# Transfer Learning in Underwater Operations

Martin Skaldebø
*Dept. of Marine Technology*
*NTNU*
Trondheim, Norway
martin.b.skaldebo@ntnu.no

Albert Sans Muntadas
*Dept. of Marine Technology*
*NTNU*
Trondheim, Norway
albert.sans@ntnu.no

Ingrid Schjølberg
*Dept. of Marine Technology*
*NTNU*
Trondheim, Norway
ingrid.schjolberg@ntnu.no

***Abstract*—This paper investigates a method for reducing the reality gap that occurs when applying simulated data in training for vision-based operations in a subsea environment. The distinction in knowledge in the simulated and real domains is denoted *the reality gap*. The objective of the presented work is to adapt and test a method for transferring knowledge obtained in a simulated environment into the real environment. The main method in focus is the machine learning framework CycleGAN, mapping desired features in order to recreate environments. The overall goal is to enable a framework trained in a simulated environment to recognize the desired features when applied in the real world. The performance of the learning transfer is measured by the ability to recreate the different environments from new test data. The obtained results demonstrates that the CycleGAN framework is able to map features characteristic for an underwater environment presented with the unlabeled datasets. Evaluation metrics, such as Average precision (AP) or FCN-score can be used to further evaluate the results. Moreover, this requires labeled data, which provides additional development of the current datasets.**

***Index Terms*—Underwater robotics, transfer learning, autonomy, CycleGAN**

## I. Introduction

Today, underwater operations experience a shift towards use of more autonomous systems, where machine learning is believed to play a central role. Especially, regarding the ability to transfer knowledge between operator and system. Human brains are experts at knowledge transfer. This might be perceived as a basic trait of the human intelligence, but is in fact extremely complicated to establish as a computational ability. The main idea is to enable machines to transfer knowledge between different domains and execute different related tasks. An overall goal is be able to to train in a simulated domain and then execute the same tasks in the real world. Regarding underwater operations the latter one is of particular interest as the deep sea is less accessible, operations are costly and challenging.

Training machines in an underwater environment is extremely time consuming and error-prone due to the harsh environment. Moreover, if machines are trained exclusively in simulations the transfer of knowledge to the real world could also generate failure. This is referred to as the reality gap [1]. Generating robust techniques for transferring the knowledge between domains is therefore of immediate interest for operators in this market. There exist several published methods dealing with this problem, however only for specific domains. This paper will investigate one such method for the use in the underwater domain. One of the most promising frameworks, CycleGAN, will be tested on two different datasets considering underwater environments. The datasets include real and rendered vision based pictures of subsea panels.

### A. Motivation

The underwater robotic market size is claimed to reach USD 6.74 Billion by 2025 [2]. This corresponds to a Compound Annual Growth Rate (CAGR) of 13.5%. By comparison, Apple Inc.'s 5 year CAGR is, per April 2019, 9.2% [3]. The same report predicts that autonomous underwater vehicles (AUV) will account for USD 1.48 billion by 2025. The Norwegian Government is investing in the ocean space when designing the concept *Ocean Space Center*. The concept has a planned investment of 4.7 billion NOK [4].

Activity, interest and economic growth within the ocean space is in other words unquestionable, and with growth advancement in the technology is forthcoming. In the last years, machine learning has experienced a substantial growth in both media coverage and technological applications. One specific area is within vision-based navigation for autonomous systems. The wide interest and willingness to achieve progress that is shown today generates motivation for further investments in the field. Machine learning is believed to play a significant role in the shift towards autonomy.

### B. Background

Underwater operations today are highly dependant on human operators. Operations previously executed by human divers are now mostly transferred to remotely operated vehicles (ROVs). Moreover, the industry is today experiencing a new shift towards more autonomous operations where ROVs becomes more independent of human operators. Increasing the level of autonomy and optimize the human-robot interaction in these operations can potentially reduce costs and increase safety [5]. A higher level of autonomy leads to new requirements and increasing the autonomous complexity. Moreover, autonomous underwater vehicles (AUVs) require higher level of autonomy than ROVs. Since global navigation satellite system (GNSS) measurements are not applicable underwater, vehicle operation in this domain lacks localization measurements and are prone to accumulation of error. Today, the most common measurements and signal data arrives from acoustic sensors. Such signals are prone to data loss due to transmission losses, acoustic noise in thrusters, signal reflections on different surfaces, absorption loss and more. Feature extraction using camera vision are rarely used, but improvements within artificial neural networks (ANN), especially convolutional neural networks (CNN), shows promising results. In the presented work, systems using visual aided navigation will be investigated. This is mostly motivated by the rapid advance withing CNN and other computer vision frameworks building on CNN.

Although it is in the last few years CNN has been given recognition for its good results, it can be traced back to 1980 and Neocognitron [6]. He proposed a hierarchical multilayered neural network performing robust visual pattern recognition. Such networks can be defined as

> ”Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.” [7]

A neural network can be defined as a computer program that is inspired by the natural neural networks in the human brain [8]. Such artificial neural networks are designed to perform cognitive functions as problem solving and machine learning. Neural networks have successfully been implemented in games [9], handwriting recognition [10] and even explosive detection [11]. Neural networks provides a method for defining a system too complex to be defined by a simple model, e.g. image recognition and other systems influenced by uncertainty.

