IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 7, 2018, accepted November 28, 2018, date of publication December 7, 2018,
date of current version January 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2885537

Detection and Pose Estimation for Short-Range Vision-Based Underwater Docking

SHUANG LIU[1,2,3,4], METE OZAY[4], (Member, IEEE), TAKAYUKI OKATANI[4,5], HONGLI XU[1,2,3], KAI SUN[1,2], AND YANG LIN[1,2,3]

[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016 , China
[2]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China
[3]University of Chinese Academy of Sciences, Beijing 101408, China
[4]Graduate School of Information Sciences, Tohoku University, Sendai 980-8577, Japan
[5]RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan

Corresponding authors: Shuang Liu (liushuang1@sia.cn) and Mete Ozay (mozay@vision.is.tohoku.ac.jp)

This work was supported in part by the China State Key Laboratory of Robotics Foundation under Grant 2016-Z08, in part by JST CREST under Grant JPMJCR14D1, in part by the Council for Science, Technology and Innovation (CSTI), Cross-Ministerial Strategic Innovation Promotion Program (Infrastructure Maintenance, Renovation, and Management), and in part by the ImPACT Program "Tough Robotics Challenge" of the Council for Science, Technology, and Innovation (Cabinet Office, Government of Japan).

ABSTRACT The potential of using autonomous underwater vehicles (AUVs) for underwater exploration is confined by its limited on-board battery energy and data storage capacity. This problem has been addressed using docking systems by underwater recharging and data transfer for AUVs. In this paper, we propose a vision-based framework by addressing the detection and pose estimation problems for short-range underwater docking using these systems. For robust and credible detection of docking stations, we propose a convolutional neural network called docking neural network (DoNN). For accurate pose estimation, a perspective-n-point algorithm is integrated into our framework. In order to examine our framework in underwater docking tasks, we collected a dataset of 2D images, named underwater docking images dataset (UDID), which is the first publicly available underwater docking dataset to the best of our knowledge. In the field experiments, we first evaluate the performance of DoNN on the UDID and its deformed variations. Next, we examine the pose estimation module by ground and underwater experiments. At last, we integrate our proposed vision-based framework with an ultra-short baseline acoustic sensor, to demonstrate the efficiency and accuracy of our framework by performing experiments in a lake. The experimental results show that the proposed framework is able to detect docking stations and estimate their relative pose more efficiently and successfully, compared with the state-of-the-art baseline systems.

INDEX TERMS Underwater docking, AUVs, detection, pose estimation, marine robotics.

I. INTRODUCTION

Autonomous underwater vehicles (AUVs) belong to the category of unmanned underwater vehicles (UUVs). UUVs are categorized into two main groups: i) remotely operated vehicles (ROVs), which require a user (human operator) input through a cable, and ii) autonomous underwater vehicles (AUVs), which provide stand-alone platforms without human supervision and support themselves by their on-board resources. Cables of ROVs supply adequate power and enable communication, but confine the scope of their activities. AUVs offer numerous advantages over ROVs such as a wider scope of activity, more compact size and higher efficiency, free from the limitation of a physical connection to an operator. However, its potential struggles with its finite on-board battery energy, processing and storage capacity.

Underwater docking has been popularly used due to its ability of autonomous battery recharging and data transfer, by making long-term underwater residence possible. Underwater docking systems guide AUVs into predesignated docking stations by using compatible sensors. Three types of sensors are used for an underwater docking task: i) electromagnetic [1], ii) acoustic [2], and iii) optical sensors [3]. Optical sensors outperform others in terms of good directional accuracy, low vulnerability to external detection and capacity for multiple tasks, but suffer from good propagation in an underwater environment owing to the speedy attenuation of light in the water [4]. Therefore optical sensors are usually utilized to take responsibility of the final short-distance stage precise docking, and they are combined with other sensors which are superior in propagation but inferior in accuracy [5].

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

We propose a framework by addressing a vision based underwater docking (VBUD) problem to perform docking with short-range precision at its final stage. Widely used VBUD systems consist of docking stations, cameras mounted on AUVs and docking algorithms on AUVs. Dedicated landmarks on docking stations are necessarily used by AUVs to identify docking stations. Landmarks may either be passive or active. Passive landmarks, such as patterns drawn on a board, do not need energy supply, but they are only visible within a close range. Active landmarks, such as light beacons, have high visibility by emitting energy. Active landmarks are usually preferred for its high visibility in far distance compared to passive ones [6]. In recent years, several VBUD systems using active landmarks have been proposed for underwater docking. It was verified that short-range optical terminal guidance acquisition at ranges of 10 meters - 15 meters are possible even in very turbid water in [12]. In their study, one light was used as an active landmark. Afterwards, different configurations of active landmarks are employed, such as six color lights mounted on a cone shape docking station [2], four 540nm green lights with a particular shape [3] and 3D landmarks with three green light and one red light [5].

Configurations of landmarks are determined according to mechanisms of VBUD algorithms. VBUD algorithms are employed in two phases: i) detection of docking stations, and ii) estimation of pose between AUVs and docking stations [7]. Detection of docking stations provides their location in 2D images captured by an on-board camera. Underwater docking detection algorithms fall into two general categories: binarization based methods [3], [13] and feature based methods [8]. Although various detection approaches are proposed in previous underwater docking works, none of them analyzed their detection performance and examined credibility and robustness of detection methods in detail. Pose estimation in underwater docking refers to recovering 3D relative position and orientation between docking stations and AUVs from 2D images. Both monocular and binocular cameras have been used for this purpose [3], [7], [8], [14]. Binocular methods require deployment of larger baselines with longer processing time for accurate estimation [7]. In [14], they estimated pose by using a binocular camera. In [3], they modeled the relationship between number of pixels to distance between AUVs and docking stations by using a monocular camera. In their case, exact locations of docking stations cannot be gained owing to factors like scattering. In [8], they took advantages of a PnP algorithm ($n = 3$) to estimate pose in its monocular module. In [13], they estimated pose by fitting ellipses. Their experiments were carried out in the range of less than 140cm which is not convincing in real underwater docking tasks (10 meters - 15 meters).

In this work, we consider VBUD systems equipped with active landmarks and a monocular camera. Successful underwater docking algorithms demand on several conditions. First, detection of docking stations should be credible. Observing the docking station for large number of times (e.g. 100 times), but only successfully detecting once, will be inefficient and a crucial fault for AUVs which will run out of battery. Second, detection methods implemented by VBUD algorithms should be robust to blurring, color shift, contrast shift, mirror images, non-uniform illumination and noisy luminaries observed in non-stationary underwater environments. Finally, algorithms used for pose estimation are required to be fast, accurate and robust to noise.

More precisely, underwater images are prone to blurring, color shift, reduced contrast and non-uniform illumination in different underwater conditions due primarily to the optical properties of water medium, in contrast to images captured in air [9]. Water is a strong attenuator of electromagnetic radiation. The light energy exponentially decays with respect to the propagation distance. Absorption coefficient of natural water varies depending on water quality which is an integration of effects of various constituents, such as dissolved salts, organic compounds and phytoplankton [10]. These factors dominantly cause varying degrees of color shift in underwater images. Color images are more informative than gray images for object detection tasks. Any luminary will be mapped to bright pixels no matter what color it is if gray scale images obtained by transforming color images are used. Therefore, color images are used in this work. Scattering is divided into forward scattering and backward scattering depending on scattering angle. The former results in blurring and low contrast, while the latter results in a visible bright haze in images [11]. Moreover, underwater noisy luminary may come from ambient noise light, water-surface bubble mixed with oil or other underwater light sources as shown in [3]. Noisy luminary affects performance of binary object detection methods crucially. Removing all possible noisy luminary requires employment of image pre-processing methods developed by domain experts. In addition, deformed objects are observed due to scale and rotation variance. Images of docking stations captured at different locations and viewpoints (orientation degrees) give rise to geometric deformations, addressing a challenging problem for underwater docking. Under certain conditions, mirror images of docking stations are also observed due to *total internal reflection*. According to Snell's law, there exists a *critical angle* where light is totally reflected back with zero refraction. Total internal reflection occurs at all angles smaller than critical angle, and thereby water surface serves as a mirror. Mirror images are almost identical copies of original images of docking stations. Their similar appearances confuse detection and recognition algorithms implemented in AUVs.

In order to address the aforementioned problems, we propose a vision based underwater docking framework for robust detection of underwater docking stations by leveraging our proposed convolutional neural network (CNN) architecture, and for fast, accurate and robust pose estimation by integrating a perspective-n-point algorithm used to perform final stage of underwater docking with short-range precision. Our contributions can be summarized as follows:

- We provide our underwater docking dataset which was collected in our experimental pool. To the best of our knowledge, it is the first publicly available dataset used for analysis of computer vision algorithms employed for underwater docking. We labeled bounding boxes of docking stations for each image in the dataset. The dataset can help researchers to develop underwater docking algorithms and validate their algorithms in absence of underwater docking infrastructures. In addition, a series of deformation methods are proposed in this study to generate deformed *realistic* underwater images as close as possible to real-world undersea images.
- We propose a convolutional neural network named DoNN for the detection of underwater docking stations. It has two main advantages compared to state-of-the-art networks. First, it is credible. We use the area under ROC curve (AUC) to measure its performance (the formal definition is given in Section IV-A2). If the maximum performance is achieved, then the value of AUC is 1. Our proposed DoNN achieves 0.99964 AUC [42] (experimental analyses are given in Section IV-A) for detection of underwater docking stations (please see Section IV-A4). Second, the proposed detection approach is robust to various deformations, such as blurring, color shift, contrast shift, mirror images, non-uniform illumination and noisy luminary, in various complex and dynamic underwater environments. It outperforms baseline models in terms of AUC using underwater images with blurring, color shift, contrast shift and mirror images. It achieves slightly inferior but acceptable AUC performance on underwater images with non-uniform illuminations and noisy luminaries compared to baseline models.
- We integrate a perspective-n-point algorithm termed RPnP, which is fast, accurate and robust to noise, into our framework for estimation of relative position and orientation between docking stations and AUVs. Ground experiments show that the average error of position and orientation of pose estimation module are 5.927 mm and 1.970 degree, respectively. It achieves 9.432 mm and 2.353 degree in terms of average error of position and orientation in presence of strong noise.
- In our extended field experiments, we explored underwater docking and recharge processes using the proposed framework and an experimental ship in Qiandao Lake, China. An ultra-short baseline (USBL) acoustic sensor was integrated in our VBUD framework for long-range navigation, while our VBUD framework was employed for performing final stage of underwater docking with short-range precision. Our systems succeeded in three of the consecutive four underwater docking experiments, while the proposed AUV failed in one experiment because of its physical limitation, such that the AUV could not adjust its head in a short distance to the docking station due to its large size. We provide our datasets, code and videos in publicly available repositories. Our detailed analyses of the experiments and results verify and validate our proposed systems in representative real-world environments.

II. SYSTEM OVERVIEW

This section introduces an overview of our proposed underwater docking systems. Two sets of systems are designed for our experiments. Each set of systems consists of an AUV and an underwater docking station. The first system is designed for indoor pool experiments. Its size is relatively small due to the limited size of an indoor pool. The designed AUV is of good mobility, making it very flexible for operation in the indoor pool. Since the AUV is weak in resistance to waterflow, it is not available to be used in our field experiments. In order to compensate for this weakness, we designed and used the second system with relatively large size in our field experiments. In this section, we will first introduce the AUVs used in each underwater docking system, and then show the corresponding underwater docking stations.

We first introduce the system used for performing experiments in our indoor pool. In order to perform our underwater experiments in the indoor pool, we developed an AUV research platform (SIA-9) which is a small torpedo-shaped vehicle as shown in Figure 1. Its specifications are given in Table 1. The SIA-9 is mainly equipped with a control computer, Doppler Velocity Log (DVL), Compass, Inertial Measurement Unit, Radio, GPS, battery units and motors. Besides, three additional modules were installed on the SIA-9 for our underwater docking task. Firstly, a forward-looking RGB monocular camera with frame rate 20 fps was rigidly mounted inside the head of the SIA-9. Secondly, we installed an embedded computer inside the head of the SIA-9 to process underwater images as shown in Figure 1c. The embedded computer was equipped with an Intel Core i7 3.4 Ghz processor, 8 Gb RAM and a 64-bit operating system. It was used to establish communication between the control computer and the camera through LAN and PCIe, respectively. Thirdly, the head of the SIA-9 was redesigned to fit the camera and the embedded computer as shown in Figure 1b. The software architecture is based on MOOS-IvP [15] which is used to separate overall capability into distinct modules. Each module acts as a MOOS process which can publish its own information, and subscribe others' to and from MOOSDB. MOOSDB is a module used for exchanging information between different MOOS processes, and it is responsible for maintaining the consistency of information. Implementation details of the SIA-9 were given by [16].

In our field experiments, we use another AUV research platform (SIA-3) which is also a torpedo-shaped vehicle but with a larger size. Its larger size improves its resistance to water flow, which plays a key role in stable underwater traveling for AUVs in the field experiments. We give specifications of our SIA-3 in Table 1. The SIA-3 is also equipped with a control computer, Doppler Velocity Log (DVL), Compass, Inertial Measurement Unit, Radio, GPS, battery units and

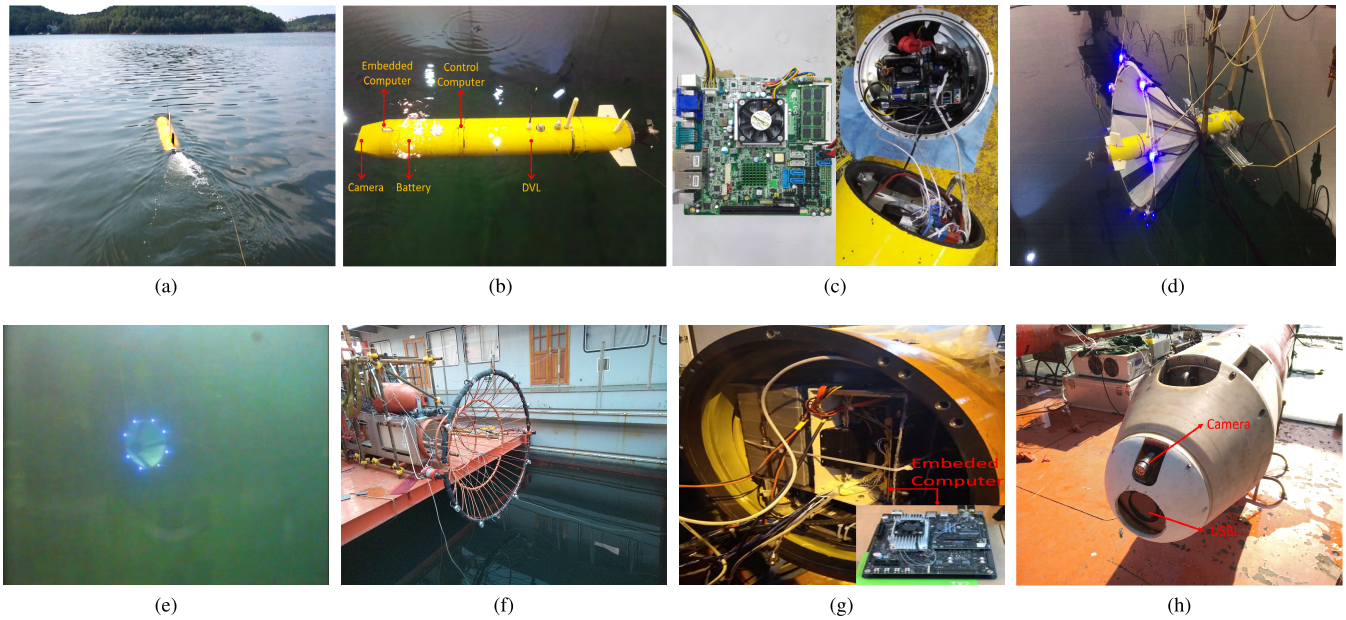


FIGURE 1. Overview of our system used in the experiments. (a) A picture of the SIA-9 located in the lake. (b) Illustration of different modules implemented in the SIA-9 after refitting for underwater docking. (c) The embedded computer used in the SIA-9 for underwater docking. The appearance of our docking station (d) with the SIA-9 on water surface, and (e) in water. (f) The docking station used in field experiments. (g) The embedded computer used in the SIA-3 for image processing. (h) The USBL and the camera used in the SIA-3.

TABLE 1. Specifications of the AUVs used in our experiments.

Item	SIA-9	SIA-3
Diameter	250 mm	384 mm
Length	1576 mm	5486 mm
Weights in air	75 kg	1500 kg
Maximum operating depth	300 m	500 m
Maximum cruising speed	3 knots	6 knots

motors, similar to the SIA-9 but with better performance. Besides, the SIA-3 is fitted with a *TrackLink*® ultra short baseline system (USBL), and a vision module as shown in Figure 1h. An USBL is an acoustic sensor mentioned in Section I. It can provide location of underwater docking stations to AUVs using long distance measurements, but it is less accurate than optical sensors using short distance measurements. We combine our proposed vision-based underwater docking system with an USBL in our field experiments. They are responsible for providing the SIA-3 precise location of the underwater docking station using short distance measurements, and rough location of the underwater docking station using long distance short distance stage, respectively. The vision module consists of a color camera and an embedded computer. We use a *NanoSeaCam*® monocular color camera whose frame rate is 20 fps in the SIA-3 as shown in Figure 1h. The camera is mounted rigidly on the head of the SIA-3. An embedded computer named Jetson TX2 is used for capturing and processing images supplied by the camera. We show it in Figure 1g. Jetson TX2 is a power-efficient embedded computing device which mainly owns a 6-core cpu, 8 Gb RAM and a 256-core GPU. Its on-board GPU can speed up processing of images captured using the camera. Jetson Tx2 communicates with the control computer through LAN.

