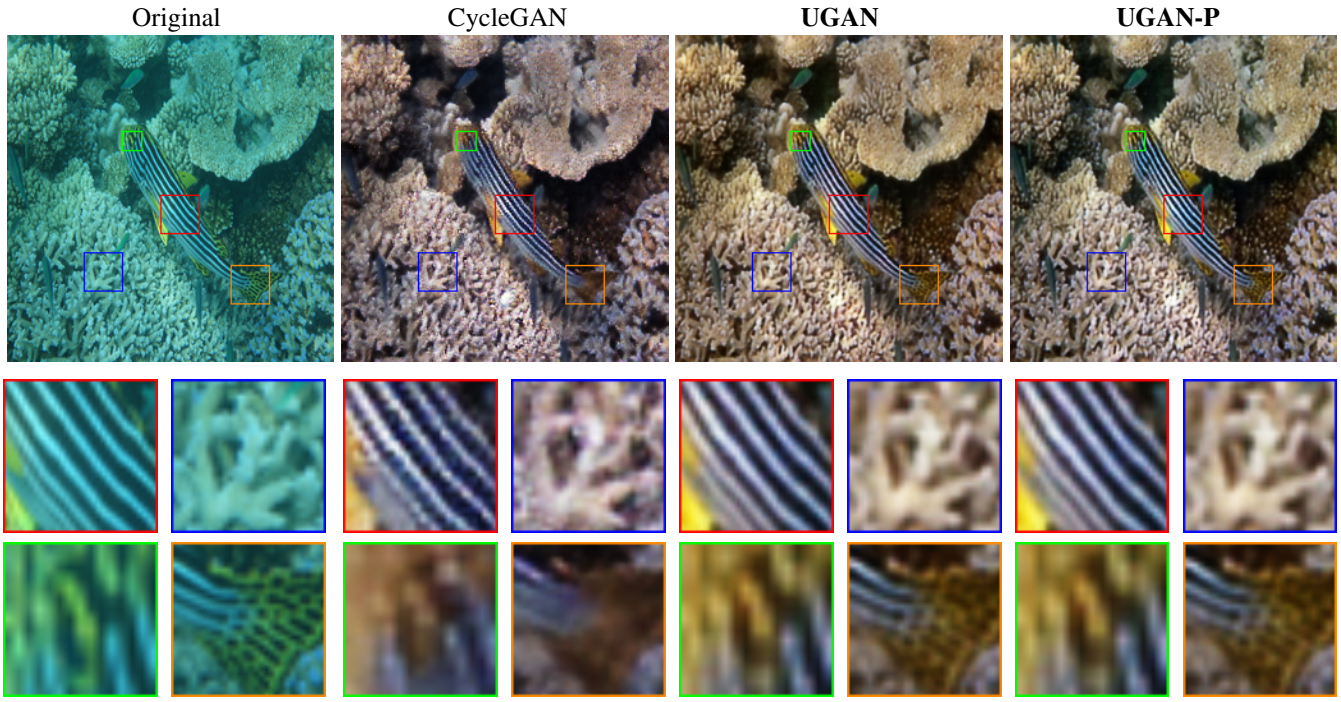


Fig. 5. Comparison of local image patches. Each patch was resized to 64×64.

TABLE II
GRADIENT DIFFERENCE LOSS METRICS

Patch/Method	CycleGAN	UGAN	UGAN-P
Red	11.53	9.39	10.93
Blue	7.52	4.83	5.50
Green	4.15	3.18	3.25
Orange	6.72	5.65	5.79