A really important parameter concerning the overall capability of the neural network is it’s architecture. The architecture concerns number of layers, number of neurons in each layer, connections between neurons etc. A nematode worm possesses only 302 neurons in total [12]. Still this presumably unintelligent worm is capable of performing complex tasks super computers today have troubles with. This is due to the complexity of the yet unknown inner mechanisms and architecture of a worm’s biological neural network. As stated before, ANNs are inspired from the biological networks in human brains. However, the extremely complex brain is still not fully understood even by scientists who have devoted large part of their professional life investigating the human brain. ANNs architecture are therefore just a mere sketch of the complex biological version. Still, through the 4th industrial revolution we are experiencing today, new methods, algorithms and frameworks emerge rapidly [13].

Machine learning applications have achieved state-of-the-art performances in multiple disciplines using ANN. Google’s *AlphaGo* has beaten the worlds best human Go player, and is arguably the strongest Go player in history [9]. *InnerEye* by Microsoft uses machine learning to develop image diagnostic tools in order to detect tumors etc. [14]. Machine learning approaches are also believed to have a dramatic impact in the fields of economics [15] in the short future. Thus, it is safe to say that machine learning will, at some extent, impact the majority of the modern generation.

### C. *Contributions*

This paper investigates a method for transfer learning in underwater domains. Existing methods have not to a large extent been tested for use in underwater domains. In the presented work, experiments are conducted for two different datasets obtained in an underwater environment. Large datasets required for machine learning applications can be expensive and difficult to acquire. Applying transfer learning methods for underwater environments can provide an alternative method for cost-effective and simple dataset generation. This paper provides a collective overview of state-of-the-art frameworks targeting transfer learning topics. Moreover, suggests solutions for reduction of the reality gap in the learning process of machines. The main contribution of the work is the application of a transfer learning framework to vision-based underwater operations.

The outline of the paper follows with Sec. II describing investigated methods involving transfer learning. Sec. III describes the experiment setup and datasets as well as conducted simulations, before the results are presented and discussed in Sec. IV. Lastly conclusions and recommendations regarding further work are presented in Sec. V.

## II. Related work

Transfer learning is a substantial problem in machine learning. A robotic arm can be trained to sort red and yellow cubes. However, such training algorithms often run into problems if the color of the cubes change to blue and green. Or, if the shape changes to triangles, or simply the lightning setting changes. Algorithms trained in a simulated environment often experience a problem when they are applied to real world data. This is referred to as the *reality gap*. Different approaches have been developed to reduce this gap between a simulated environment and the real world. A suggested solution is to train on a variation of simulated environment data. [16] developed an object detector that trained using only simulated data. The paper focused on a robotic arm that would grasp desired objects in a cluttered environment. They found it possible to train the detector to 1.5cm accuracy. The simulator they utilized consisted of randomly rendered images with variation in camera position, lighting conditions, object positions and non-realistic-textures. The objective was to perceive the real environment as just another variation. They demonstrated how their object detector could achieve high enough accuracy when tested in real life even though it only had been trained on in a simulated environment.

A breakthrough within the transfer learning topic arguably came in 2014 when Generative Adversarial Networks (GAN) was introduced [7]. The network consist of a combination of two networks, a generator and a discriminator. The generator aims to produce content, while the discriminator determines the level of authenticity of the content. They learn simultaneously and compete against each other, in what can be described as a zero-sum game. The generator produces samples x = G(z), and the discriminator attempts to determine if the samples are produced by the generator or if they come directly from the training set. The discriminator produces a probability given by D(x), indicating the probability that x is a real sample rather than a fake sample produced by the generator. The end-goal of GAN is that the discriminator will be unable to distinguish the real samples from the fake and produce a constant probability of 0.5. The discriminator will focus on learning to correctly classify samples as real or fake, while the generator will simultaneously try to generate as real looking samples as possible to fool the discriminator. This model can be highly under-constrained, but there exist several published methods and frameworks solving this.

Coupled Generative Adversarial Network (CoGAN) is a framework for learning joint distributions between individual domains [17]. The model aims to obtain a learning based on the joint distributions between domains rather than learning from corresponding images in different domains. This simplifies the requirements of the datasets, because CoGAN doesn't require corresponding images in the different domains. The framework discovers the joint distribution instead. CoGAN has been applied for color and depth images, as well as on face images with different attributes and demonstrated successfully image transformations between domains.