Docking stations designed for our indoor pool experiments and field experiments share some common features. They are both cone-shaped, and equipped with eight active landmarks mounted uniformly around the rim of the docking station as shown in Figure 1d and 1f. Same blue LED lights with 460 nm wavelength were used as landmarks due to good propagation of blue light in water for both of them. The proposed design schemes of docking stations can mechanically guarantee a successful docking once an AUV enters the cone. The main difference between them is their diameter owing to different sizes of the corresponding AUVs. The diameters of the docking station designed for the indoor pool experiments and field experiments are 1200 mm and 2014 mm, respectively.

III. UNDERWATER DOCKING ALGORITHM

In this section, we provide our underwater docking algorithm consisting of two modules which are used for i) detection of underwater docking stations, and ii) estimation of pose between docking stations and AUVs. The detection module takes underwater images as input, and outputs location of underwater docking stations in 2D images. In other words, it is used to determine whether the docking station is within the field of view of AUVs, and where it is located in the captured image. Pose estimation module computes the relative position and orientation between AUVs and docking stations once the predesignated docking station is detected. AUVs conduct a line tracking task, taking current position as

the start point and the position given by the pose estimation module following the detection and pose estimation phases. Line tracking is a common procedure used to operate AUVs, and its analysis is out of scope of this work.

A. DETECTION OF UNDERWATER DOCKING STATION USING DEEP NEURAL NETWORKS

A robust and credible detection algorithm is highly desirable in practical underwater docking as mentioned in Section I. Underwater docking detection suffers from blurring, color shift, reduced contrast, mirror images, non-uniform illumination and noisy luminaries more compared to overwater detection tasks. A robust detection can guarantee detection performance in various non-stationary underwater environment while a credible one can improve the docking efficiency. The docking efficiency is crucial for AUVs which are in low battery state, and which will recharge their battery by underwater docking. However, none of them draw enough attention in the previous works, and no detection performance was reported with a detailed analysis. In this work, we propose a convolutional neural network (CNN), called Docking Neural Network (DoNN), inspired by the YOLO [17] aiming at robust and credible detection of docking stations. In this section, we first provide a brief background of CNNs, and then introduce our proposed DoNN for detection of underwater docking stations.

1) BACKGROUND OF CONVOLUTIONAL NEURAL NETWORKS

In this subsection, we introduce the basic knowledge of CNNs required to elucidate our algorithmic motivation for development of our proposed DoNN and describe its base components for multidisciplinary researchers who do not have background on DNNs.

A CNN is a deep neural network (DNN) which employs convolution layers as building blocks to model spatial patterns. CNNs draw several key ideas from Hubel and Wiesel's discovery on cat's visual cortex [18], and they have been used as one of the most successful methods used to perform robot vision tasks in the recent years [21]–[23]. A CNN is used to estimate a function f defined by $Y = f(X; \boldsymbol{\theta})$, where X and Y is input and output of the CNN. The estimated function $\hat{f}(X; \boldsymbol{\theta})$ is parameterized by the network parameters (weights) $\boldsymbol{\theta}$ of the CNN. In the training phase, the network parameters are estimated by minimizing a loss function $l(\boldsymbol{\theta})$ as

$$\boldsymbol{\theta_L} = \text{argmin } B_{\boldsymbol{\theta}} l(\boldsymbol{\theta}). \tag{1}$$

A forward propagation follows the training step for prediction of the output $\hat{Y} = f(X; \boldsymbol{\theta})$. A typical CNN layer consists of three basic operations; convolution, nonlinear activation function and pooling. In CNNs, a convolution operation is defined for an input 2D image $\boldsymbol{I}$ by

$$\tau(x, y) = (\boldsymbol{I} \otimes \boldsymbol{F})(x, y) = \sum_{m,n} \boldsymbol{I}(x + m, y + n)\boldsymbol{F}(m, n), \tag{2}$$

where $\otimes$ denotes the convolution operation, $\boldsymbol{F} \in \mathbb{R}^{m \times n}$ denotes *filters* or *kernels* and $\tau(x, y)$ is the value of *feature maps* τ computed at location (x, y). After computation of a convolution step at a location (x, y), the filter $\boldsymbol{F}$ shifts to the next location to perform the next step of the convolution, and the amount shift is controlled by *stride*.

In CNNs, convolution is usually followed by a nonlinear activation function. A commonly used nonlinear activation function is rectified linear unit (ReLU) [24]. Pooling is a form of non-linear down-sampling. It substitutes the input $\boldsymbol{I}(x, y)$ at a location (x, y) with $\boldsymbol{\Psi}(N(x, y))$, where $N(x, y)$ denotes a neighbourhood of (x, y), and $\boldsymbol{\Psi}$ is a pooling function with summary statistics such as *max pooling* [25] which utilized an operation $\max(a, b) = a, \quad if \quad a > b$. The neighbourhood $N(x, y)$ of (x, y) can be also viewed as a sliding window w_p centering at (x, y). w_p slides across every possible location (x, y) of $\boldsymbol{I}$ step by step. If max pooling is used, then it takes the largest element within w_p at each step. Stride of pooling controls the shifting units used at each pooling step. The amount of parameters and computational burden are significantly reduced through pooling. Meanwhile pooling provides invariance to small translations of inputs within the receptive field of the corresponding units (neurons) of the CNN.

CNNs are *efficient* in terms of memory requirements and statistical efficiency. The efficiency is obtained by two main features: local connectivity and parameter sharing. Local connectivity means that each neuron is connected only to a local region of its input. Parameter sharing is based on the assumption that different patches of local regions share some collective features, e.g. edges. These two features significantly reduce the amount of parameters of CNNs.

2) DOCKING NEURAL NETWORK

In this subsection, we introduce our docking neural network (DoNN) proposed for detection of underwater docking stations. Object detection is one of the most important tasks studied in robot vision. Generally speaking, CNN based detection approaches fall into two categories; region proposal based detection and proposal free detection [26]. In region proposal based detection, such as Fast-RCNN [27] and Faster-RCNN [28], first some candidate regions are proposed by using another neural network, such as region proposal network (RPN) in Faster-RCNN or selective search [29] in Fast-RCNN. Then, objects are detected in the proposals. Proposal free detection used by YOLO [17] poses detection as a regression problem. Bounding-boxes and their confidence are predicted simultaneously through one pass. DoNN is inspired by YOLO [17]. In our proposed method, we redesigned the loss function used in YOLO, in compatible with our datasets which contain one object.

DoNN consists of nine convolution layers and seven pooling layers. Its architecture is illustrated in Figure 2. Detection problem is considered as a regression problem in DoNN. In underwater docking tasks, DoNN maps underwater images to location of docking stations in images. DoNN learns feature representations of underwater docking stations from the

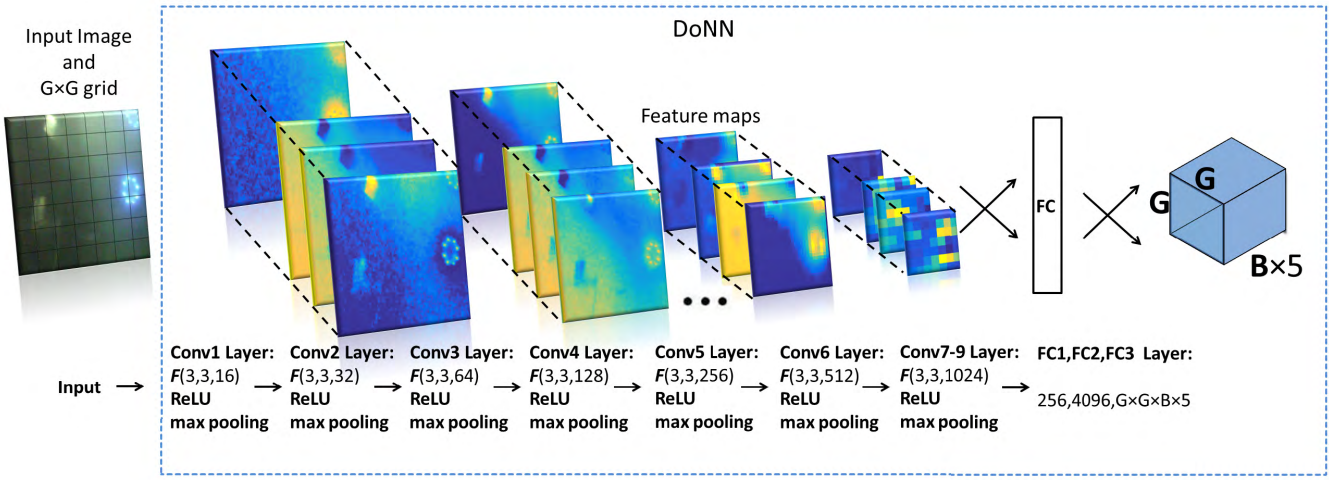


FIGURE 2. An illustration of the architecture of the proposed DoNN. DoNN consists of nine convolution layers (Conv), seven max pooling layers, and three fully connected layers (FC). *F*(width,height,depth) indicates a convolution filter with the size of width, height and depth. Sliding windows w_p of all max pooling layers have size 2 × 2. The stride is set to 1 in all convolution layers, and it is set to 2 in all max pooling layers. The number of neurons in the first fully connected layer (FC1), the second fully connected layer (FC2), and the third fully connected layer (FC3) is 256,4096 and $G \times G \times B \times 5$, respectively. DoNN takes three channel images as input, and predicts a $G \times G \times B \times 5$ tensor from which the final detection is obtained.

whole image through minimization of the loss function given in (3) in the training phase, and predicts the location of docking stations on unseen underwater images by employing learned feature representations in the inference phase.

DoNN takes the whole image as input. Input images fed are first divided into $G \times G$ grids, as illustrated in Figure 2. DoNN predicts the positions and sizes of multiple candidates for the bounding box along with their confidence score. We fix the number of the candidates and denote it by B. We further explain bounding boxes and associated confidence score as follows.

1) B bounding boxes: The b^{th} bounding-box of the i^{th} grid is denoted by $\mathcal{B}_{i,b} = (x_{i,b}, y_{i,b}, w_{i,b}, h_{i,b})$, $i \in \{1, 2, \ldots, G^2\}, b \in \{1, 2, \ldots, B\}$. The center of $\mathcal{B}_{i,b}$ is located in the i^{th} grid, and $(x_{i,b}, y_{i,b})$ is the coordinate of the center of the bounding box $\mathcal{B}_{i,b}$. It is represented by the offsets of the i^{th} grid bound. The terms $w_{i,b}$ and $h_{i,b}$ denote the width and height of the bounding-box, respectively. They are divided by the image width and height to be normalized to the range between 0 and 1.
2) B bounding-box confidence score: We denote a confidence score of the b^{th} bounding-box residing in the i^{th} grid by $S_{i,b}$.

DoNN is trained to minimize the discrepancy between predictions and manually labeled ground truth for each grid. This discrepancy is expressed by the loss function

$$l_{DoNN}(\boldsymbol{\theta}) = \lambda_{\mathcal{B}} l_{\mathcal{B}}(\boldsymbol{\theta}) + \lambda_d l_d(\boldsymbol{\theta}) + \lambda_{\bar{d}} l_{\bar{d}}(\boldsymbol{\theta}). \tag{3}$$

We compute each sub-loss function as follows;

$$l_{\mathcal{B}}(\boldsymbol{\theta}) = \sum_{i=1}^{G^2} \sum_{b=1}^{B} l_{i,b}^{dock} [(x_{i,b} - \hat{x}_{i,b})^2 + (y_{i,b} - \hat{y}_{i,b})^2]$$
$$+ \sum_{i=1}^{G^2} \sum_{b=1}^{B} l_{i,b}^{dock}$$
$$\times [(\sqrt{w_{i,b}} - \sqrt{\hat{w}_{i,b}})^2 + (\sqrt{h_{i,b}} - \sqrt{\hat{h}_{i,b}})^2], \tag{4}$$

$$l_d(\boldsymbol{\theta}) = \sum_{i=1}^{G^2} \sum_{b=1}^{B} l_{i,b}^{dock} (S_{i,b} - \hat{S}_{i,b})^2, \tag{5}$$

and

$$l_{\bar{d}}(\boldsymbol{\theta}) = \sum_{i=1}^{G^2} \sum_{b=1}^{B} l_{i,b}^{nodock} (S_{i,b} - \hat{S}_{i,b})^2, \tag{6}$$

where

- $l_{\mathcal{B}}(\boldsymbol{\theta})$ penalizes the difference between the predicted bounding-box $\hat{\mathcal{B}}_{i,b} = (\hat{x}_{i,b}, \hat{y}_{i,b}, \hat{w}_{i,b}, \hat{h}_{i,b})$ and their ground truth $\mathcal{B}_{i,b} = (x_{i,b}, y_{i,b}, w_{i,b}, h_{i,b})$ for each grid cell. Each grid cell has two mutually exclusive states: i) *containing docking stations*, and ii) *not containing docking stations*. Containing docking stations and not containing docking stations represents if the center of docking stations falls into the i^{th} grid or not, respectively. When the i^{th} grid contains docking stations, $\mathcal{B}_{i,b}$ also has two states. One is *responsible for prediction*, and the other is *not responsible for prediction*. $\mathcal{B}_{i,b}$ is *responsible for prediction* when it has the largest IoU [30] with the ground truth bounding-box among B bounding-boxes predicted in the i^{th} grid. $l_{i,b}^{dock}$ is equal to 1 if a) the i^{th} grid contains docking stations, and b) $\mathcal{B}_{i,b}$ is responsible for prediction. Otherwise it is 0.
- $l_d(\boldsymbol{\theta})$ penalizes confidence score loss for the grids containing docking stations. $S_{i,b}$ and $\hat{S}_{i,b}$ is the ground truth confidence score and predicted confidence score for the $\mathcal{B}_{i,b}$, respectively. $l_{i,b}^{dock}$ is a indicator function which

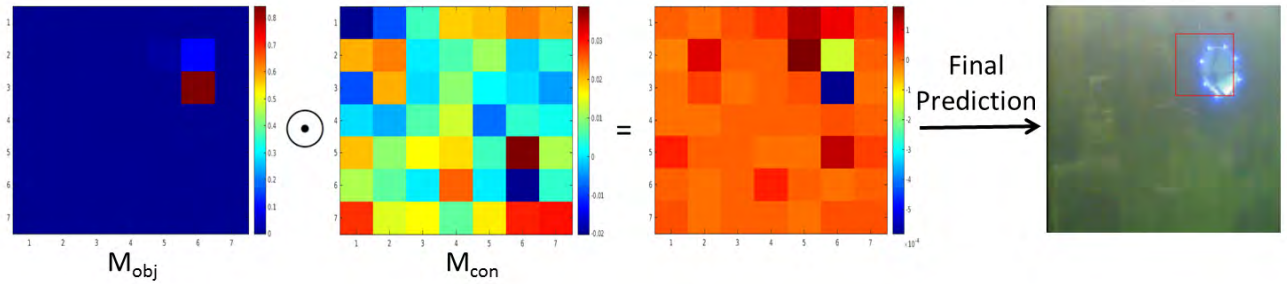


FIGURE 3. A visualization of $P_r^i(Objectness)$ and $P_r^i(Class_i|Objectness)$ computed using YOLO. We denote $P_r^i(Objectness)$ and $P_r^i(Class_i|Objectness)$ computed for all grids by $M_{obj} = \{P_r^i(Objectness)|i = 1, 2, \ldots, G \times G\}$ and $M_{con} = \{P_r^i(Class_i|Objectness)|i = 1, 2, \ldots, G \times G\}$, respectively. The left two figures show M_{obj} and M_{con}, respectively. We use $\odot$ to indicate pixel-wise multiplication of M_{obj} and M_{con}. The red bounding-box depicted in the last figure indicates the final prediction provided by YOLO. M_{obj} predicts the location of the docking station correctly, however the final prediction is corrupted after multiplication with M_{con}.

is equal to 1 if the i^{th} grid contains docking stations, otherwise it is 0.

- $l_{\bar{d}}(\boldsymbol{\theta})$ penalizes confidence score loss for the grids that do not contain docking stations. $l_{i,b}^{nodock}$ is an indicator function indicating appearance of docking stations in the i^{th} grid. $l_{i,b}^{nodock}$ is equal to 1 if the i^{th} grid does not contain docking stations.

The parameters $\lambda_{\mathcal{B}}$, λ_d and $\lambda_{\bar{d}}$ are used to control the contribution of different parts of the loss function (3). We experimentally analyze how the configuration of these two parameters affect detection performance in detail in Section IV-A3.

DoNN receives an unseen underwater image as input, and outputs a $G \times G \times B \times 5$ tensor for inference by predicting B bounding-boxes (parameterized by $\mathcal{B}_{i,b}$ using 4 elements), and B confidence scores (parameterized by $S_{i,b}$ using 1 element) for each grid. The final prediction is computed by

$$\mathcal{B}_{pred} = \mathcal{B}_{\hat{i},\hat{b}}, \quad \text{where } \hat{i}, \hat{b} = \underset{i,b}{\operatorname{argmax}}\, \hat{S}_{i,b}, \tag{7}$$

$\hat{S}_{i,b}$ is the predicted confidence score used in (5), and $\mathcal{B}_{pred}$ is the final predicted bounding-box of the docking station.

The major difference between DoNN and YOLO is the loss function. In addition to the loss function (3), (5) and (6), another loss function called *class loss* is used in YOLO. The process of learning in YOLO can be viewed as learning an *objectness* probability and a conditional probability of $Class_c$ for each grid. They are formulated by $P_r^i(Objectness)$ and $P_r^i(Class_c|Objectness)$ for the i^{th} grid, respectively. Then $P_r^i(Objectness) \cdot P_r^i(Class_c|Objectness)$ is used to predict the final class score for each grid. However, in our case, $P_r^i(Class_c|Objectness)$ introduces instability as illustrated in Figure 3, due to observation of one target in our task and our relative small datasets. To remedy this problem, we redesign the loss function used in YOLO by estimating only $P_r(dock)$ instead of an objectness probability and a conditional probability.