TABLE III
MEAN AND STANDARD DEVIATION METRICS

Patch/Method	Original	CycleGAN	UGAN	UGAN-P
Red	0.43 ± 0.23	0.42 ± 0.22	0.44 ± 0.23	0.45 ± 0.25
Blue	0.51 ± 0.18	0.57 ± 0.17	0.57 ± 0.17	0.57 ± 0.17
Green	0.36 ± 0.17	0.36 ± 0.14	0.37 ± 0.17	0.36 ± 0.17
Orange	0.3 ± 0.15	0.25 ± 0.12	0.26 ± 0.13	0.27 ± 0.14

swimming in arbitrary directions. MDPM tracker is designed for underwater robots to follow scuba divers by tracking distinct frequency-domain signatures (high-amplitude spectra at 1-2Hz) pertaining to human swimming. Amplitude spectra in frequency-domain correspond to the periodic intensity variations in image-space over time, which is often eroded in noisy underwater images [35].

Fig. 6 illustrates the improved performance of MDPM tracker on generated images compared to the real ones. Underwater images often fail to capture the true contrast in intensity values between foreground and background due to low visibility. The generated images seem to restore these eroded intensity variations to some extent, causing much improved positive detection for MDPM tracker.

in the image-space over time. These variations correspond to distinct signatures in the frequency-domain (high-amplitude spectra at 1-2Hz), which can be used for reliable detection.

V. CONCLUSION

Future work: Getting a better and larger dataset. I think that would help a great deal. Also data augmentation after CycleGAN, such as gaussian blurring, noise, particle effect, more lighting effects etc.

ACKNOWLEDGMENT

The authors are grateful to Oliver Hennigh for his implementation of the Gradient Difference Loss measure.

REFERENCES

[1] F. Shkurti, A. Xu, M. Meghjani, J. C. G. Higuera, Y. Girdhar, P. Giguere, B. B. Dey, J. Li, A. Kalmbach, C. Prahacs, *et al.*, "Multi-domain monitoring of marine environments using a heterogeneous robot team," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 1747–1753, IEEE, 2012.

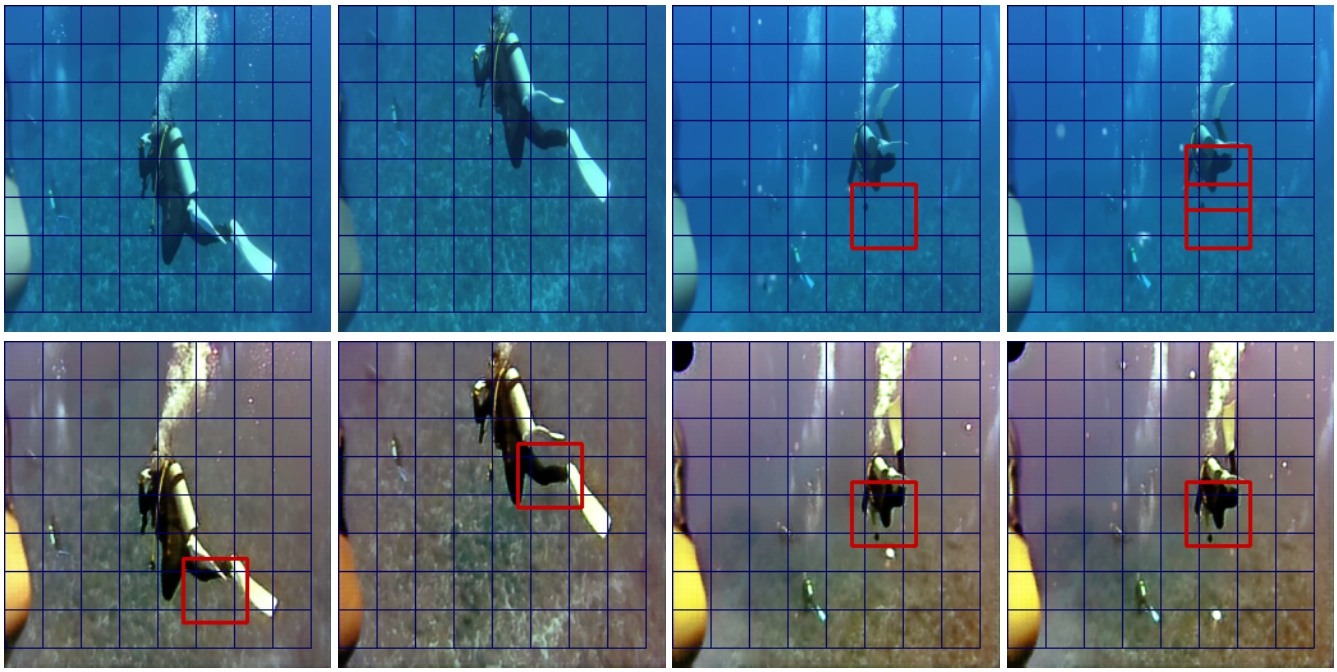
[2] L. Whitcomb, D. R. Yoerger, H. Singh, and J. Howland, "Advances in underwater robot vehicles for deep ocean exploration: Navigation, control, and survey operations," in *Robotics Research*, pp. 439–448, Springer, 2000.