Based on the CoGAN framework, [18] illustrates a method for unsupervised image-to-image translation. The method learns a joint distribution between individual domains, by assuming there exists a shared-latent space. The shared-latent space assumption assumes a pair of corresponding images in different domains can be mapped in the same latent domain. The authors demonstrated image-to-image translation between two domains without any corresponding images in the training datasets. Moreover, a limitation of the presented translation is a unimodal model due to the Gaussian latent space assumption. A unimodal model means there exist only one peak, i.e. one right answer. Another limitation is possible unstable training due to the saddle point searching problem.

pix2pix uses conditional GAN (cGAN) to learn the translation between domains [19]. Since the release of the framework, a large number of different experiments has been conducted by different people. The framework shows promising results. The downside of pix2pix is the need for correlating image pairs in the source and target domain. A modified version of GAN, CycleGAN, is a method to perform image-to-image translation between domains without paired images in each domain [20]. The independence from paired images as well as wide range of domains CycleGAN has been applied to, are the main reasons why CycleGAN is the contemplated framework for this paper.

### A. CycleGAN

A thorough description of the CycleGAN framework can be found in [20]. Moreover, an overall description of the framework and how the cycle consistency is implemented in the framework is summarized here. The image-to-image translation is achieved by adding an additional generator and discriminator. The framework attempts to learn the mapping $y = G(x)$ and $x \approx F(G(x))$, where $G$ and $F$ are two different generators. CycleGAN is one of the recent most successful approaches to the image domain transformation topic. Introducing $x \approx F(G(x))$ provides an additional loss function, the cycle consistency loss, in addition to the adversarial loss. The adversarial loss is defined with

$$\begin{aligned} L_{GAN}(G, D_Y, X, Y) = &\mathbf{E}_{y\sim p_{data}} \log D_Y(\mathbf{y}) \\ &+ \mathbf{E}_{x\sim p_{data}} \log \Big(1 - D_Y\big(\mathbf{G}(\mathbf{x})\big)\Big), \end{aligned} \tag{1}$$

where $G$ is the mapping function attempting to generate images $G(x)$ similar to images in domain $Y$. $D_Y$ attempts to distinguish between the generated images, $G(x)$, and the real images $y$.

In order to implement a desired cycle consistent mapping, the cycle consistency loss is added, (2). This loss ensures that for each image, $x$ or $y$, the original image is reconstructed after the image translation cycle, i.e. $X \approx F(G(X))$ and $Y \approx G(F(Y))$, as previously mentioned.

$$\begin{aligned} L_{cyc}(G, F) = &\mathbf{E}_{x\sim p_{data}} ||F\big(G(x)\big) - x||_1 \\ &+ \mathbf{E}_{y\sim p_{data}} ||G\big(F(y)\big) - y||_1 \end{aligned} \tag{2}$$

The objective in CycleGAN will concequently be a sum of the adversiaral loss and the cycle consistency loss, represented with the final loss function

$$\begin{aligned} L_{CycleGAN}(G, F, D_x, D_y) = &L_{GAN}(G, D_Y, X, Y) \\ &+ L_{GAN}(G, D_X, Y, X) \\ &+ \lambda L_{cyc}(G, F). \end{aligned} \tag{3}$$

$\lambda$ determines the relative importance of the two objectives. Notice that the final loss function is represented with two functions for adversarial loss. This is to ensure the losses for mapping between both domains are accounted for. Considering the loss function given by (3), the objective of CycleGAN will be to solve

$$G^*, F^* = \arg\min_{G,F} \max_{D_X, D_Y} L_{CycleGAN}(G, F, D_x, D_y). \tag{4}$$

As mentioned, CycleGAN offers unpaired image-to-image translation. Regarding datasets, this provide a great advantage, because datasets can be extracted from already existing data in the industry. The framework can also provide the translation with unlabeled dataset, which means time spent on labeling each element in vast amounts of data can then be avoided.

## III. Experimental Setup

In this section the two contemplated datasets will be introduced. Parameters regarding the training and testing of the CycleGAN framework will also be presented.

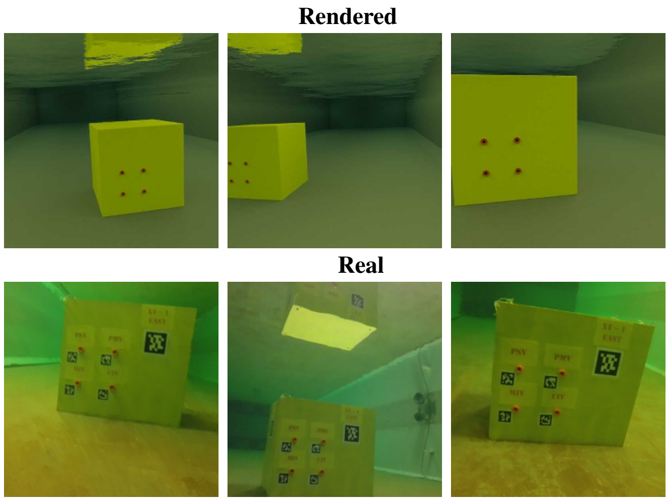


Fig. 1: Dataset 1.