B. POSE ESTIMATION

In Section III-A, we explained how to compute 2D locations $\mathcal{B}_{pred}$ of docking stations in 2D images using the proposed DoNN. In this section, we provide a method which is used to recover the relative 3D position and orientation between docking stations and AUVs from the 2D image patch determined by the estimated 2D location. The 3D position is represented by $X = (x, y, z) \in \mathbb{R}^3$, and the orientation is represented by *Euler angles* $(yaw, pitch, roll)$, as illustrated in Figure 4. The relative 3D position and orientation between docking stations and AUVs is called *pose*, collectively. The process of pose recovery is called *pose estimation*. Pose estimation requires recovering the pose from the 2D image patch defined by $\mathcal{B}_{pred}$ described in (7).

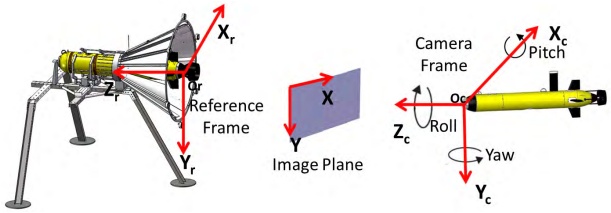


FIGURE 4. Coordinate frames used in underwater docking. Left: A reference frame. The origin of the reference frame is located at the centroid of the circle formed by landmarks. Middle: An image plane, and an image coordinate frame. Right: A camera frame, and illustrated Euler angles.

Pose estimation is performed if docking stations can be fully observed. We categorize observed docking stations into full observation, and partial observation of docking stations. A full observation of docking stations is a case where all eight landmarks can be observed. Otherwise, it is called a partial observation. In order to recognize full observations, we segment the image patch determined by $\mathcal{B}_{pred}$ using an adaptive segmentation method proposed by reference [31]. Since other interference, such as noisy luminaries and mirror images, has been addressed during detection, it is needless to worry about the sensitivity of segmentation. After the segmentation, several connected components are available. If the

number of connected components is less than eight, then the observation is a partial observation of docking stations. The number of connected components may be greater than eight. which is observed if one light is segmented into more than one connected component. In this case, we use the k-means algorithm [32] to cluster adjacent components and obtain centroids of eight landmarks. At the final step, the centroids are used for pose estimation.

If two coordinate frames are attached to the docking station and the AUV, separately, then we model the geometric relationship (the translation and rotation) between two coordinate frames for pose estimation. In pose estimation, three coordinate frames are used as illustrated in Figure 4: i) image coordinate frame, ii) camera coordinate frame and iii) reference coordinate frame. The image coordinate frame is a 2D coordinate system in pixels which is established on the image plane. The camera coordinate and reference coordinate frames are 3D coordinate systems in millimeter. The origin of the camera coordinate resides on the optical center of the camera. We attach a reference coordinate on the center of the circle formed by eight lights. Calculating the pose between AUVs and docking stations is equivalent to determination of the transformation between the camera coordinate frame and reference coordinate frame, since the camera is fixed on the head of AUVs rigidly, and only rigid-body motion is considered.

Next, we explain determination of a transformation between camera and reference coordinate frames. Considering a pinhole camera, the transformation between image coordinate and camera coordinate is computed by

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KX_c = \begin{bmatrix} k_x & k_\theta & u_0 & 0 \\ 0 & k_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}, \tag{8}$$

where $(u_0, v_0) \in \mathbb{R}^3$ is the principal point measured in pixels, $k_\theta \in \mathbb{R}$ is the skew coefficient, $(u, v) \in \mathbb{R}^3$ is the coordinate in the image frame, $(x_c, y_c, z_c) \in \mathbb{R}^3$ is the coordinate in the camera frame, $k_x \in \mathbb{R}$ and $k_y \in \mathbb{R}$ denote the scaling factor converting space metrics to pixel units, and $K \in \mathbb{R}^{3\times4}$ is the *intrinsic matrix* of inherent parameters of a camera. It describes the transformation between image frames and camera frames, and can be obtained by camera calibration. The relationship between the camera frame and reference frame can be described by the extrinsic matrix $E \in \mathbb{R}^{4\times4}$ as follows:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = EX_r = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{bmatrix} \tag{9}$$

where $(x_r, y_r, z_r) \in \mathbb{R}^3$ is the coordinate of the reference coordinate frame, $R \in \mathbb{R}^{3\times3}$ is a rotation matrix with constraints $R^TR = RR^T = I$ and $\det(R) = +1$. Translation matrix $T \in \mathbb{R}^3$ is computed with respect to the coordinate of the origin point O_r of the reference frame with respect to the camera frame. O_r is the target point used for line tracking mentioned in Section III.

Computation of R and T using n corresponding points between 2D images points and 3D coordinates is addressed as perspective-n-point (PnP) problems. PnP was first coined by [33], for computation of R and T given a calibrated camera, a set of correspondences between 3D reference points and their 2D images points. At least four correspondences are required to assure computation of a unique solution for R and T. Usually two types of methods are used to solve PnP problems: analytical and iterative methods. A popularly used analytical method is Direct Linear Transformation (DLT) [34]. DLT calculates R and T by solving 11 entries in linear equations derived by (8) and (9) from at least six corresponding points. DLT is computationally efficient but suffer from instability in the presence of noise. Iterative methods address PnP problems by minimization of an error criterion, such as re-projection error proposed by [35], and collinearity error proposed by [36]. Iterative methods are less sensitive to noise, and they are more accurate but computationally expensive.

In our framework, RPnP [37] is employed to estimate the pose between AUVs and docking stations. In Figure 5, P_i and p_i denote the i^{th} 3D point and its corresponding image point, respectively. Given n correspondences $P_i \leftrightarrow p_i$, RPnP first splits a set of n reference points into $(n-2)$ subsets, each of which contains 3 points. Each subset is illustrated in Figure 5. According to the law of cosines, the following constraints are satisfied for each subset;

$$\begin{aligned} d_1^2 + d_2^2 - 2d_1d_2\cos\gamma &= d_{12}^2, \\ d_2^2 + d_3^2 - 2d_2d_3\cos\alpha &= d_{23}^2, \\ d_1^2 + d_3^2 - 2d_1d_3\cos\beta &= d_{13}^2. \end{aligned} \tag{10}$$

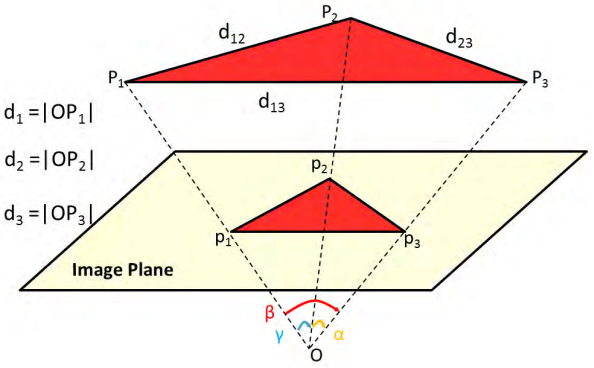


FIGURE 5. Illustration of the RPnP algorithm. P_1,P_2 and P_3 are three 3D points. p_1,p_2 and p_3 are the corresponding 2D image points located on the image plane. Point O is the optical center of the camera.

Then, (10) is converted into a fourth order polynomial for the j^{th} subset by

$$h_j(x) = a_jx^4 + b_jx^3 + c_jx^2 + d_jx + e_j = 0, \quad \forall j = 1, 2, \ldots, n-2. \tag{11}$$

Since $(n-2)$ subsets are available, we use $(n-2)$ polynomial equations (III-B), which form a system of nonlinear equations. Rather than solving this nonlinear equation system, RPnP analyzes the local minima of the seventh order polynomial cost function H, which is defined by $H = \sum_{j=1}^{n-2} h_j^2(x)$ that has at most 4 minima. The final solution with the least reprojection residual is selected from these minima as the pose estimation result.

The RPnP used in our work has the following properties:

- RPnP is accurate and highly efficient. It is a non-iterative solution and achieves as accurate solutions as iterative methods provide with less computational cost. The accuracy of RPnP is 1.5 degree median rotation error, and 0.5% median translation error for $n = 8$ co-planar points with Gaussian noise $\mathcal{N}(\mu = 0, \sigma^2 = 9)$ as reported by [37] in simulations. RPnP consumes less than 1 ms if $n = 8$.
- RPnP is stable in the co-planar case. Many PnP solutions [38], [39] suffer from pose ambiguity which results in highly unstable results. Pose ambiguity refers to the fact that orientation cannot be determined uniquely [40]. Allowing a co-planar arrangement of landmarks can reduce complexity for designing docking stations. Reference [37] showed that the mean error of rotation and translation converges when number of correspondences is larger than eight. Therefore, eight landmarks are designed in our docking station.
- The computational complexity of RPnP is $O(n)$. Its computational time grows linearly with the number of correspondences. It offers flexibility for increase and decrease of the number of landmarks according to practical requirements. Therefore, we can re-configure the amount of landmarks without substantial increase of the computational cost.

IV. EXPERIMENTAL ANALYSES

In the experimental analyses, we first analyze robustness and credibility of the proposed DoNN using our datasets. Then, we analyze the accuracy and efficiency of the pose estimation method by ground and underwater docking experiments. Finally, we give results of our field experiments in which we incorporate the aforementioned detection and pose estimation methods.

Code of the proposed algorithms used in the experiments are available in http://vision.is.tohoku.ac.jp/~liushuang/a-vision-based-underwater-docking-system/code, and the UDID dataset is available in http://vision.is.tohoku.ac.jp/~liushuang/a-vision-based-underwater-docking-system/dataset.

A. ANALYSIS OF DETECTION PERFORMANCE

In this section, we first introduce our proposed dataset UDID, and performance measures used to evaluate detection performance. Then, we analyze convergence properties of DoNN for the proposed loss functions. Finally, we compare and analyze detection performance of DoNN, and state-of-the-art YOLO and FasterRCNN methods using the UDID and its deformed variations.

Our major experimental results and observations are summarized as follows:

- Section IV-A3 (Analysis of hyperparameters of loss functions of DoNN); AUC performance of DoNN does not benefit a lot from a large number of scored bounding-boxes that are predicted by the DoNN for each grid. DoNN achieves the best AUC performance using a moderate number of grids into which images are partitioned, instead of using very fine grained grids. DoNN provides better performance when its parameters penalize difference between predicted bounding-boxes more than confidence scores.
- Section IV-A4-IV-A11; DoNN is quite robust to deformation of images observed in real-world underwater environments. This is attributed to its success for learning of feature representations and estimation of distributions of spatial structural patterns. In conclusion, DoNN outperforms YOLO and FasterRCNN overall in terms of credibility and robustness.
- Section IV-A4 (Comparison of performance of detection algorithms using the original test dataset D_{lv}); Experiments on the original test dataset show that the AUC of DoNN is 0.99964 which is better than FasterRCNN and YOLO.
- Section IV-A5 (Comparison of performance of detection algorithms using blurred images); Experiments show that DoNN achieves the best overall performance on a blurred dataset among the three models. DoNN learns better feature representations of underwater docking stations compared to FasterRCNN.
- Section IV-A6 (Comparison of performance of detection algorithms under color shift); Experiments performed on a color shift dataset show that DoNN is more robust and credible than FasterRCNN for detection of underwater docking images deformed by color shift. Specifically, DoNN outperforms FasterRCNN in experiments on underwater docking images deformed using all degrees of hue shift and saturation shift in our experiments.
- Section IV-A7 (Comparison of performance of detection algorithms under contrast shift); DoNN outperforms significantly in experiments on images with high contrast. It lags behind Faster-RCNN by a tiny margin using images with low contrast.
- Section IV-A8 (Comparison of performance of detection algorithms for mirror images); Experiments conducted on the proposed mirror images dataset show that DoNN performs well in the detection of underwater docking images with the mirror, and slightly better than Faster-RCNN. DoNN not only learns feature representations of appearance of underwater docking stations, but also distributions of relative spatial locations between real and mirror images of docking stations.

- Section IV-A9 (Comparison of performance of detection algorithms under non-uniform illumination); Non-uniform illumination is quite common in underwater imaging due to the location of the light source, such as the sun, and properties of water. Robust deteciton of underwater docking station under non-uniform illumination is of greate importance to practical applications in the field. Our experiments show that DoNN is affected very little from non-uniform illumination. AUC of DoNN conducted on the proposed non-uniform dataset is 0.99824 while the AUC of DoNN conducted on the dataset without non-uniform illumination is 0.99964.
- Section IV-A10 Experiments performed on the underwater docking dataset with noisy luminaries indicate that the negative effect of noisy luminaries on DoNN is very tiny although noisy luminaries are as bright as landmarks. AUC of DoNN only decreases by 0.001 compared to AUC in the dataset without noisy luminaries.

to complete data collectiontcolor

1) OUR PROPOSED UDID DATASET

In order to evaluate the performance of DoNN, we first set up an underwater docking images dataset (UDID), which is collected in our experimental pool. It took one month to complete data collection tasks and establish the UDID. The experimental pool is 15 m long, 10 m wide and 9 m deep with the docking station fixed at 2 m deep underwater. It comprises a training set D_{ltr} and a test set D_{lv} as illustrated in Table 2. We call images containing docking stations *foreground*, and images not containing docking stations *background*.

TABLE 2. Our proposed dataset UDID. The training subset D_{ltr} contains 8252 foreground images and no background images. The test subset D_{lv} consists of 1128 foreground images and 1114 background images.

	Foreground Images	Background Images	Total Images
D_{ltr}	8252	0	8252
D_{lv}	1128	1114	2242

As mentioned in Section I, real-world underwater images suffer from (1) blurring, (2) color shift, (3) reduced contrast, (4) mirror images, (5) non-uniform illumination and (6) noisy luminaries [3], [11]. Therefore, we also deform the test set D_{lv} using various deformation methods to assess the performance of DoNN in simulated dynamic underwater environments. All images are resized to 448 × 448 for training and testing of DoNN, and the images are resized to the original size along with detected bounding-box for pose estimation.

As mentioned in Section III-A2, there are two types of CNN based detection methods. One is region proposal based detection and the other is proposal free detection. In the experimental analyses, we compare our proposed DoNN with Faster-RCNN and YOLO, which are the sate-of-art region proposal based and proposal free detection methods, respectively. For a comparison, we used the same architecture for design of YOLO and DoNN before employment of the fully connective layers. A Faster-RCNN which employs a ZF network [41] is employed for comparison. Fully connective layers of the Faster-RCNN are updated for our two-class classification for end-to-end training using D_{ltr}.

2) PERFORMANCE MEASURES USED FOR EVALUATION OF DETECTION ALGORITHMS

Detection performance is evaluated using receiver operating characteristic (ROC) curve and its area under curve (AUC) [42]. ROC is a plot of true positive rate (TPR) against false positive rate (FPR) computed at various threshold settings. TPR and FPR are defined by

$$TPR = \frac{TP}{TP + FN} \quad and \; FPR = \frac{FP}{FP + TN}, \tag{12}$$

where TP, FP, TN, FN is the number of true positive, false positive, true negative and false negative samples, respectively. TP, FP, TN and FN are illustrated by a confusion matrix in Table 3. If a prediction and a true value of a sample is docking station, then the prediction result is evaluated as a true positive (TP). If both the prediction and the true value are non-docking stations, then the prediction is evaluated as a true negative (TN). If the predicted value is a docking station while the true value is non-docking station, then the prediction is evaluated as false positive (FP). False positive means that a docking station is detected when a docking station is not actually there. If the predicted value is non-docking station while the true value is a docking station, then the prediction is false negative (FN). We label the true value of bounding-boxes whose IoU with the ground truth bounding-box exceed 50% as docking stations as suggested by [30]. Otherwise, they are labeled as non-docking stations.

TABLE 3. Confusion matrix of predicted and true values.

		True Values: Docking Stations	True Values: Non-docking Stations
Predicted Values	Docking Stations	TP	FP
	Non-docking stations	FN	TN

The area under the ROC curve is called AUC for short, which can give an insight into the general performance of detection algorithms. AUC is equal to 1 if the maximum performance is achieved. In the following sections, we analyze the performance of DoNN, YOLO and Faster-RCNN in aforementioned test set D_{lv} and its various deformed versions.

3) ANALYSIS OF HYPERPARAMETERS OF LOSS FUNCTIONS OF DONN

Our proposed DoNN was trained using the loss function given in (3). The loss function contains three parts: i) $l_{\mathcal{B}}(\boldsymbol{\theta})$, ii) $l_d(\boldsymbol{\theta})$ and iii) $l_{\vec{d}}(\boldsymbol{\theta})$. Five hyperparameters $\lambda_{\mathcal{B}}$, λ_d, $\lambda_{\vec{d}}$, B and G are used during the training phase.

TABLE 4. AUC performance of DoNN obtained using different λ_d, $\lambda_{\mathcal{B}}$ and $\lambda_{\bar{d}}$ on D_{lv}. B and G are set to 2 and 7 in these experiments, respectively. DoNN achieves the best AUC performance for $\lambda_d = 0.5$, $\lambda_{\bar{d}} = 0.1$, $\lambda_{\mathcal{B}} = 3$. (a) Results for $\lambda_d = 0.1$. (b) Results for $\lambda_d = 0.5$. (c) Results for $\lambda_d = 1$. (d) Results for $\lambda_d = 3$. (e) Results for $\lambda_d = 5$.