[3] B. Bingham, B. Foley, H. Singh, R. Camilli, K. Delaporta, R. Eustice, A. Mallios, D. Mindell, C. Roman, and D. Sakellariou, "Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle," *Journal of Field Robotics*, vol. 27, no. 6, pp. 702–717, 2010.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[5] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European Conference on Computer Vision*, pp. 649–666, Springer, 2016.

[6] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint arXiv:1703.10593*, 2017.



	Correct detection	Wrong detection	Missed detection	Total # of frames
On real images	42	14	444	500
On generated images	147	24	329	500

Fig. 6. Performance of MDPM tracker [34] on both real (top row) and generated (second row) images; the Table compares the detection performance for both sets of images over a sequence of 500 frames.

[7] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 110, 2016.

[8] A. Jordt, *Underwater 3D reconstruction based on physical models for refraction and underwater light propagation*. Department of Computer Science, Univ. Kiel, 2014.

[9] L. A. Torres-Méndez and G. Dudek, "Color correction of underwater images for aquatic robot inspection," in *EMMCVPR*, pp. 60–73, Springer, 2005.

[10] J. Li, K. A. Skinner, R. M. Eustice, and M. Johnson-Roberson, "Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images," *arXiv preprint arXiv:1702.07392*, 2017.

[11] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544, 2016.

[12] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint arXiv:1611.07004*, 2016.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[15] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *arXiv preprint arXiv:1701.04862*, 2017.

[16] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," *arXiv preprint ArXiv:1611.04076*, 2016.

[17] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[18] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.

[19] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," *arXiv preprint arXiv:1609.03126*, 2016.

[20] C. Villani, *Optimal transport: old and new*, vol. 338. Springer Science & Business Media, 2008.

[21] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *arXiv preprint arXiv:1511.05440*, 2015.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, 2015.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 448–456, PMLR, 07–09 Jul 2015.

[24] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[25] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[26] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[27] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[28] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *European Conference on Computer Vision*, pp. 702–716, Springer, 2016.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, IEEE, 2009.

[30] J. Canny, "A computational approach to edge detection," *IEEE Trans-*

actions on pattern analysis and machine intelligence, no. 6, pp. 679–698, 1986.

[31] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[33] S. Pertuz, D. Puig, and M. A. Garcia, "Analysis of focus measure operators for shape-from-focus," *Pattern Recognition*, vol. 46, no. 5, pp. 1415–1432, 2013.

[34] M. J. Islam and J. Sattar, "Mixed-domain biological motion tracking for underwater human-robot interaction," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4457–4464, IEEE, 2017.

[35] F. Shkurti, W.-D. Chang, P. Henderson, M. J. Islam, J. C. G. Higuera, J. Li, T. Manderson, A. Xu, G. Dudek, and J. Sattar, "Underwater multi-robot convoying using visual tracking by detection," in *2017 IEEE International Conference on Intelligent Robots and Systems*, IEEE, 2017.