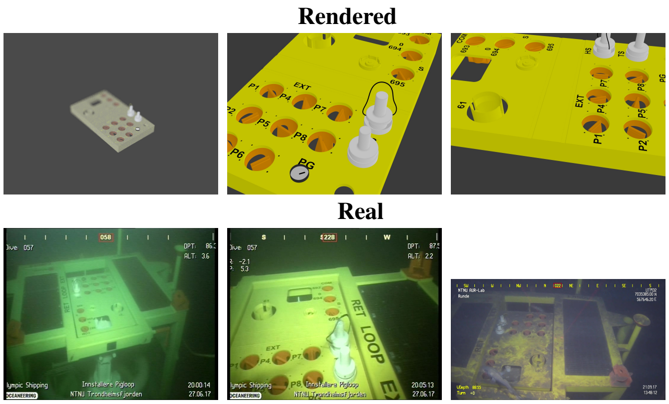


Fig. 2: Dataset 2.

### A. Dataset

The datasets that will be used for simulations are two sets containing real and rendered images of a subsea panel. Subsea panels are installed on oil and gas templates on the Norwegian Continental Shelf. The panels are accessed by ROVs for e.g. valve operations and the ROVs operators are totally dependent of good images. In case of autonomous valve operations, automatic systems based on machine learning techniques and the CycleGAN framework is one solution for image characterization. The datasets contains no corresponding images in the training sets, meaning there exist no specific image for one domain corresponding to another image in the other domain. The datasets are also unlabeled. The framework is therefore required to map the features between the domains without being told the correspondence between them. Dataset 1 contains images of a subsea panel placed in the marine cybernetics laboratory (MC-lab) at NTNU [21], as well as rendered images of the same environment. This dataset contains four different directories.

- **trainA**: Containing 4868 rendered .jpg images of the subsea panel at the bottom of the MC-lab.
- **trainB**: Containing 2947 .jpg real images of the subsea panel at the bottom of the MC-lab
- **testA**: Containing 132 rendered .jpg images of the subsea panel at the bottom of the MC-lab
- **testB**: Containing 118 .jpg real images of the subsea panel at the bottom of the MC-lab

The images are taken from a videostream filming the subsea panel at different angles, while the rendered images are rendered using the software blender. Fig. 1 represents image examples taken from the dataset.

Dataset 2 contains real and rendered images of a subsea panel placed in the the fjord outside Trondheim. These are images taken at a more realistic setting, which naturally contains more noise than the images from the laboratory. The rendered images are taken from a computer aided design (CAD) model where angles, distance and different noise patterns are altered to ensure the dataset contains variance. The images can be seen in Fig. 2. The dataset is split into 4 directories with

- **trainA**: Containing 1786 rendered .jpg images of the subsea panel.
- **trainB**: Containing 406 .jpg real images of the subsea panel at the bottom the fjord.
- **testA**: Containing 200 rendered .jpg images of the subsea panel.
- **testB**: Containing 46 .jpg real images of the subsea panel at the bottom of the fjord.

Both datasets represents an underwater environment, however at different extent. Dataset 1 is from a laboratory and the images are characterized by clear water and light conditions, not unlike a surface environment. Dataset 2 are more characterized by a typical underwater environment. The environment is dark, reflection from light source occurs and fouling are present at a representative amount of the images. Another reoccurring issue in underwater environments is marine snow which leads to occluded images. Moreover, the images in this dataset are taken from inside a fjord, which results in marine snow being almost non-existent with relative clear images due to calm water.

### B. Framework

Dataset 1 discussed above includes images with size $256x256$. When training on this network, both load size and crop size of the framework are set to $256x256$ in order to maintain the resolution of the input images. Dataset 2 includes images with different image sizes. The rendered images includes images in sizes 1080x980 and 1080x800. The real images varies between 1920x1080 and 720x576. When training on this dataset, the load size is set to $286x286$ and crop size $256x256$. The different sizes of input images are therefore coped with by loading all images into the framework with equal size before they are cropped.

The model trains for 200 epochs, which represents going through the entire dataset 200 times. The training is conducted with a constant learning rate of 0.0002 for the first 100 epochs, before decaying towards 0 for the last 100 epochs. For all simulations, $\lambda$ from equation 3 is set to $\lambda = 10$. The model is saved every 3000 iterations. In order to keep track of the progress during training, examples of the current state are generated every time the model is saved. The discriminator network architectures are $70x70$ PatchGAN networks. The regular GAN discriminator maps from a 256x256 input to a scalar output to determine real or fake. In comparison, the PatchGAN discriminator maps from a 256x256 input to a NxN network of $X$ outputs, where $X_{ij}$ signifies whether the patch $ij$ in the input image is real or fake. The architecture of the generator networks contains two 2-stride convolutions, nine residual blocks and two fractionally-strided convolutions with stride 1/2. Further, the frameworks is trained with a batch size of 3 with batch normalization. The batch normalization ensures that the loss is calculated over the batch and not for each instance. The framework is trained on a *Nvidia GeForce RTX 2080 Ti/PCIe/SSE2* graphics card.