(a)

AUC $\lambda_{\mathcal{B}}$ / $\lambda_{\bar{d}}$	0.1	0.5	1	3	5
0.1	0.99494	0.99856	0.99862	0.99697	0.99564
0.5	0.96462	0.99689	0.99435	0.99410	0.99297
1	0.92583	0.99509	0.99348	0.99134	0.99067
3	0.87398	0.99708	0.99166	0.98235	0.99042
5	0.87772	0.99492	0.99311	0.99137	0.99231

(b)

AUC $\lambda_{\mathcal{B}}$ / $\lambda_{\bar{d}}$	0.1	0.5	1	3	5
0.1	0.99863	0.99918	0.99878	0.99964	0.99902
0.5	0.99376	0.99847	0.99923	0.99800	0.99847
1	0.98665	0.99705	0.99822	0.99827	0.99579
3	0.95976	0.99383	0.99730	0.99806	0.99674
5	0.95500	0.99367	0.98950	0.99605	0.99648

(c)

AUC $\lambda_{\mathcal{B}}$ / $\lambda_{\bar{d}}$	0.1	0.5	1	3	5
0.1	0.99900	0.99928	0.99930	0.99958	0.99958
0.5	0.99557	0.99907	0.99917	0.99931	0.99916
1	0.98988	0.99893	0.99892	0.99861	0.99787
3	0.96917	0.99721	0.99777	0.99789	0.99835
5	0.95736	0.99383	0.99751	0.99150	0.99697

(d)

AUC $\lambda_{\mathcal{B}}$ / $\lambda_{\bar{d}}$	0.1	0.5	1	3	5
0.1	0.99370	0.99719	0.99907	0.99951	0.99953
0.5	0.98607	0.99888	0.99904	0.99945	0.99895
1	0.98597	0.99752	0.99905	0.99935	0.99952
3	0.96551	0.99153	0.99388	0.99526	0.99807
5	0.95920	0.99428	0.99507	0.99577	0.99434

(e)

AUC $\lambda_{\mathcal{B}}$ / $\lambda_{\bar{d}}$	0.1	0.5	1	3	5
0.1	0.98886	0.99951	0.99920	0.99895	0.99931
0.5	0.98908	0.99577	0.99906	0.99924	0.99961
1	0.98509	0.99340	0.99916	0.99897	0.99928
3	0.97131	0.99635	0.99313	0.99463	0.99908
5	0.96330	0.99158	0.99239	0.98866	0.99797

$\lambda_{\mathcal{B}}$, $\lambda_{\bar{d}}$ and λ_d are employed to balance their contribution, since three parts of the loss function (3) take values from different scales, as shown in Figure 6. They can also be viewed as three weights acting on three parts of the loss function. We show AUC performance of different settings of these three hyperparameters in Table 4. In average,

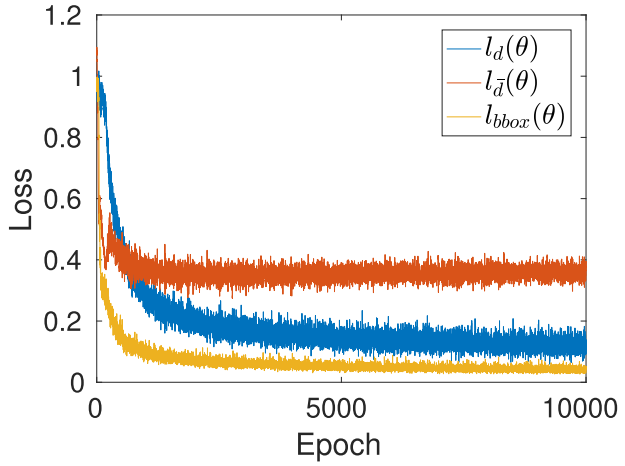


FIGURE 6. Change of value of loss functions $l_{\mathcal{B}}(\theta)$, $l_{\boldsymbol{d}}(\theta)$ and $l_{\bar{\boldsymbol{d}}}(\theta)$ during training phase.

TABLE 5. AUC performance of DoNN obtained using different settings of B on D_{lv}. We use $\lambda_{\boldsymbol{d}} = 0.5$, $\lambda_{\bar{\boldsymbol{d}}} = 0.1$, $\lambda_{\mathcal{B}} = 3$, $G = 7$ for experiments given in this table. Best performance is obtained using $B = 2$.

	$B=1$	$B=2$	$B=3$	$B=5$	$B=10$
AUC	0.99944	0.99964	0.99962	0.99953	0.99915

DoNN achieves *acceptable* AUC performance in most of the hyperparameter settings. DoNN provides better performance when the parameters weight $l_{\mathcal{B}}(\boldsymbol{\theta})$ more than $l_d(\boldsymbol{\theta})$ and $l_{\bar{d}}(\boldsymbol{\theta})$. DoNN also performs better when less weight is given to $l_{\bar{d}}(\boldsymbol{\theta})$ with $\lambda_{\mathcal{B}}$ and λ_d fixed. This is observed when the number of grids not containing docking stations is larger than the number of grids containing docking stations. The parameter $l_{\bar{d}}(\boldsymbol{\theta})$ dominates the value of the loss function if equal weights are given to three parts of the loss function. DoNN achieves the best performance for $\lambda_d = 0.5$, $\lambda_{\bar{d}} = 0.1$, $\lambda_{\mathcal{B}} = 3$. Therefore, we use this setting in the following experiments.

The hyperparameter B controls the number of scored bounding-boxes that are predicted by the DoNN for each grid. The larger B is, the more parameters and computation cost are required. We show that the AUC performance of DoNN at the setting of $\lambda_d = 0.5$, $\lambda_{\bar{d}} = 0.1$ and $\lambda_{\mathcal{B}} = 3$ in Table 5. The AUC performance of DoNN does not benefit a lot from a large B. Since the DoNN achieves the best performance for $B = 2$, B is set to 2 in the following experiments.

The hyperparameter G controls the number of grids into which an image is divided. Fine grained grids increase the number of parameters and computational cost. We perform experiments at $G = \{2, 4, 7, 14\}$, since G must be a factor of the width and height of the input image, which are both 448 in our experiments. The experimental results are shown in Table 6. The best AUC performance is achieved at $G = 7$, and G is set to 7 in the experiments.

TABLE 6. AUC performance of DoNN obtained using different settings of G on D_{lv}. We use $\lambda_{\boldsymbol{d}} = 0.5$, $\lambda_{\bar{\boldsymbol{d}}} = 0.1$, $\lambda_{\mathcal{B}} = 3$, $B = 2$ in experiments of this table. Best performance is obtained using $G = 7$.

	$G=2$	$G=4$	$G=7$	$G=14$
AUC	0.99900	0.99819	0.99964	0.99890

4) COMPARISON OF PERFORMANCE OF DETECTION ALGORITHMS USING THE ORIGINAL TEST DATASET D_{LV}

We first analyze detection performance of YOLO, Faster-RCNN and DoNN using the original test dataset D_{lv}. Figure 7 shows the ROC curve and the associated AUC of three models. The results show that DoNN performs slightly better than Faster-RCNN, and they both outperform YOLO. The AUC of DoNN and Faster-RCNN are 0.99964 and 0.99958, respectively, achieving good performance on D_{lv}. We will show that Faster-RCNN is not as robust to various deformations of D_{lv} as DoNN in the following analyses.

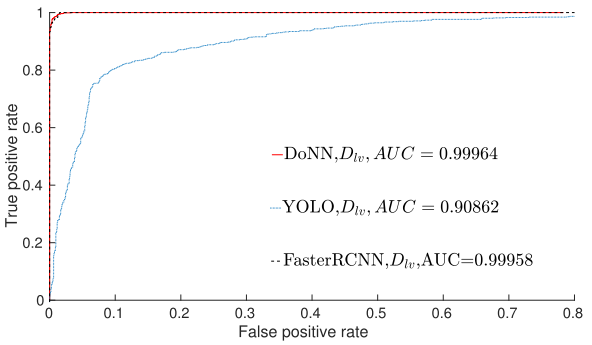


FIGURE 7. ROC curves and the associated AUCs of DoNN, YOLO, and Faster-RCNN computed using D_{lv}.

As a concrete example, we depict feature maps $\mathcal{T}_i$ computed at the i^{th} convolution layer, and the detection result of typical samples for three models in Figure 8, 9 and 10. A feature map $\mathcal{T}_i$ is computed by

$$\mathcal{T}_i = \sum_{j=1}^{N_i} \tau_{i,j}(x, y), \tag{13}$$

where $\tau_{i,j}(x, y)$ denotes the j^{th} feature map of N_i feature maps computed using (2) at the i^{th} convolutional layer. After each pooling, $\mathcal{T}_i$ is down-sampled by a factor which is determined by the architecture of the network. Each $\mathcal{T}_i$ is coupled with a color-bar which indicates its corresponding color scale. For DoNN and YOLO, an additional confidence map $S = \{\bar{S}_i\}_{i=1}^{G^2}$, where $\bar{S}_i = \max(\hat{S}_{i,1}, \hat{S}_{i,2}, \ldots, \hat{S}_{i,B})$ and $|S| = G^2$, is depicted in the last but one figure. Each pixel belonging to the set S indicates the predicted confidence score for its corresponding grid.

Comparing a map $\mathcal{T}_i$ computed using these three models, we conjecture that all three models can be used to learn feature representations of spatial structural patterns of docking stations, which are invariant to change of light to a different extent. In Figure 9b, we depict a map $\mathcal{T}_1$ computed using

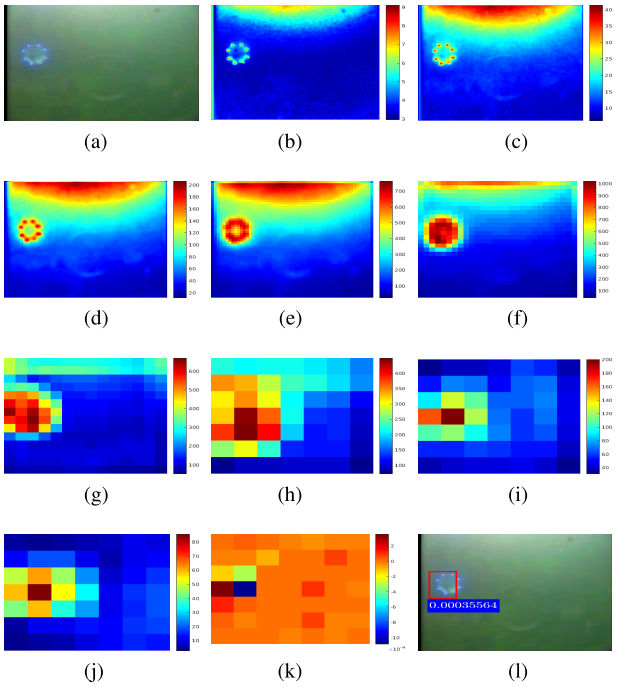


FIGURE 8. Feature maps computed for a false detection predicted by *YOLO* and its detection result on D_{lv}. Figures given from left to right, and top to bottom correspond to its input image, feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i, i = 1, 2 \ldots, 9$, the confidence map and the final detection result. (a) Input image. (b) $\mathcal{T}_1$. (c) $\mathcal{T}_2$. (d) $\mathcal{T}_3$. (e) $\mathcal{T}_4$. (f) $\mathcal{T}_5$. (g) $\mathcal{T}_6$. (h) $\mathcal{T}_7$. (i) $\mathcal{T}_8$. (j) $\mathcal{T}_9$. (k) S. (l)Detection result.

FasterRCNN. We observe that features are activated only in a region containing docking station, although the upper ambient light is stronger than the lower. As for DoNN, effect of ambient light is gradually eliminated as shown in Figure 10. We depicted a map $\mathcal{T}_5$ computed using DoNN in Figure 10f. The results show that features are activated only in a region that contains a docking station. Therefore, DoNN is robust to light variance. As a result, the grid corresponding to the docking station obtained the highest confidence score while others obtained almost zero score. Figure 8 shows a false prediction provided by YOLO. IoU of the prediction is low since estimated conditional class distributions corrupt the final confidence map S, as illustrated in Figure 3.

Since AUC performance of YOLO is worse compared to FasterRCNN and DoNN, we further analyzed the performance of FasterRCNN and DoNN in the following sections.

5) COMPARISON OF PERFORMANCE OF DETECTION ALGORITHMS USING BLURRED IMAGES

Blurred underwater images are observed due to scattering of light [11]. We analyze change of performance of the detection methods using blurred underwater images in a controlled setting for different blurring patterns. Therefore, we generate blurred images from the UDID by employing Gaussian filters [43]

$$I(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d^2}{2\sigma^2}} \tag{14}$$

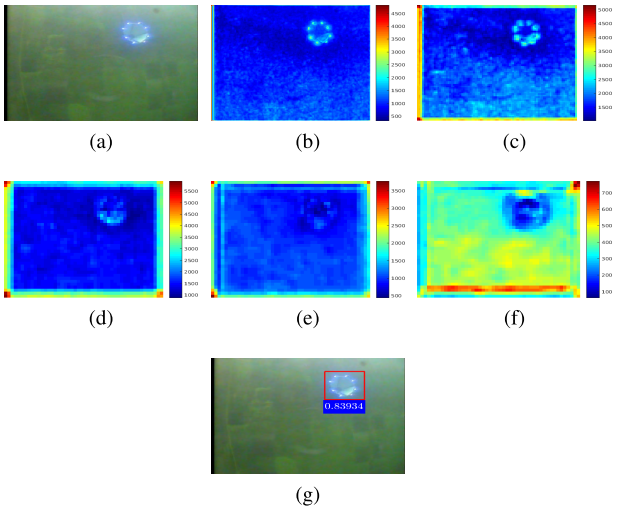


FIGURE 9. Feature maps computed using Faster-RCNN, and its detection result on D_{lv}. Figures given from left to right, and top to bottom correspond to its input image, feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i, i = 1, 2 \ldots, 9$, and the final detection result. (a) Input image. (b) $\mathcal{T}_1$. (c) $\mathcal{T}_2$. (d) $\mathcal{T}_3$. (e) $\mathcal{T}_4$. (f) $\mathcal{T}_5$. (g) Detection result.

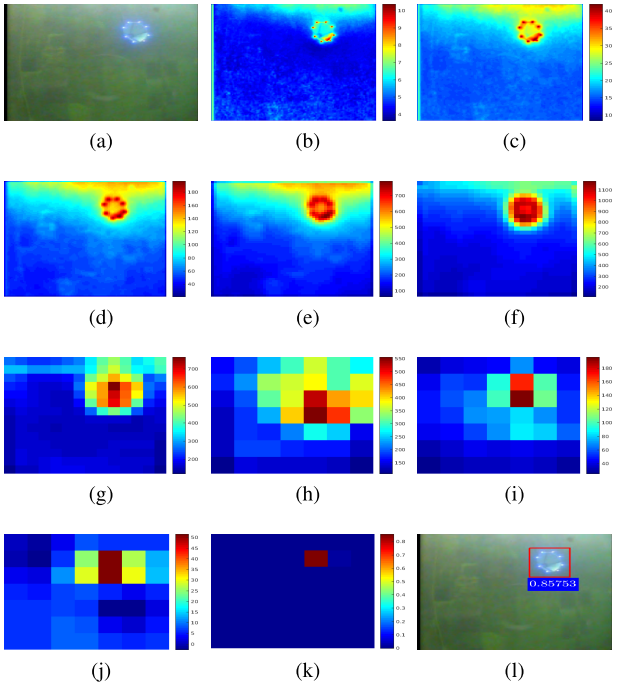


FIGURE 10. Feature maps computed using DoNN, and its detection result on D_{lv}. Figures given from left to right, and top to bottom correspond to its input image, feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i, i = 1, 2 \ldots, 9$, the confidence map, and the final detection result. (a) Input image. (b) $\mathcal{T}_1$. (c) $\mathcal{T}_2$. (d) $\mathcal{T}_3$. (e) $\mathcal{T}_4$. (f) $\mathcal{T}_5$. (g) $\mathcal{T}_6$. (h) $\mathcal{T}_7$. (i) $\mathcal{T}_8$. (j) $\mathcal{T}_9$. (k) S. (l)Detection result.

with varying standard deviation σ to simulate blurring in underwater images, where $d = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ is the distance between a pixel (x, y) and a filter center pixel (x_c, y_c). The filter size is set to $2 \times \lceil 2 \times \sigma \rceil + 1$. We sample the deviation σ uniformly by $\sigma \in [1, 10]$. The larger values the σ takes, the more blurred images are obtained. The dataset obtained after employment of blurring is called by $D_{bl\sigma}$.

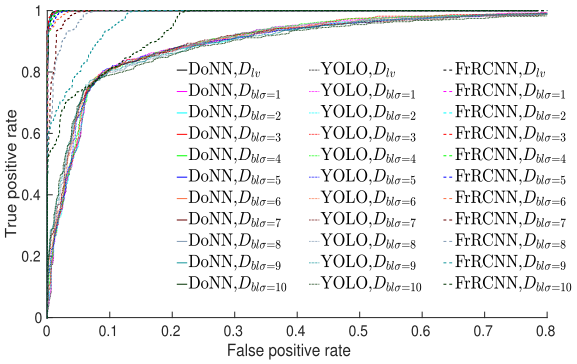


FIGURE 11. ROC curves of DoNN, YOLO, Faster-RCNN computed for various degrees of blurring. The corresponding AUCs are given in Table 10.

Table 7 shows sample blurred images and the corresponding detection results predicted using FasterRCNN and DoNN, respectively. The predicted probability values decrease as σ increases. This indicates that both FasterRCNN and DoNN provide less confident results in their prediction by the increase of blurring. FasterRCNN suffers from blurring more than DoNN as observed in Figure 11 and Table 10. The AUC of FasterRCNN degrades from 0.99958 to 0.95785 as σ varies from 1 to 10 while that of DoNN degrades from 0.99964 to 0.99948. DoNN outperforms FasterRCNN in nine out of ten levels of blurring. Fine-grained details of docking stations are lost by the increase of blurring, but by preserving spatial structural patterns of docking stations. We conjecture that DoNN outperforms FasterRCNN in blurred underwater images since DoNN can be used to learn *better* feature representations of docking stations compared to FasterRCNN. DoNN can still keep a high activation in its feature maps although blurring is very high, as shown in Table 9. However, activation of FasterRCNN in feature maps becomes almost as weak as background in cases of $\sigma = 8$ and $\sigma = 10$, resulting in incorrect detection, as shown in Table 8.

6) COMPARISON OF PERFORMANCE OF DETECTION ALGORITHMS UNDER COLOR SHIFT

Color shift is determined by various factors in underwater environments, such as attenuation. In order to compare performance of three models in underwater images with color shift, underwater images with color shift at different rates of hue, saturation and value shift are created by

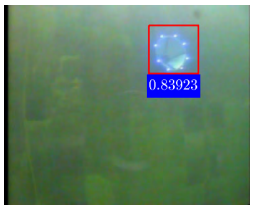
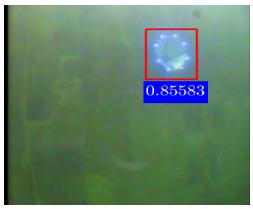
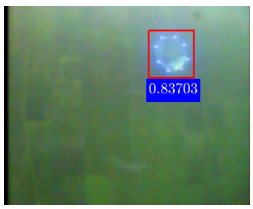
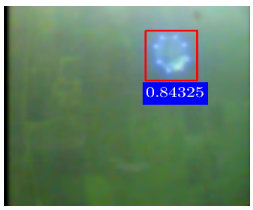
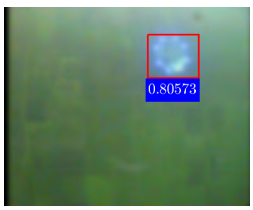
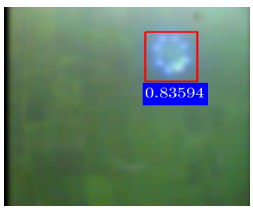
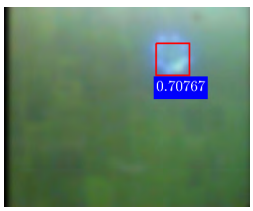
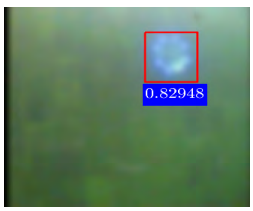
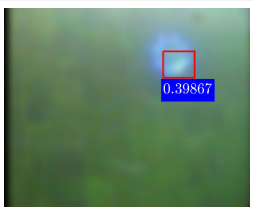
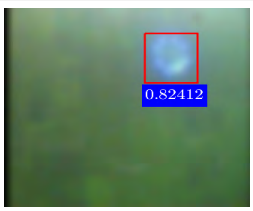
$$L_{out_p} = \lambda_p L_{in_p}, p \in \{H, S, V\}, \tag{15}$$

where L_{in_p} denotes one of HSV components of images in D_{lv}. Datasets after hue, saturation and value shift are indicated by D_{λ_h}, D_{λ_s} and D_{λ_v} respectively.

In order to set a reasonable range for λ_p, we first compute the distribution of λ_p by

$$\bar{\lambda}_p = \frac{1}{w \cdot h} \sum_{c=1}^{3} \sum_{x=1}^{w} \sum_{y=1}^{h} \frac{L_{out_p}(x, y)}{L_{in_p}(x, y)}, \quad p \in \{H, S, V\}, \tag{16}$$

TABLE 7. Sample blurred images, and detection results provided by FasterRCNN and DoNN on the set of blurred images.

σ	FrRCNN	DoNN
1		
2		
4		
6		
8		
10		

where $\{L_{out_p}, L_{in_p}\}$ is an image pair from D_{lv}, (x, y) is the image coordinate, w is the width of the image and h is the height of the image. We compute $\bar{\lambda}_p$ for all possible image pairs in D_{lv}. The distribution of $\bar{\lambda}_p$ is shown in Figure 12a, 12b and 12c. Due to their symmetry, we sample λ_p from [0.5, 1].

Sample images after hue shift deformation and corresponding detection results of DoNN and FasterRCNN are shown in Table 11. In this sample, the increment of hue shift results in less confidence for DoNN while total incorrect detection

TABLE 8. Feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i$, $i = 1, 2, 5$, and a detection result of FasterRCNN for $\sigma = 8$ and $\sigma = 10$.

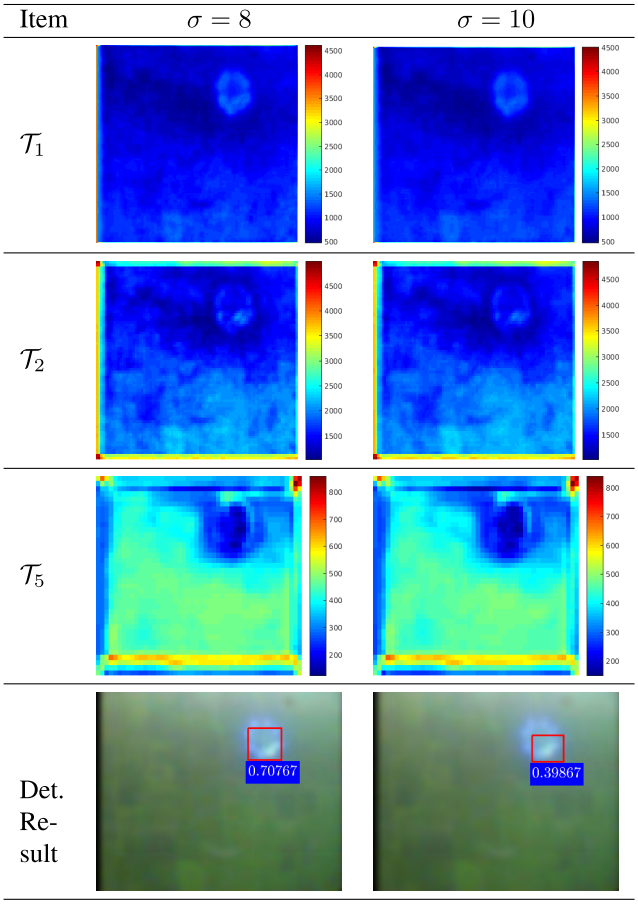
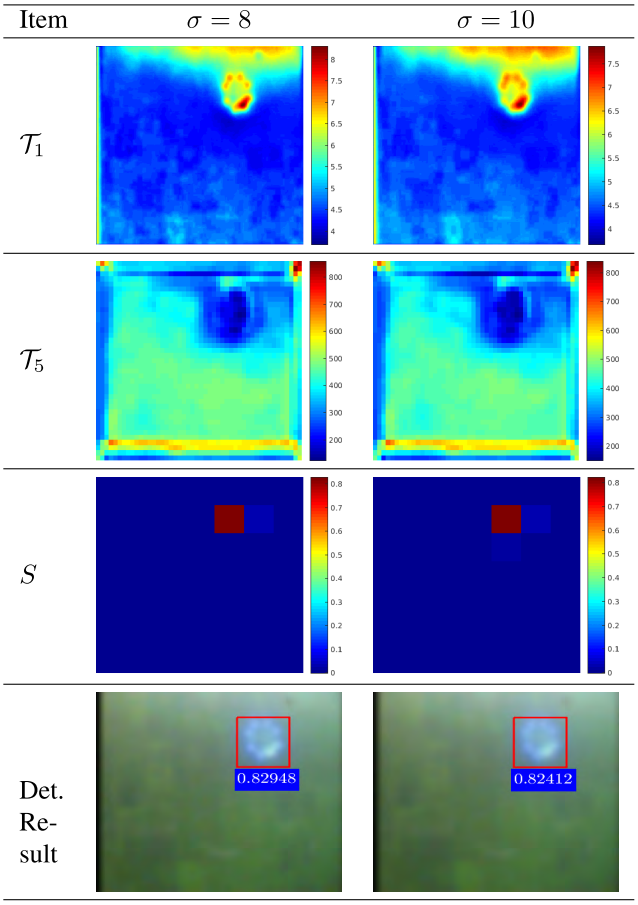


TABLE 9. Feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i$, $i = 1, 5$, confidence map, and a detection result of DoNN for $\sigma = 8$ and $\sigma = 10$.



for FasterRCNN. Figure 13 and Table 13 show the ROC curve and associated AUC of three models in hue shift, respectively. DoNN outperforms FasterRCNN in all cases. DoNN can still achieve an acceptable performance in the extreme case $\lambda_h = 0.5$ while FasterRCNN performs poorly. Notable performance difference between these two models occurs at $\lambda_h = 0.5$ and $\lambda_h = 0.6$. Table 16 shows feature maps of the first, second and fifth convolution layer of FasterRCNN at $\lambda_h = 0.5$ and $\lambda_h = 0.6$. Activation of $\mathcal{T}_1$ and $\mathcal{T}_5$ is very weak in the region of the docking station, giving rise to the final incorrect detection. Table 17 shows feature maps of the first, fifth convolution layer, confidence map S and detection results of DoNN at $\lambda_h = 0.5$ and $\lambda_h = 0.6$. DoNN remains relatively high activation in the docking station region in $\mathcal{T}_1$ in Table 17, resulting in less confident but correct detection of docking stations. The confidence map S of DoNN shows that DoNN feels more uncertain than cases without hue shift. Three neighbouring grids are with similar confidence, but the correct grid overwhelms.

Saturation indicates the amount of grey in the color. A color is grey when its saturation value is 0 while is primary color when its saturation value is 1. Table 12 shows sample images after saturation shift and associated detection results

TABLE 10. AUC of DoNN, YOLO, Faster-RCNN computed for various degrees of blurring, and the corresponding ROC curves are given in Figure 11.

AUC / Model / σ	DoNN	YOLO	FrRCNN
D_{lv}	0.99964	0.90862	0.99958
1	0.99964	0.91411	0.99959
2	0.99966	0.91025	0.99943
3	0.99964	0.91446	0.99971
4	0.99965	0.91465	0.99942
5	0.99967	0.91147	0.99899
6	0.99967	0.91453	0.99854
7	0.99953	0.91484	0.99493
8	0.99957	0.91549	0.99210
9	0.99958	0.91333	0.97791
10	0.99948	0.90734	0.95785
Average	0.99961	0.91305	0.99185

of FasterRCNN and DoNN. The color becomes closer to grey with the increment of λ_s. We show ROC curve and associated AUC of three models in Figure 14 and Table 14. As depicted, DoNN outperforms FasterRCNN in all levels of Saturation

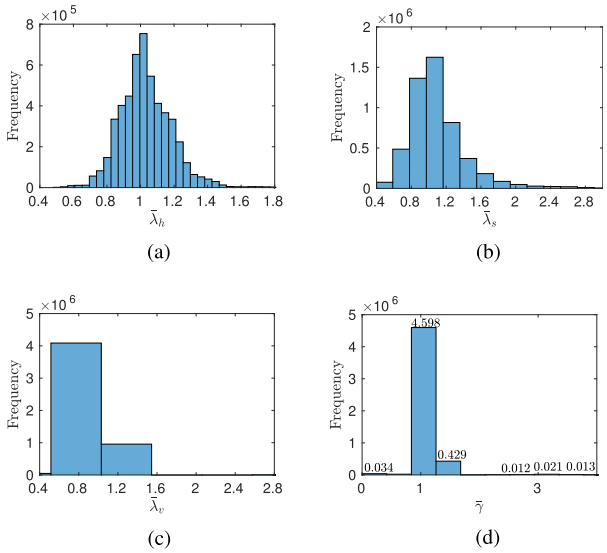


FIGURE 12. Distribution of $\bar{\lambda}_h$, $\bar{\lambda}_s$, $\bar{\lambda}_v$ and $\bar{\gamma}$. (a) Distribution of $\bar{\lambda}_h$. (b) Distribution of $\bar{\lambda}_s$. (c) Distribution of $\bar{\lambda}_v$. (d) Distribution of $\bar{\gamma}$.

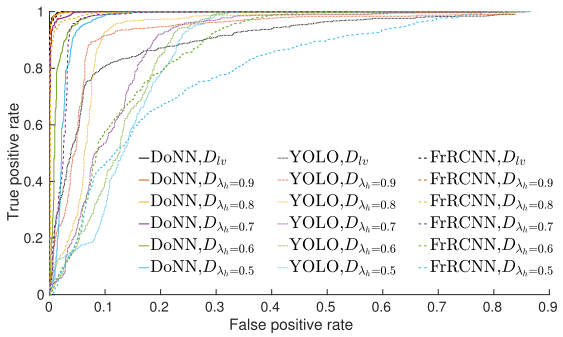


FIGURE 13. ROC curves of DoNN, YOLO, Faster-RCNN computed under hue shift. Corresponding AUCs are given in Table 13.

shift. The AUC of FasterRCNN declines from 0.99958 to 0.99713 while DoNN from 0.99964 to 0.99953 as λ_s varies from 1 to 0.5.

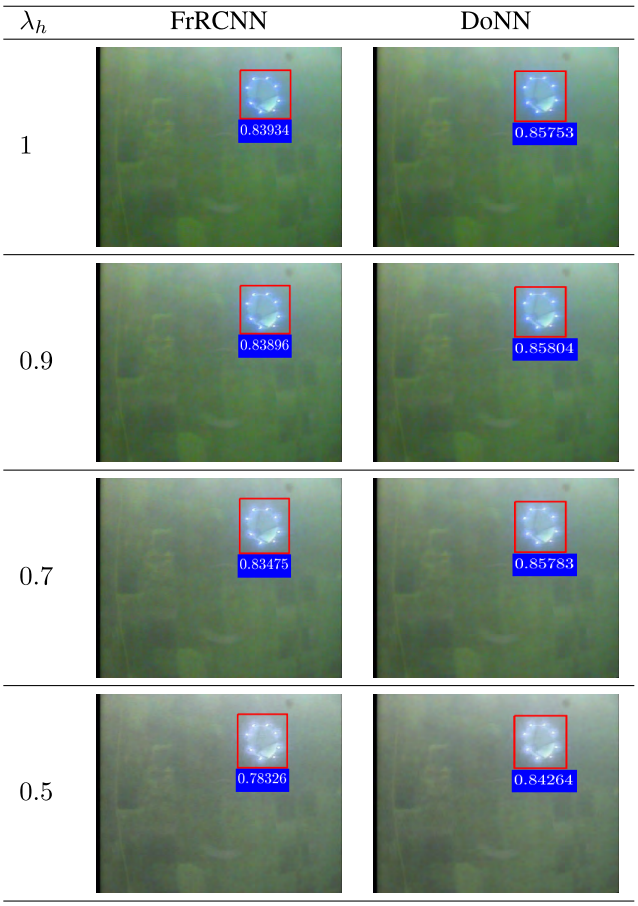
Value component describes the brightness of colors. Images become less bright with the increment of λ_v. λ_v with a low value corresponds to a dim underwater environment. As ROC curves and associated AUC shown in Figure 15 and Table 15, DoNN and FasterRCNN are both robust to Value shift. No significant degradation arises in different levels of Value shift. DoNN goes ahead in the case of $\lambda_v = 0.5$ and lags slightly behind FasterRCNN in other cases. But DoNN outperforms FasterRCNN in terms of average performance. The average AUC of DoNN is 0.99965 overall in contrast to 0.99774 of FasterRCNN.

To sum up, DoNN is more robust and credible than Faster-RCNN in underwater images with color shift.

7) COMPARISON OF PERFORMANCE OF DETECTION ALGORITHMS UNDER CONTRAST SHIFT

In order to compare the performance of the detection methods in underwater environment under change of contrast, we generate contrast adjustment datasets using gamma

TABLE 11. Sample images after hue shift deformation and detection results predicted by FasterRCNN and DoNN.



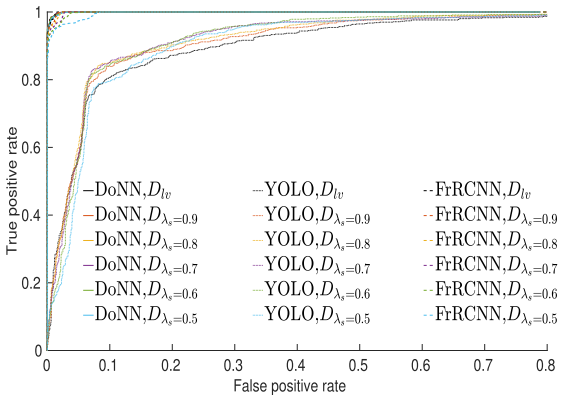


FIGURE 14. ROC curves of DoNN, YOLO, and Faster-RCNN computed under saturation shift. The corresponding AUCs are given in Table 14.

transformation

$$I_{out} = I_{in}^{\gamma}, \tag{17}$$

where I_{out}, I_{in} and γ are input images, output images and the parameter of gamma transformation, respectively. Contrast of images increases as $\gamma < 1$ while the contrast decreases as $\gamma > 1$. The dataset obtained after contrast deformation is denoted by $D_{ct\gamma}$.

TABLE 12. Sample images after saturation shift deformation and detection results predicted by FasterRCNN and DoNN.

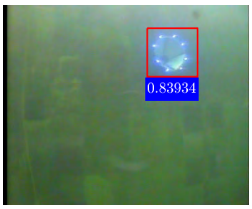
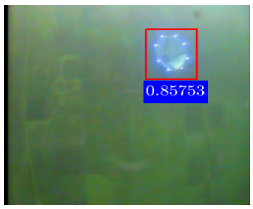
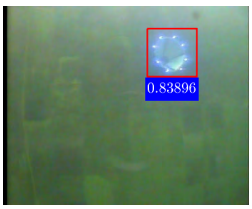
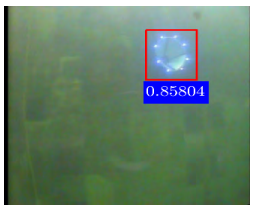
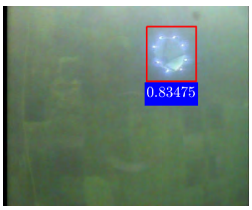
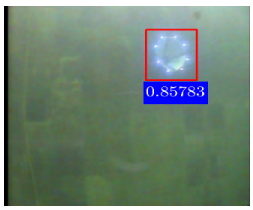
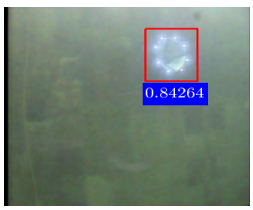
λ_s	FrRCNN	DoNN
1		
0.9		
0.7		
0.5		

TABLE 13. AUCs of DoNN, YOLO and Faster-RCNN computed under hue shift, and the corresponding ROC curves are shown in Figure 13.