## IV. Results and Discussion

In this section the results from the simulations will be presented. Both datasets have been tested on transfer learning between the two domains and the results vary. The varying results will also be discussed and suggested improvements and solutions will be presented.

### A. Results

The results are obtained by testing the framework on the test directories with the trained weights. A part of the obtained results are depicted in Fig. 3. The figures includes six original input images from the rendered domain and the corresponding generated output. For both datasets. Testing has been conducted between both domains, in order to see if the framework has correctly mapped the relevant features in the two domains. However, regarding the task of generating datasets for future machine learning applications, the results presented in Fig. 3 would be the most interesting. Proper generation of images in the real world domain enables generation of vast datasets from rendered images. If large and decent datasets are hard to obtain, this method can provide a more cost-effective alternative.

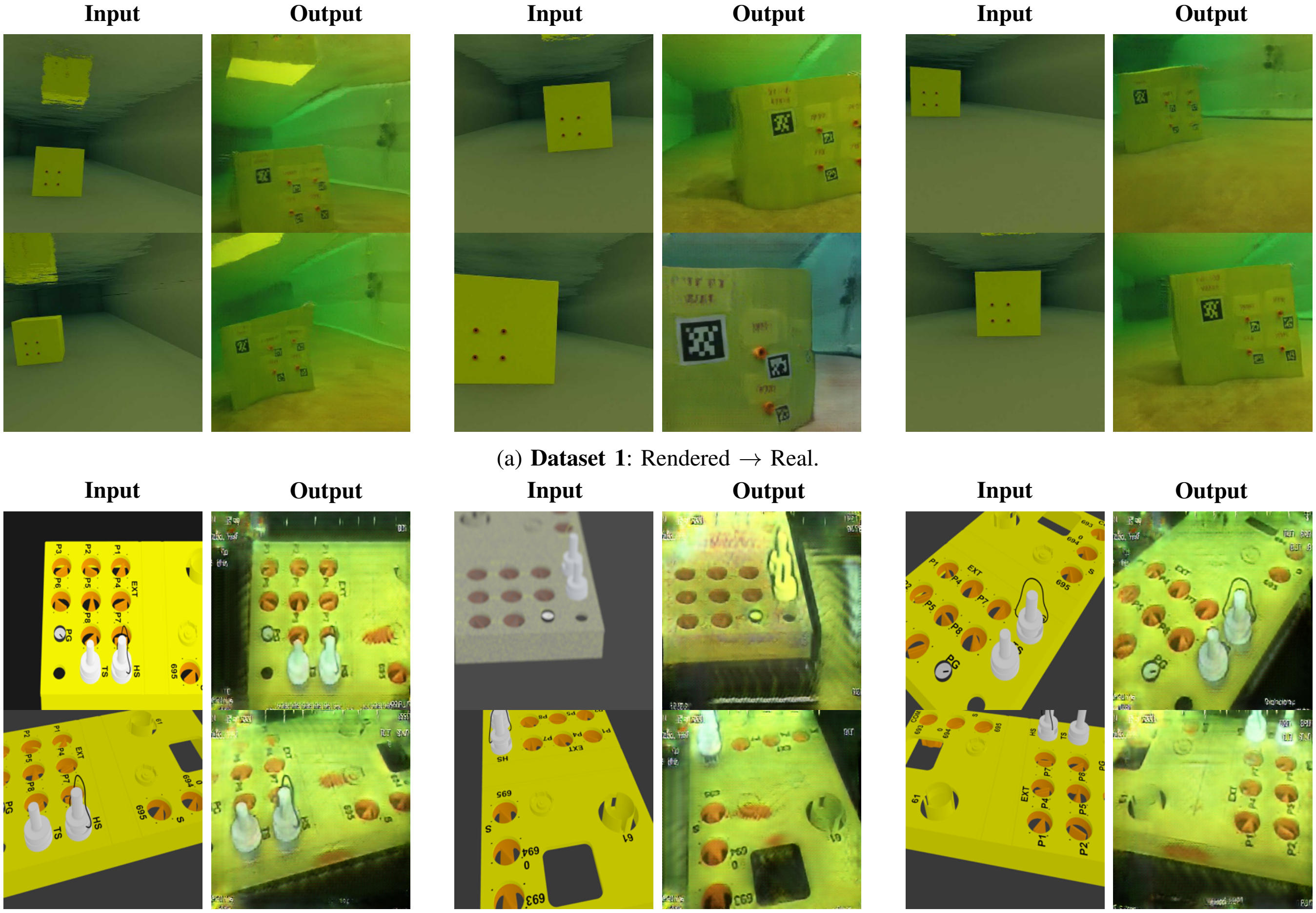
The results are most satisfying for dataset 2. It can bee seen from the figures that for dataset 2, the framework manages to transfer the subsea panel into the other domain in a good manner. The overall structure from the input is kept while the domain changes towards the real domain. Regarding dataset 1 the results are not as satisfying. The output drastically deviates from the input. The relative angle between the camera and structure as well as the spatial features are changed in the output relative to the input. The details on the subsea panel also seems to be randomly placed on the panel. This suggests that the mapping between the domains has been unsuccessful. The domains possess some different features, e.g. QR-codes are neglected in the rendered domain. The framework might encounter issues mapping features that is simply non-existent in one of the two domains.