AUC / Model / λ_h	DoNN	YOLO	FrRCNN
1	0.99964	0.90862	0.99958
0.9	0.99956	0.93559	0.99888
0.8	0.99951	0.93065	0.99564
0.7	0.99824	0.89211	0.97628
0.6	0.98667	0.87664	0.87605
0.5	0.97405	0.86540	0.80774
Average	0.99160	0.9008	0.93092

We computed the distribution of γ, which is shown in Figure 12d, by

$$\bar{\gamma} = \frac{1}{w \cdot h} \sum_{c=1}^{3} \sum_{x=1}^{w} \sum_{y=1}^{h} \frac{logI_{out}(x, y)}{logI_{in}(x, y)}, \tag{18}$$

where w, h and c are the width, height and channel of I_{in} and I_{out}. $\{I_{in}, I_{out}\}$ is an image pair belonging to D_{lv}. $I_{out}(x, y)$ denotes the pixel value of I_{out} at the (x, y) location. According to the distribution of γ shown in Figure 12d, we sample γ from $\gamma \in [0.2, 3.5]$.

TABLE 14. AUCs of DoNN, YOLO, Faster-RCNN computed under various levels of saturation shift, and the corresponding to ROC curves are shown in Figure 14.

AUC / Model / λ_s	DoNN	YOLO	FrRCNN
1	0.99964	0.90862	0.99958
0.9	0.99961	0.91990	0.99941
0.8	0.99961	0.92510	0.99922
0.7	0.99960	0.92634	0.99891
0.6	0.99959	0.92680	0.99851
0.5	0.99953	0.91429	0.99713
Average	0.99959	0.92249	0.99864

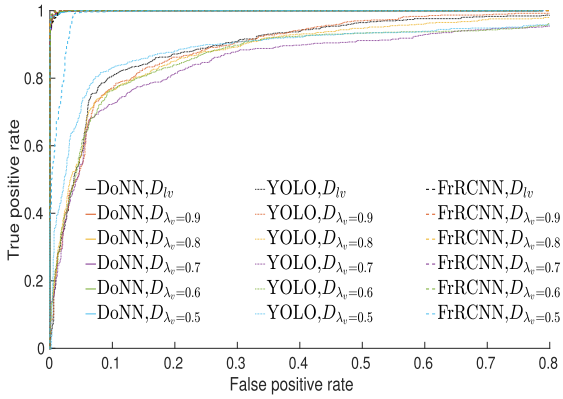


FIGURE 15. ROC curves of DoNN, YOLO, Faster-RCNN computed under for value shift. Corresponding AUCs are given in Table 15.

TABLE 15. AUCs of DoNN, YOLO, Faster-RCNN computer for various levels of value shift, and the corresponding ROC curves are shown in Figure 15.

AUC / Model / λ_v	DoNN	YOLO	FrRCNN
1	0.99964	0.90862	0.99958
0.9	0.99964	0.90621	0.99981
0.8	0.99965	0.89723	0.99988
0.7	0.99963	0.87105	0.99976
0.6	0.99968	0.88644	0.99979
0.5	0.99965	0.90317	0.98946
Average	0.99965	0.89282	0.99774

Table 21 shows sample images obtained by contrast adjustment of different levels. Intuitively, the docking station is more clearly observed as γ increases, and is less clear as γ decreases in the image. We give ROC curves and the associated AUCs of three detection algorithms in Figure 16 and Table 20, respectively. On average, DoNN performs better than FasterRCNN. The average AUC of DoNN is 0.97857 over all levels of contrast adjustment while that of Faster-RCNN is 0.84282. As γ is high, Faster-RCNN outperforms DoNN by a tiny margin, but lags behind significantly if γ is

TABLE 16. Feature maps $\mathcal{T}_i$ of the i^{th} layer $Conv_i$, $i = 1, 2, 5$ and detection results provided by FasterRCNN for $\lambda_h = 0.6$ and $\lambda_h = 0.5$.

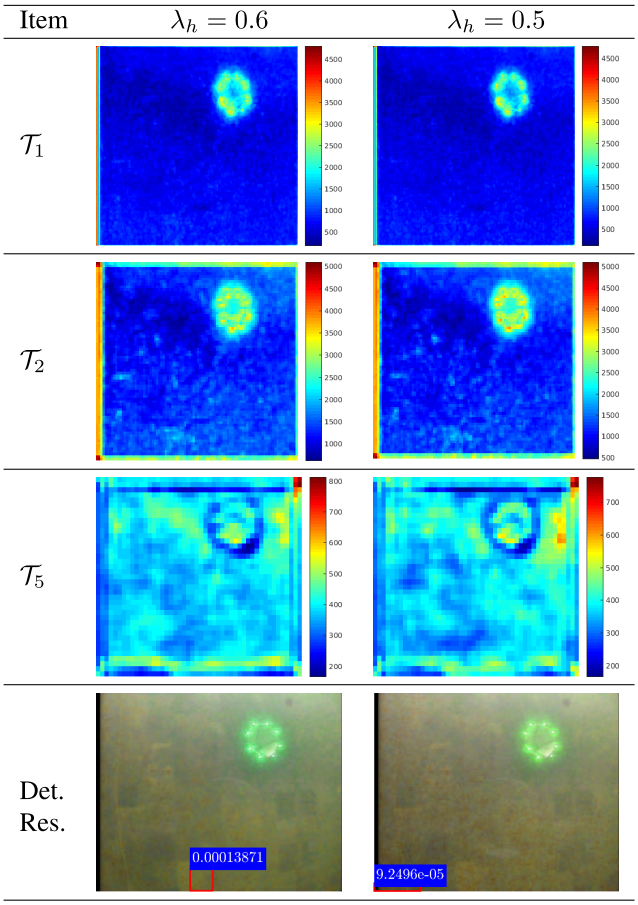
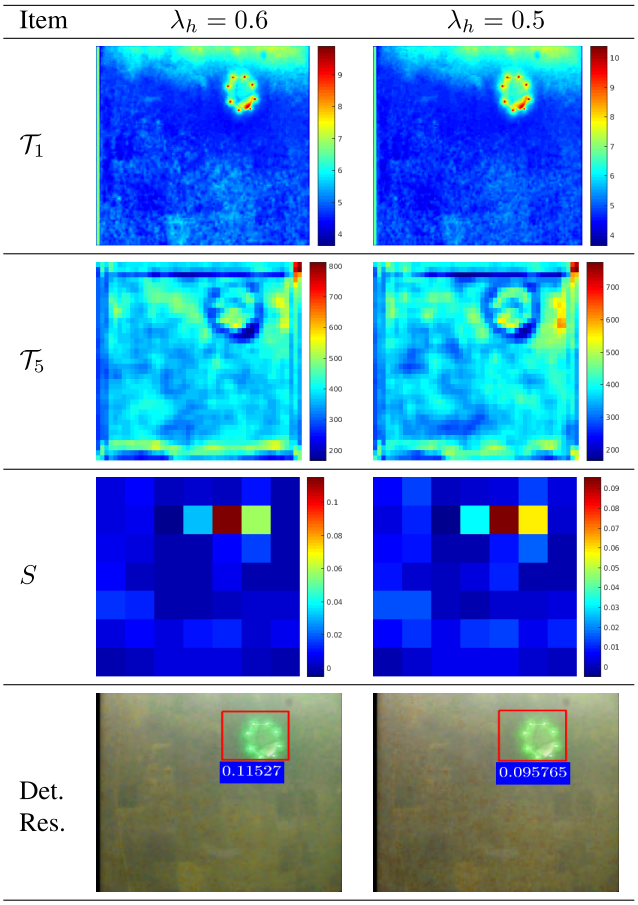


TABLE 17. Feature maps $\mathcal{T}_i$ of the i^{th} layer $Conv_i$, $i = 1, 5$, confidence map and detection results provided by DoNN for $\lambda_h = 0.6$ and $\lambda_h = 0.5$.



low. The AUC of Faster-RCNN becomes 0.90320 if $\gamma = 0.4$, and even more acute with $AUC = 0$, for $\gamma = 0.2$. This means that all the detections fail to surpass the IoU criterion (50%). It is primarily owing to the weak activation of FasterRCNN in feature maps. We compare feature maps $\mathcal{T}_1$ and $\mathcal{T}_5$ of DoNN and FasterRCNN in Table 22 and 23 to examine the performance difference between DoNN and FasterRCNN. For $\gamma = 0.4$, activation of $\mathcal{T}_1$ and $\mathcal{T}_5$ of FasterRCNN becomes quite weak in the docking station region, resulting in the final incorrect detection, as shown in Table 22. It becomes more acute for $\gamma = 0.2$. Activations computed using $\mathcal{T}_1$ and $\mathcal{T}_5$ are almost as weak as computed in their background in the docking station region, and thus they provide incorrect prediction. However, it provides high activation values, and salient spatial structural patterns in $\mathcal{T}_1$ and $\mathcal{T}_5$ of DoNN for $\gamma = 0.4$ as shown in Table 23. As a result, relative high confidence is obtained for only one grid in the confidence map S of DoNN. When γ is equal to 0.2, DoNN keeps a distinguishable docking station spatial pattern in its $\mathcal{T}_1$ and $\mathcal{T}_5$, although activation gets weaker than $\gamma = 0.4$. The confidence map S contains more grids with relatively high confidence, but only the correct one overwhelms as shown in Table 23.

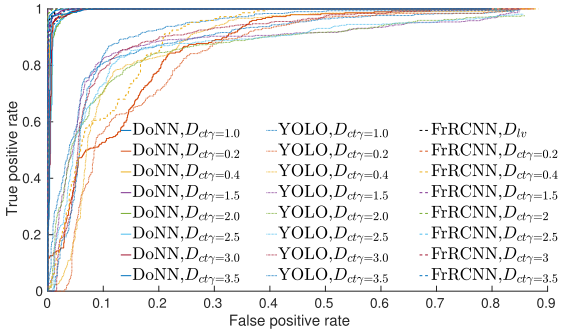


FIGURE 16. ROC curves of DoNN, YOLO, and Faster-RCNN computer under various contrast conditions. The corresponding AUCs are shown in Table 20.

8) COMPARISON OF PERFORMANCE OF DETECTION ALGORITHMS FOR MIRROR IMAGES

As mentioned in Section I, mirror images result from *total internal reflection*, and they are observed when cameras are within the *critical angle*, as shown in Figure 19d. It poses a nasty problem for detection of docking stations due to its very similar appearance with real docking stations. In order to compare performance of three methods under *total internal*

TABLE 18. Feature maps $\mathcal{T}_i$ of the i^{th} layer $Conv_i$, $i = 1, 2, 5$ and detection results provided by FasterRCNN for $\lambda_s = 0.6$ and $\lambda_s = 0.5$.

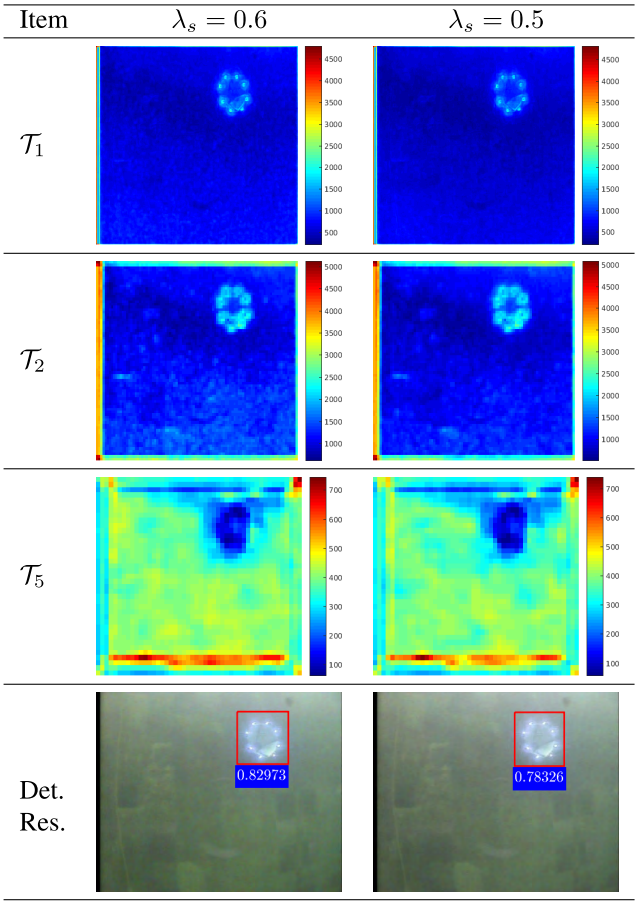
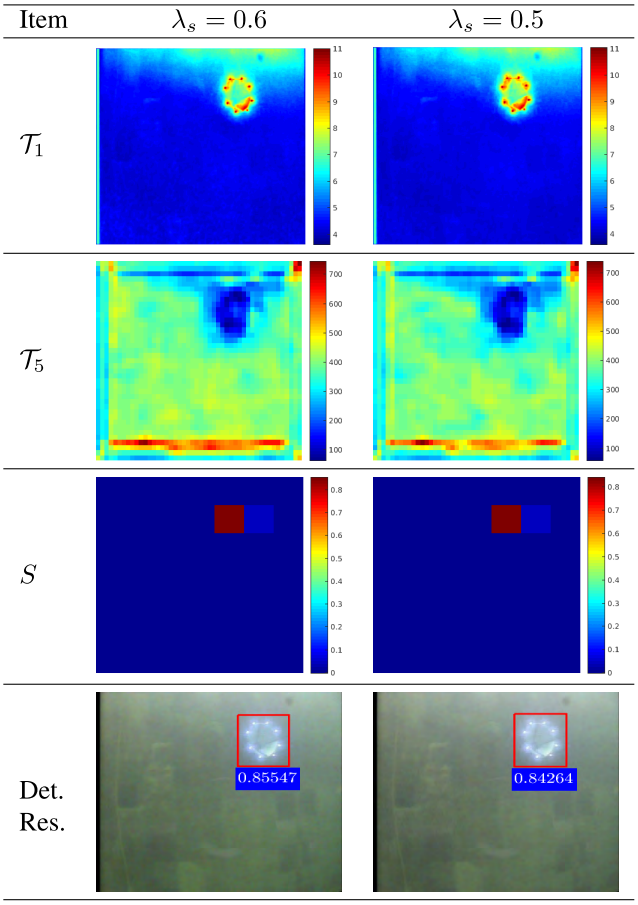


TABLE 19. Feature maps $\mathcal{T}_i$ of the i^{th} layer $Conv_i$, $i = 1, 5$, confidence map and detection results provided by DoNN for $\lambda_s = 0.6$ and $\lambda_s = 0.5$.



reflection, we establish a dataset D_{mirr} by attaching a mirror image to every foreground image in D_{lv}. To this end, image editing [44] is employed to merge a mirror image patch to the upper side of the docking station of original images as real as possible. The merging process is illustrated in Figure 17.

It is shown in Figure 20 that all three CNN-based models are able to distinguish real docking stations from their mirror images with no performance degradation. This attributes the success to the learning ability of CNNs. It is also shown in Figure 20 that DoNN slightly outperforms FasterRCNN. Figure 18 and 19 show feature maps of a natural mirror image generated by DoNN and FasterRCNN, respectively. It is shown in Figure 18 that activations computed for the real docking station computed in $\mathcal{T}_1$ of FasterRCNN is stronger than those for the mirror docking station. This enables FasterRCNN to discriminate real docking stations from mirror ones. Figure 19 shows $\mathcal{T}_1$, $\mathcal{T}_5$ and confidence map S of DoNN. Activation of the mirror docking station is almost as high as the real docking station in $\mathcal{T}_1$, $\mathcal{T}_5$ of DoNN. Even so, confidence map S of DoNN contains only one highly confident grid which is the correct prediction. Next, we will analyze this phenomenon. We conjecture that DoNN can be used to estimate distribution of relative spatial locations between

TABLE 20. AUCs of DoNN, YOLO, and Faster-RCNN computed under various contrast conditions, and the corresponding ROC curves are shown in Figure 16.

AUC / Model / γ	DoNN	YOLO	FrRCNN
1	0.99964	0.90862	0.99958
0.2	0.87402	0.84775	0
0.4	0.99738	0.88640	0.90320
1.5	0.99884	0.88991	0.99997
2.0	0.99439	0.87390	0.99990
2.5	0.99392	0.87943	0.99962
3	0.99505	0.90531	0.99873
3.5	0.99637	0.91929	0.99833
Average	0.97857	0.88600	0.84282

mirror and real docking stations. DoNN learns feature representations of mirror images of docking stations that appear more likely on the upper side rather than real docking stations.

In the experimental analyses, we have to make sure that it is the relative spatial location or appearance that enables DoNN to distinguish real docking stations from mirror ones. To this end, we carry out two experiments. First, in

TABLE 21. Sample images obtained by contrast adjustment, and the corresponding detection results provided by FasterRCNN and DoNN.