Fig. 4 also illustrates an insufficient mapping between the domains. For both datasets. This figure depicts the results of applying the domain transfer on the real domain and transferring the input images to the rendered domain. Dataset 1 demonstrates the same issues as for the opposite domain transfer, where angles and spatial features of the input images are changed in the domain transfer. This strengthen the theory that the features of the domains has not been properly mapped due to the low level of details in the rendered images. The domain transfer for dataset 2 seems to encounter much of the same issues. The input images includes parts of the structure that are not included in the rendered domain. The additional structure circumventing the subsea panel as well as the robotic manipulator in the images are unknown to the rendered domain. Consequently, the framework have problems transferring theses features into the rendered domain where they are completely absent.

As previously stated, CycleGAN compares the original image to a reconstructed image in order to calculate the cycle consistency loss. This is depicted in Fig. 5. The figure depicts the input image fed to the framework and the output is the corresponding image in the other domain. The reconstructed image is generated by taking the output image as input and and then transferred back to the first domain. The reconstructed images represents the input very well. Notice that for the second image line, the four orange dots are almost gone in the reconstructed image. This demonstrates that the framework perceives these features as non-important. The orange dots are the only features on the rendered subsea panel with information about where the QR-codes should be placed. If these features are seen as non-important it could explain why the generated subsea panels from Fig. 3a are often flipped.

### B. Discussion

The results are promising and illustrates a decent mapping between the domains, especially for dataset 2. However, less satisfactory results were also obtained, which indicates that the feature mapping may not be as robust as desired. It should be noted that the level of details on the CAD models could be a reason. The CAD model in dataset 1 illustrates a yellow box with four orange dots. The different QR-codes that are present on the real model are neglected in the CAD model. This may confuse the framework when it attempts to map these exact features between the two domains. This might also be a reason why the generated images of the subsea panel often is flipped for this dataset. The largest QR-code are placed in the upper right corner of the real model. See Fig. 1. However, for the constructed images, is seems randomly placed in either of the upper corners. Placing QR-codes on the rendered model could help ensure that the placement is perceived as a more important feature to the framework. This dataset is obtained in the MC-Lab at NTNU, and the images are not characterized by the dark underwater environment. It is therefore believed that the issues of mapping the features between the domains, is due to the lack of details in the CAD model rather than the fact that the domains are underwater. Moreover, we do



(a) **Dataset 1**: Rendered → Real.

(b) **Dataset 2:** Rendered → Real.

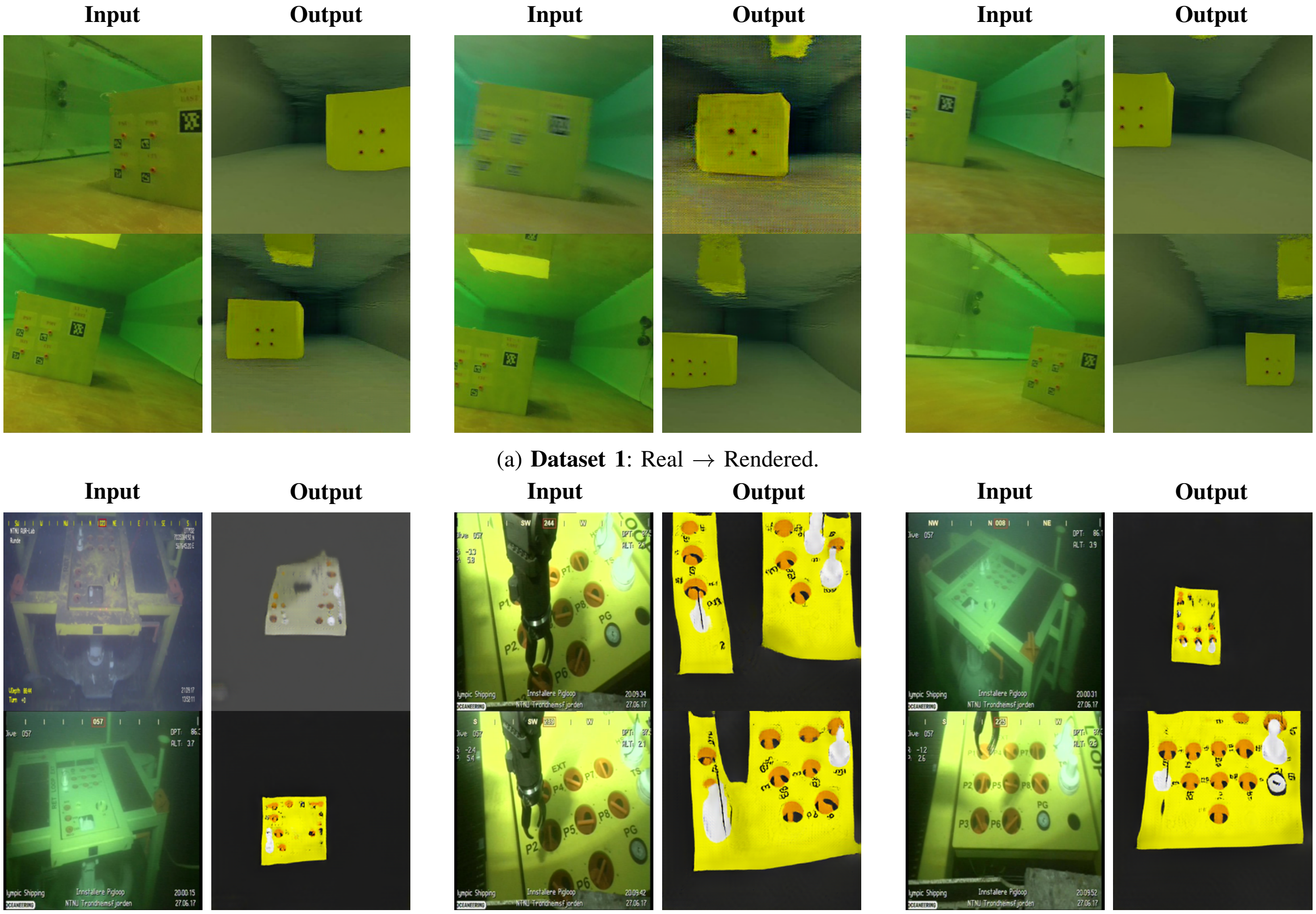
Fig. 3: Results from generating images in the real domain with rendered images as input.

not know exactly how the features are mapped in the neural network, which makes it difficult to determine how much such changes would improve the framework. It is also unknown if they would improve the results at all.

The level of details are increased for dataset 2. However, even though the CAD model is relatively detailed, it only includes the panel itself. The real images shows a panel placed on a larger subsea structure with a manipulator often occurring in the images as well. The additional subsea structure around the panel is non-existent in the CAD-model, which causes some mapping issues. Constructing images of the panel from a distance cause varying results due to this missing structure in the CAD-model. When constructing close up images of the panel on the other hand, the framework performs very well. On the close up images, the level of details are quite similar for the CAD-model and the real images. This provides good circumstances for the framework to map the features between the domains. This dataset is also much more characterized by an underwater environment. When the panel is seen from a distance it is perceived blurry, and the lighting becomes a strong feature. Due to the dark environment, a source of light is necessary in order to light up the subsea panel. Fouling on the structure, distance of the camera, occluding of camera or light source an other factors provides different reflections on the structure and provides a challenging domain to map. Still, the framework is able to represent this light reflection in a good manner.

Overall, the framework performs well. Limitations that occurs in the domain transfers are believed to arrive from different features not being present in both domains, rather than features characterized by underwater environments. Since the framework is able to comprehend with such circumstantial features, the obtained limitations should be possible to improve similarly to surface limitations.

A limitation with the dataset is the absent of marine snow in the images. The environments on the Norwegian Continental Shelf are deeper, darker and more demanding than the datasets presented in this paper. Another limitation is that the results are hard to evaluate with a metric, due to the dataset being unlabeled. A labeled dataset provides a ground truth, which the results can be compared to. Two popular evaluation metrics for the results presented in this paper are the Average precision (AP), which is often used when measuring accuracy of classifiers [22], and the FCN-score used in [19].



(a) **Dataset 1**: Real → Rendered.

(b) **Dataset 2:** Real → Rendered.

Fig. 4: Results from generating images in the rendered domain with real images as input.

## V. Conclusions and Future Work

This paper investigate methods for reducing the reality gap for vision based systems in the underwater segment. Simulations have been conducted on two different underwater datasets in order to apply existing methods at underwater environments. CycleGAN has been used as the contemplated framework. The datasets consist of rendered and real images of two different subsea panels. The framework was trained for 200 epochs on the two different datasets and the results demonstrated a partially successful mapping between the domains. Some results were satisfactory, but less satisfactory results revealed a less robust feature mapping. The framework proved to be able to map features characterized by underwater environments, such as dark images and light reflection. It is therefore believed that increasing the level of details on the CAD models could provide a solution for increasing the robustness of the feature mapping. Moreover, using labeled datasets can also provide possibilities for using evaluation metrics. These issues should be addressed in future work.

## Acknowledgments

Special thanks to MSc. student Daniel Harper for providing the CAD model utilized in dataset 2. Also thanks to MSc. student Fabian Skjølsvik for editing the images in dataset 2 and setting it up as a functioning dataset.

## References

[1] K. Bousmalis and S. Levine, "Closing the simulation-to-reality gap for deep robotic learning," Available at https://ai.googleblog.com/2017/10/closing-simulation-to-reality-gap-for.html, October 2017.