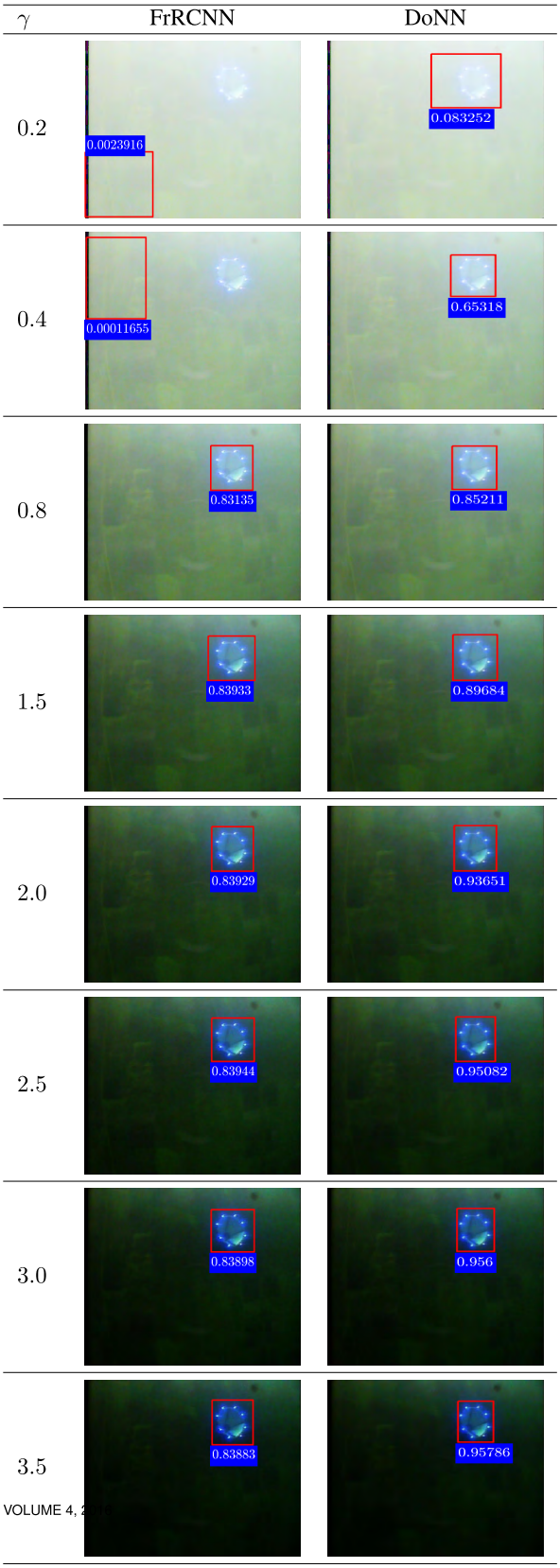


TABLE 22. Feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i$, $i = 1, 2, 5$, and the detection result provided by FasterRCNN for $\gamma = 0.4$ and $\gamma = 0.2$.

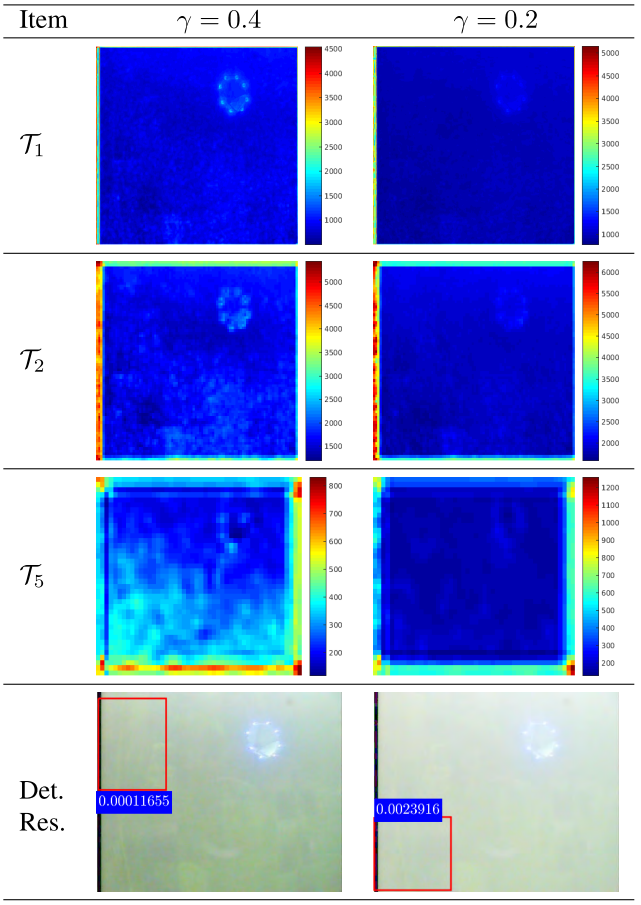
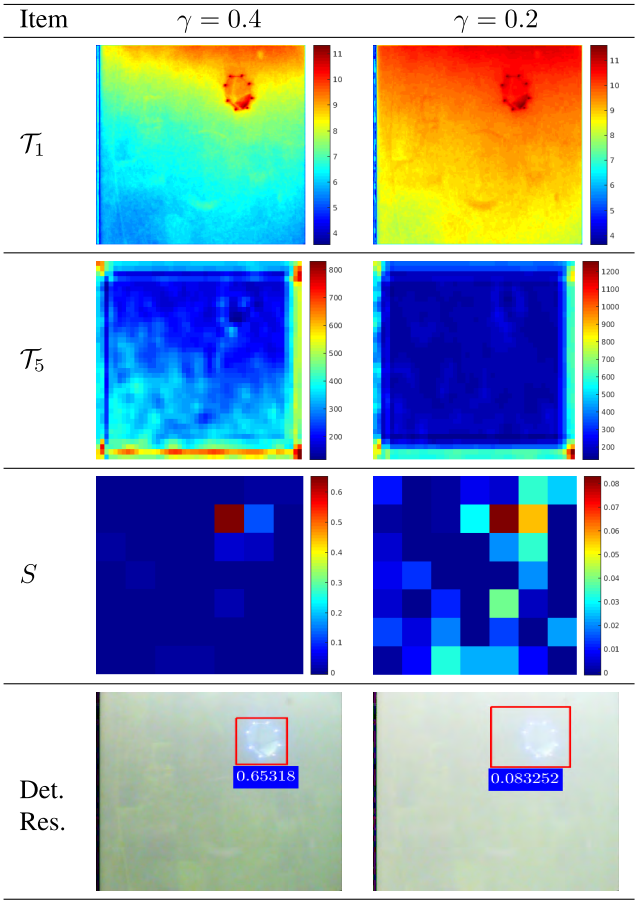


Figure 19d, we replace the patch located in the second quadrant of the input image by the patch located in the third quadrant such that docking stations observed in the second and third quadrant share the same appearance, gaining Figure 21d. Then, we input the simulated image into DoNN. The image obtained by replacement, its feature maps, confidence map and detection results are shown in Figure 21. Due to their same appearance, feature maps computed in the second and the third quadrant are exactly same. DoNN both assigns high confidence to two docking stations, but more to the lower one. In other words, DoNN tends to believe that the lower one is the real docking station. In addition, we compare the confidence of prediction before and after the replacement in Figure 19d and 21d, respectively. The confidence falls from 0.74806 computed before replacement to 0.49781 computed after replacement. It reflects that the appearance contributes to the prediction of DoNN as well. Second, we replace the patch located in the third quadrant of the input image given in Figure 19d by the patch located in the second quadrant of itself, such that all docking stations observed in the image are mirror ones. Then, the obtained

TABLE 23. Feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i$, $i = 1, 5$, confidence map, and the detection result provided by DoNN for $\gamma = 0.4$ and $\gamma = 0.2$.



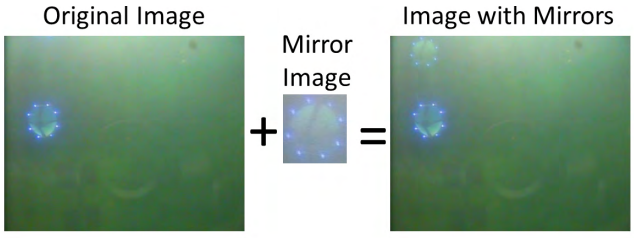


FIGURE 17. The merging process of synthetic mirror images. A mirror image is merged to the original image on the upper side of the real docking station.

image is fed to DoNN. The result is shown in Figure 22. DoNN still provides higher prediction score for the lower one compared to the upper one, although they are both identical mirror images. But the score in Figure 22d is less than the score in Figure 21d. Therefore, we can conclude that DoNN has learned not only feature representations of appearance, but also the distribution of relative spatial locations between real and mirror docking stations.

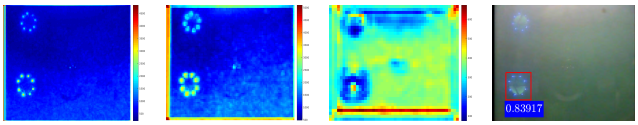


FIGURE 18. Feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i$, $i = 1, 2, 5$, and the corresponding detection result of FasterRCNN for mirror images. (a) $\mathcal{T}_1$. (b) $\mathcal{T}_2$. (c) $\mathcal{T}_5$. (d) Detection results.

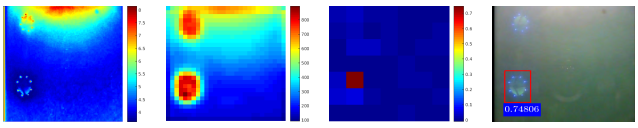


FIGURE 19. Feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i$, $i = 1, 5$, confidence map, and the detection result of DoNN for mirror images. (a) $\mathcal{T}_1$. (b) $\mathcal{T}_5$. (c) S. (d) Detection results.

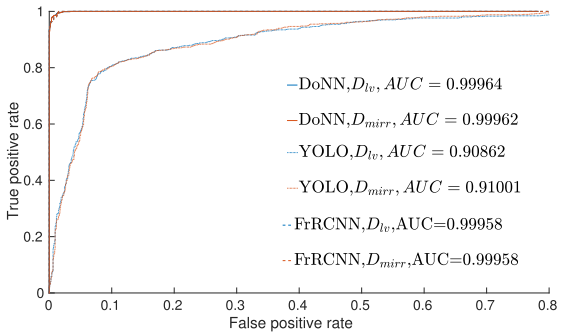


FIGURE 20. ROC curves of DoNN, YOLO, and Faster-RCNN computed for D_{mirr} which contains mirror images.

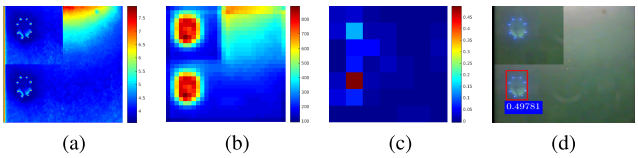


FIGURE 21. Feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i$, $i = 1, 5$, confidence map, and the detection result of DoNN for replacement of the second quadrant of Figure 19d by its third quadrant. (a) $\mathcal{T}_1$. (b) $\mathcal{T}_5$. (c) S. (d) Detection results.

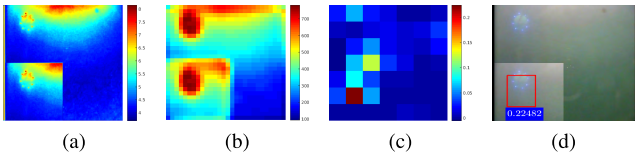


FIGURE 22. Feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i$, $i = 1, 5$, confidence map, and the detection result of DoNN for replacement of the third quadrant of Figure 19d by its second quadrant. (a) $\mathcal{T}_1$. (b) $\mathcal{T}_5$. (c) S. (d) Detection results.

9) COMPARISON OF PERFORMANCE OF DETECTION ALGORITHMS UNDER NON-UNIFORM ILLUMINATION

Non-uniform illumination is commonly observed in real underwater images. In order to compare detection

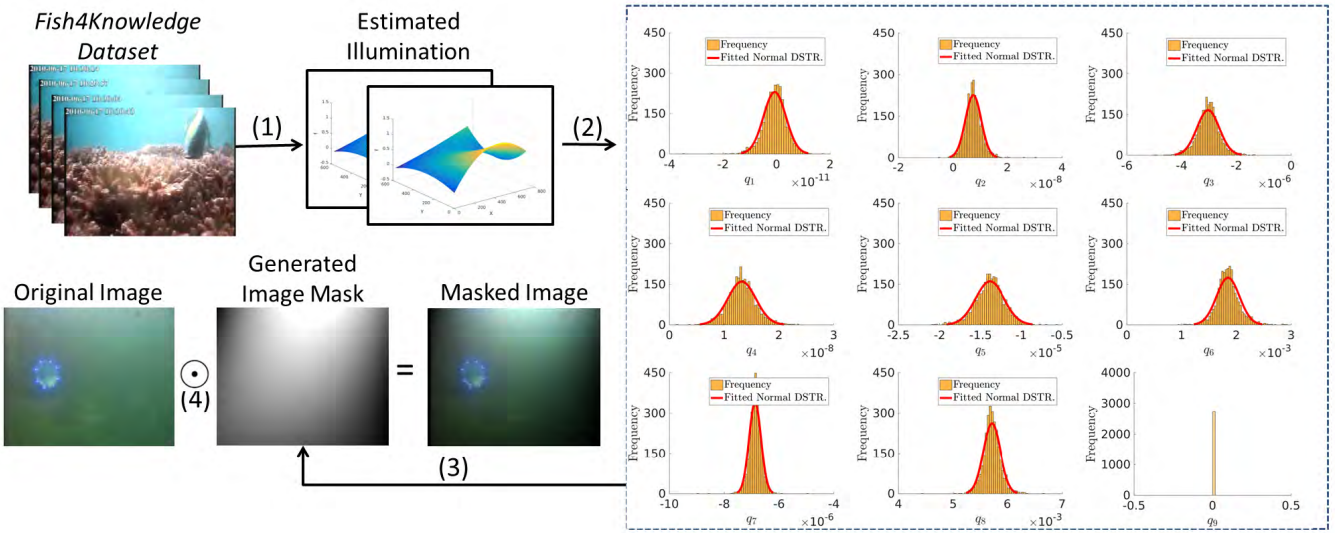


FIGURE 23. An illustration of our method proposed used for generating non-uniform illumination. (1) Estimation of undersea non-uniform illumination: Our proposed method first takes the value component (V) of images belonging to the *Fish4Knowledge* dataset as input. Next, illumination values are estimated, and polynomial coefficients Q are computed by fitting (19). (2) Modeling distribution of coefficients: Distributions of coefficients are estimated by fitting the distributions to Gaussian models $\mathcal{G}$. (3) Generating new samples using non-uniform illumination: First, a set of samples $\hat{Q}$ is drawn from the Gaussian model $\mathcal{G}$ estimated in the step (2). Then, the polynomial function $f\left(x, y; \hat{Q}\right)$ is computed using the drawn samples $\hat{Q}$ in (19). (4) Generated illumination values are applied to D_{lv}.

performance in an underwater environment as close as to real undersea environment of non-uniform illumination, we apply the non-uniform illumination drawn from a subset of the *Fish4Knowledge* dataset (*luminosity changes*) [45], to our D_{lv} dataset which was collected in an indoor pool. The *Fish4Knowledge* dataset was constructed using images captured in a real outdoor undersea environment, and used for detecting targets in noisy underwater environments. The subset *luminosity changes* is specific for underwater luminosity changes. The new dataset obtained after applying undersea non-uniform illumination to D_{lv} is denoted by D_{nu}.

Polynomials are utilized for non-uniform illumination correction in transmission electron microscopy (TEM) images [46]. In our work, it is used in an opposite way, in order to generate non-uniform illumination. Details of non-uniform illumination generation procedure are shown in Figure 23, and explained as follows:

1) Estimation of undersea non-uniform illumination: In order to estimate undersea non-uniform illumination, we fit a low-order ($m = n = 2$) bivariate polynomial to the Value component (HSV color space) of every frame in the *Fish4Knowledge* dataset. The polynomial is represented by

$$\begin{aligned} f(x, y; Q) = {} & q_1 x^n y^m + q_2 x^{(n-1)} y^m + \ldots + q_{n+1} y^m \\ & + \ldots + q_{n+2} x^n y^{(m-1)} + \\ & + q_{n+3} x^{(n-1)} y^{(m-1)} + \\ & + \ldots + q_{2(n+1)} y^{(m-1)} \\ & + \ldots + q_{m(n+1)+1} x^n + q_{m(n+1)+2} x^{(n-1)} \\ & + \ldots + q_{(n+1)(m+1)}, \end{aligned} \tag{19}$$

where Q stands for the set of parameters. After fitting, $Q_i = (q_{i,1}, \ldots, q_{i,j}, \ldots, q_{i,(n+1)(m+1)}) \in \mathbb{R}^{(n+1)(m+1)}$ is obtained for the i^{th} frame.

2) Modeling distribution of coefficients: Suppose that $q_j (j = (1, 2, \ldots, (n+1)(m+1)))$ are independent random variables. Then, $q_{i,j}$ is viewed as the i^{th} sample drawn from the distribution of q_j. The distribution of q_j is shown in the histogram given in Figure 23. Obviously, the distribution of q_j forms a Gaussian shape. It is fitted by a Gaussian Distribution $\mathcal{G}_j(\mu_j, \sigma_j^2)$, as depicted in Figure 23.

3) Generation of new underwater images using non-uniform illumination: Generation of new illumination involves drawing samples from the obtained distribution $\mathcal{G}_j(\mu_j, \sigma_j^2)(j = (1, \ldots, (n+1)(m+1)))$, and an evaluation of polynomial function (19). For each image $I_o \in D_{lv}$, a set of samples $\hat{Q}$ is drawn from $\mathcal{G}_j(\mu_j, \sigma_j^2)$, and a new non-uniform illumination I_δ is generated.

4) Applying generated illumination to D_{lv}: Since an image can be modeled as a multiplicative effect [46], the value component (V) I_{nuv} of newly generated image I_{nu} with non-uniform illumination can be obtained by

$$I_{nuv} = I_{ov} \odot I_\delta \tag{20}$$

where I_{ov} is the value (V) component of I_o in HSV color space, and $\odot$ indicates pixel-wise multiplication.

Figure 24 shows ROC curves and the associated AUCs of three models. It is observed that both FasterRCNN and DoNN are robust to non-uniform illumination. Their corresponding feature maps are shown in Figure 25 and 9. The feature map $\mathcal{T}_1$ is generated as if there is no non-uniform illumination.

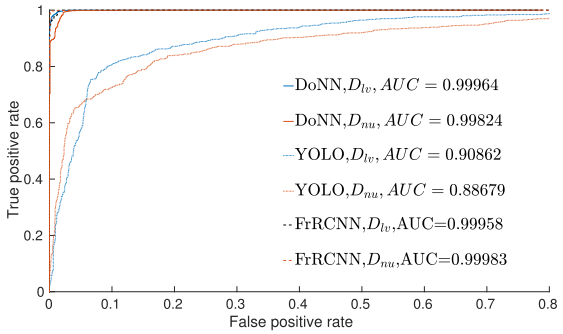


FIGURE 24. ROC curves of DoNN, YOLO, and Faster-RCNN obtained for undersea nonuniform illumination.