[2] ResearchAndMarkets.com, *Underwater Robotics Market Size, Share Trends Analysis Report By Type (ROV, AUV), By Application (Commercial Exploration, Defense  Security, Scientific Research), By Region, And Segment Forecasts, 2018 - 2025*. ResearchAndMarkets.com, 2018.

[3] finbox.io, "Revenue cagr (5y) for apple inc." Available at https://finbox.io/AAPL/explorer/revenue_cagr_5y, November 2018.

[4] sintef.no, "Ocean space centre får grønt lys fra kvalitetssikrer," Available at http://www.oceanspacecentre.no/wp-content/uploads/2018/04/Supplerende-analyse-Ocean-Space-Centre-forkortet.pdf, October 2018.

[5] I. Schjølberg, T. B. Gjersvik, A. A. Transeth, and I. B. Utne, "Next generation subsea inspection, maintenance and repair operations," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 434 – 439, 2016, 10th IFAC Conference on Control Applications in Marine SystemsCAMS 2016.

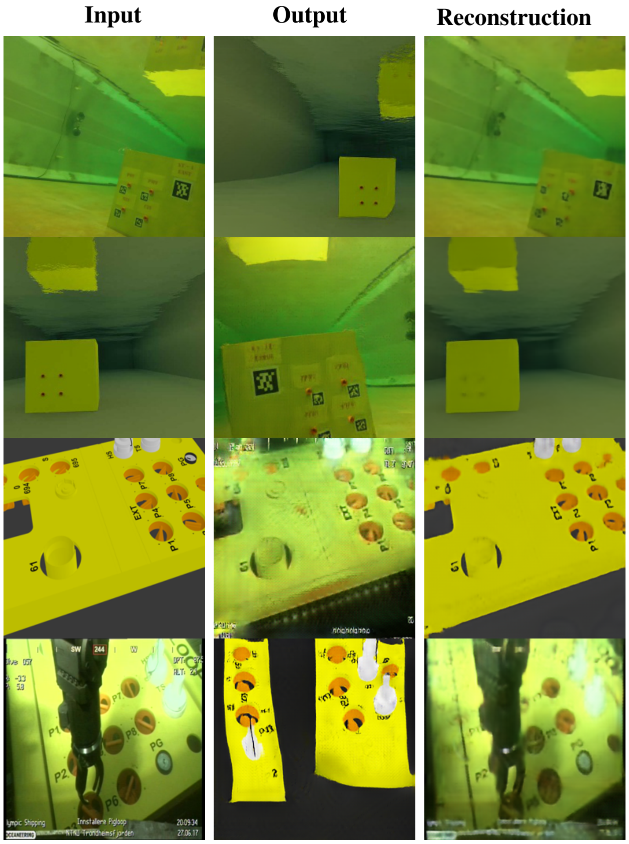


Fig. 5: Input and output with reconstructed image.

[6] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, April 1980.

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[8] V. Zwass, “Neral network,” Available at https://academic.eb.com/levels/collegiate/article/neural-network/126495, November 2018.

[9] “Alphago,” Available at https://deepmind.com/research/alphago/, November 2018.

[10] R. Almodfer, S. Xiong, M. Mudhsh, and P. Duan, “Multi-column deep neural network for offline arabic handwriting recognition,” in *Artificial Neural Networks and Machine Learning – ICANN 2017*, A. Lintas, S. Rovetta, P. F. Verschure, and A. E. Villa, Eds. Cham: Springer International Publishing, 2017, pp. 260–267.

[11] H. Wang, Y. Li, Y. Yang, S. Hu, B. Chen, and W. Gao, “Study of artificial neural network on explosive detection with pftna method,” in *IEEE Nuclear Science Symposium Conference Record, 2005*, vol. 1, Oct 2005, pp. 471–473.

[12] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner, “The structure of the nervous system of the nematode caenorhabditis elegans,” *Philosophical transactions of the Royal Society of London*, vol. 314, no. 1165, pp. 1–340, 1986.

[13] B. Marr, “What everyone must know about industry 4.0,” Available at https://www.forbes.com/sites/bernardmarr/2016/06/20/what-everyone-must-know-about-industry-4-0/2941b856795f, June 2016.

[14] “Project innereye – medical imaging ai to empower clinicians,” Available at https://www.microsoft.com/en-us/research/project/medical-image-analysis/, 2018.

[15] S. Athey, “The impact of machine learning on economics,” Available at https://www.nber.org/chapters/c14009.pdf, January 2018.

[16] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 23–30.

[17] M.-Y. Liu and O. Tuzel, “Coupled generative adversarial networks,” in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 469–477.

[18] M. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” *CoRR*, vol. abs/1703.00848, 2017.

[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *arxiv*, 2016.

[20] J. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” *CoRR*, vol. abs/1703.10593, 2017.

[21] “Marine cybernetics teaching laboratory,” Available at https://www.ntnu.edu/imt/lab/cybernetics.

[22] J. Hui, “map (mean average precision) for object detection,” Available at https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173, March 2018.