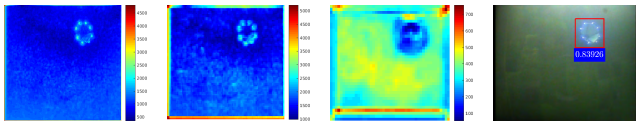


FIGURE 25. Feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i$, $i = 1, 2, 5$, and detection result of FasterRCNN obtained for non-uniform illumination. (a) $\mathcal{T}_1$. (b) $\mathcal{T}_2$. (c) $\mathcal{T}_5$. (d) Detection results.

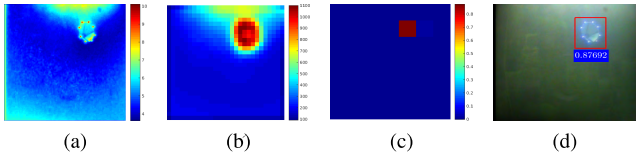


FIGURE 26. Feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i$, $i = 1, 5$, confidence map, and detection result of DoNN obtained for non-uniform illumination. (a) $\mathcal{T}_1$. (b) $\mathcal{T}_5$. (c) S. (d) Detection results.

Comparing $\mathcal{T}_5$ without and with non-uniform transformation in Figure 26 and 10, we observed that the maps $\mathcal{T}_5$ are almost identical. Therefore, non-uniform illumination does not affect the final prediction. The notation $\odot$ indicates pixel-wise multiplication.

10) COMPARISON OF PERFORMANCE OF DETECTION ALGORITHMS UNDER NOISY LUMINARIES

Similar to synthesis of mirror images, noisy luminaries which are similar to docking stations in terms of brightness and structure are also merged to every foreground image in D_{lv} by using image editing [44], forming a dataset D_{nlum}. Images with noisy luminaries are pinned three times at three random locations of the original images. We show the merging process of noisy luminaries in Figure 27.

We provide ROC curves and the corresponding AUCs of three models in Figure 28. The negative effect of noisy luminaries on three models is very tiny, although noisy luminaries are as bright as docking stations, showing strong robustness of CNN based methods to noisy luminaries. DoNN achieves an acceptable performance where AUC is 0.99846 in the presence of random synthetic noisy luminaries. It lags behind FasterRCNN whose AUC is 0.99928. Figure 29 shows feature

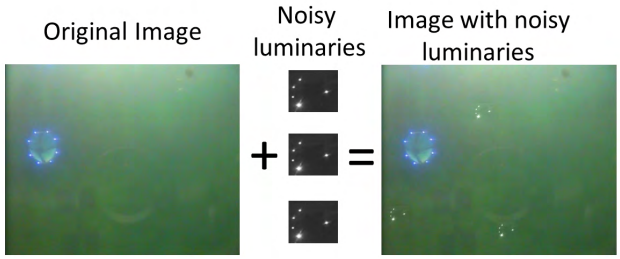


FIGURE 27. The merging process of synthetic noisy luminaries images. Three noisy luminaries are merged at random locations to the original image.

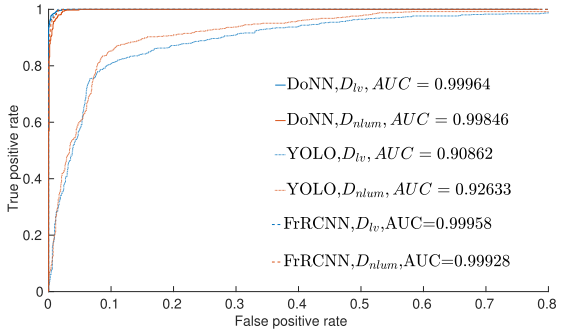


FIGURE 28. ROC curves of DoNN, and YOLO, Faster-RCNN computed for noisy luminaries.

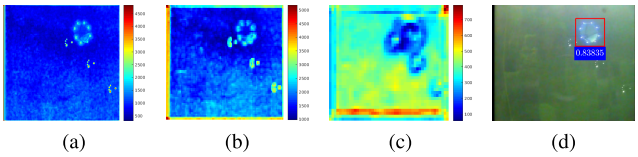
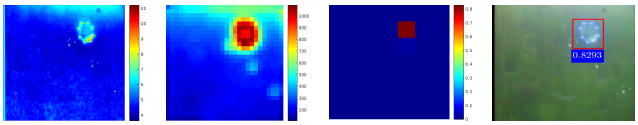


FIGURE 29. Feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i$, $i = 1, 2, 5$, and the detection result of FasterRCNN obtained for noisy luminaries. (a) $\mathcal{T}_1$. (b) $\mathcal{T}_2$. (c) $\mathcal{T}_5$. (d) Detection results.

maps and the detection result of a sample image predicted by FasterRCNN. The region of noisy luminaries is activated stronger than the docking station region in the feature map $\mathcal{T}_1$ of FasterRCNN. However, the region of the docking station overwhelms in the feature map $\mathcal{T}_5$ of FasterRCNN. Figure 30 shows feature maps, confidence maps and the detection result of a sample image predicted by DoNN. Activation of noisy luminaries is as strong as the docking station in the map $\mathcal{T}_1$ of DoNN, but vanishes in the map $\mathcal{T}_5$. We conjecture that it is the learned feature representation of spatial structural patterns that enables FasterRCNN and DoNN to avoid suffering from noisy luminaries.

11) COMPARISON OF PERFORMANCE OF DETECTION ALGORITHMS FOR NONLINEAR COMBINATION OF DEFORMATIONS

So far we have analyzed the performance of three detection methods for different deformations. Among them, hue shift and contrast shift affect the performance difference between DoNN and FasterRCNN mostly as shown above. Hence, the



(a) $\mathcal{T}_1$ (b) $\mathcal{T}_5$ (c) S (d) Detection results

FIGURE 30. Feature maps $\mathcal{T}_i$ computed at the i^{th} layer $Conv_i$, $i = 1, 5$, confidence map, and the detection result of DoNN obtained for noisy luminaries. (a) $\mathcal{T}_1$. (b) $\mathcal{T}_5$. (c) $\mathcal{T}_5$. (d) Detection results.

TABLE 24. AUCs of DoNN, FasterRCNN and YOLO computed for combination of hue and contrast shift.

Deformation	DoNN	FrRCNN	YOLO
$\lambda_h = 0.5, \gamma = 0.2$	**0.82383**	0.00000	0.00000
$\lambda_h = 0.5, \gamma = 0.4$	**0.91931**	0.00000	0.79135
$\lambda_h = 0.5, \gamma = 0.6$	**0.93689**	0.83695	0.85912
$\lambda_h = 0.5, \gamma = 0.8$	**0.95685**	0.80759	0.81419
$\lambda_h = 0.5, \gamma = 1.0$	**0.97405**	0.80774	0.86540
$\lambda_h = 0.5, \gamma = 1.2$	**0.93893**	0.86421	0.86557
$\lambda_h = 0.5, \gamma = 1.4$	**0.92623**	0.86660	0.86074
$\lambda_h = 0.5, \gamma = 1.6$	**0.92196**	0.87586	0.87612
$\lambda_h = 0.5, \gamma = 1.8$	**0.91799**	0.87833	0.89935
$\lambda_h = 0.5, \gamma = 2.0$	**0.91227**	0.86694	0.90924
$\lambda_h = 0.7, \gamma = 0.2$	**0.85802**	0.00000	0.72217
$\lambda_h = 0.7, \gamma = 0.4$	**0.94511**	0.00000	0.82594
$\lambda_h = 0.7, \gamma = 0.6$	**0.99237**	0.85504	0.86705
$\lambda_h = 0.7, \gamma = 0.8$	**0.99381**	0.92334	0.88413
$\lambda_h = 0.7, \gamma = 1.0$	**0.99824**	0.97628	0.89211
$\lambda_h = 0.7, \gamma = 1.2$	**0.99776**	0.96343	0.88061
$\lambda_h = 0.7, \gamma = 1.4$	**0.99740**	0.95966	0.89431
$\lambda_h = 0.7, \gamma = 1.6$	**0.99743**	0.94142	0.91749
$\lambda_h = 0.7, \gamma = 1.8$	**0.99530**	0.93769	0.93151
$\lambda_h = 0.7, \gamma = 2.0$	**0.99276**	0.93607	0.93536
$\lambda_h = 0.9, \gamma = 0.2$	0.83497	0.00000	**0.87129**
$\lambda_h = 0.9, \gamma = 0.4$	**0.99431**	0.92728	0.87093
$\lambda_h = 0.9, \gamma = 0.6$	**0.99829**	0.96821	0.91910
$\lambda_h = 0.9, \gamma = 0.8$	**0.99862**	0.99262	0.92616
$\lambda_h = 0.9, \gamma = 1.0$	**0.99956**	0.99888	0.93559
$\lambda_h = 0.9, \gamma = 1.2$	**0.99976**	0.99871	0.91672
$\lambda_h = 0.9, \gamma = 1.4$	**0.99930**	0.99902	0.92118
$\lambda_h = 0.9, \gamma = 1.6$	**0.99918**	0.99837	0.92003
$\lambda_h = 0.9, \gamma = 1.8$	0.99726	**0.99782**	0.92106
$\lambda_h = 0.9, \gamma = 2.0$	0.99658	**0.99678**	0.92229

performance of algorithms is explored for images that are deformed by combination of hue and contrast shift in this section. We sample λ_h and the γ in contrast deformation from {0.5, 0.7, 0.9} and {0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2}, respectively. The performance of DoNN, FasterRCNN and YOLO computed for combination of transformations is shown in Table 24. DoNN outperforms FasterRCNN by a large margin in all cases except two cases, and slightly lags behind for $\lambda_h = 0.9, \gamma = 1.8, 2.0$. Therefore we draw the conclusion that DoNN works better than FasterRCNN and YOLO in non-stationary underwater environments generally.

B. ANALYSIS OF THE POSE ESTIMATION ALGORITHM

In this section, we first show results on ground experiments to assess the accuracy and robustness of our pose estimation method to noise. Next, we examine the effectiveness of our pose estimation method by underwater docking experiments.

1) GROUND EXPERIMENTS

Since it is not feasible to obtain the underwater ground truth of the relative pose, we validate the accuracy and robustness of our pose estimation method to noise using ground experiments. Specifically in the experiments, we moved the camera around the docking station to capture its images with various pose from distance 3 m – 5 m with a stationary docking station, as shown in Figure 31a. Then, we manually labeled coordinates of the landmarks in 2D images. Finally, poses were computed by using our pose estimation method. In order to validate the robustness in presence of noise, we added different levels of Gaussian noise to the manually obtained coordinates of the landmarks. We provided mean estimation results averaged over 1000 trials for each noise level. The ground truth of orientation is obtained using an electronic compass which rigidly bounds together with the camera. In our ground experiments, half of eight landmarks are white LED, and the other half are blue LED. Same results can be obtained as the configuration of all blue LED, owing to manually labeled image points in ground experiments.

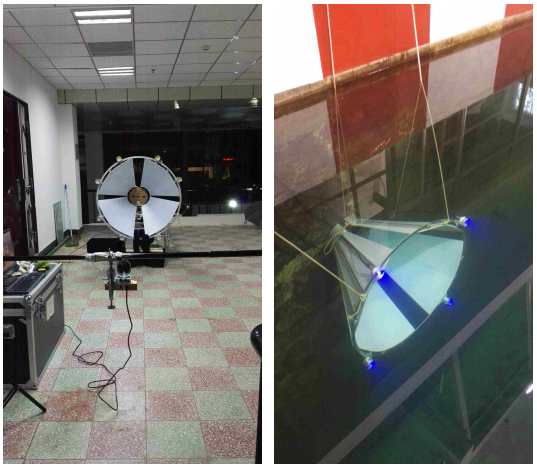
Table 25 shows the pose estimation results and their ground truth. Three levels of Gaussian noise are added: standard deviation $\sigma' = 0$, $\sigma' = 3$ and $\sigma' = 5$. Without adding any noise, the average error of predicted orientation and position are 1.970° and 5.927 mm respectively. As σ' increases to 3, orientation error and position error increase by 0.096 degree and 0.708 mm respectively. As noise level becomes $\sigma' = 5$, orientation error and position error increase by 0.383 degree and 3.505 mm, comparing to the case of $\sigma' = 0$. Therefore, we can draw the conclusion that our pose estimation algorithm is accurate and robust in the presence of noise.

2) UNDERWATER EXPERIMENTS

We analyzed our pose estimation method using experiments performed in our pool with 10 m in width, 15 m in length and 9 m in depth, where UDID was collected. The SIA-9 (see Section II) is employed for docking in this set of experiments. The docking station is mounted underwater in depth 2 m as shown in Figure 31b. The approaching speed of AUV is 0.5m/s on average. In order to eliminate other distractions, such as ambient light and false detection, we first shut down all ambient light in the experimental pool, remaining only landmarks emitting light. We launched the SIA-9 at initial locations out of the scope of critical angle so that mirror images are impossible. Meanwhile, the whole docking station is assured to observe in the captured images. Under these settings, a binarization-based detection method is employed for detection.

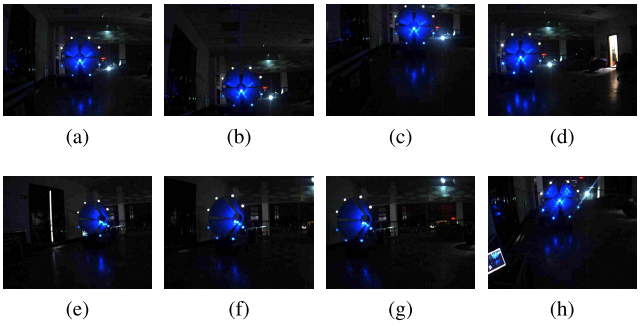
TABLE 25. Ground pose estimation performance. This table shows results corresponding to the figures given in Figure 32.

No.	Ground Truth		$\sigma' = 0$		$\sigma' = 3$		$\sigma' = 5$	
	Orien.(deg)	Pos.(mm)	Orien.(deg)	Pos.(mm)	Orien.(deg)	Pos.(mm)	Orien.(deg)	Pos.(mm)
(a)	Yaw:0.0 Pitch:4.5 Roll:-0.8	x: 99.2 y: -73.6 z: 3686.5	1.231 2.809 -1.273	10.366 -71.517 3687.955	0.960 1.932 -1.258	99.916 -72.216 3679.886	0.293 2.069 -1.244	98.862 -71.717 3666.954
(b)	Yaw:6.8 Pitch:-9.2 Roll:-1.3	x: 98.1 y: 839.8 z: 3598.1	2.115 -11.302 -2.142	96.559 837.126 3590.579	1.196 -11.911 -2.135	95.318 834.394 3598.303	1.127 -12.314 -2.128	95.288 832.257 3572.824
(c)	Yaw:2.7 Pitch:12.2 Roll:-1.2	x: 435.8 y: -1036.2 z: 3553.8	-3.714 16.827 -1.915	433.207 -1029.845 3538.244	-4.509 16.277 -1.841	431.473 -1028.301 3531.964	-4.443 16.198 -1.898	430.364 -1024.642 3518.536
(d)	Yaw:18.9 Pitch:-0.4 Roll:-2.3	x:-1212.8 y: -167.8 z: 3453.7	22.741 3.573 -2.042	-1213.576 -168.660 3460.211	22.503 3.067 -2.237	-1211.537 -168.734 3453.575	22.265 3.050 -2.183	-1206.448 -167.966 3438.054
(e)	Yaw:-37.0 Pitch:1.9 Roll:-2.9	x: 515.5 y: -422.3 z: 4098.4	-35.471 2.576 -2.571	513.549 -427.395 4088.068	-33.238 2.563 -2.624	518.459 -430.459 4115.437	-29.508 3.374 -2.817	522.176 -430.230 4120.387
(f)	Yaw:-28.3 Pitch:2.5 Roll:-1.2	x:-393.6 y: -374.2 z: 3307.9	-27.256 3.573 -1.363	-394.989 -374.406 3304.232	-26.977 3.472 -1.328	-395.335 -375.174 3309.443	-25.288 3.615 -1.402	-396.326 -377.715 3329.823
(g)	Yaw:-12.0 Pitch:2.6 Roll:-1.9	x:-1296.5 y: -327.8 z: 3102.0	-11.813 3.991 -2.317	-1284.283 -323.670 3073.298	-11.366 4.033 -2.326	-1285.865 -324.082 3078.568	-10.421 4.006 -2.331	-1290.430 -325.703 3090.632
(h)	Yaw:10.4 Pitch:22.8 Roll:-3.0	x:-59.8 y: -966.9 z: 3728.8	4.375 25.979 -3.420	-62.959 -972.440 3745.472	4.274 23.945 -3.508	-63.164 -973.179 3741.764	3.078 22.672 -3.665	-63.927 -971.425 3730.413
Average Error	N/A	N/A	1.970	5.927	2.066	6.715	2.353	9.432



(a) (b)

FIGURE 31. Infrastructures used for ground experiments and the underwater docking station located in water.



(a) (b) (c) (d)

(e) (f) (g) (h)

FIGURE 32. Ground test. Images were taken from different positions and orientations while keeping the docking station still.

The SIA-9 is launched at directly facing, left side and right side initial points in distance 10 m-15 m. Figure 33 shows a successful docking process. It is worth noting that intrinsic matrices used in the air and water are quite different from each other due to the change of medium. Thus, the camera was re-calibrated in the water before performing underwater experiments.

C. FIELD EXPERIMENTAL RESULTS OF UNDERWATER DOCKING AND RECHARGE

We conducted our field experiments of underwater docking and recharge in Qiandao Lake in China in 2017. We leveraged the acoustic sensor USBL for long-range navigation

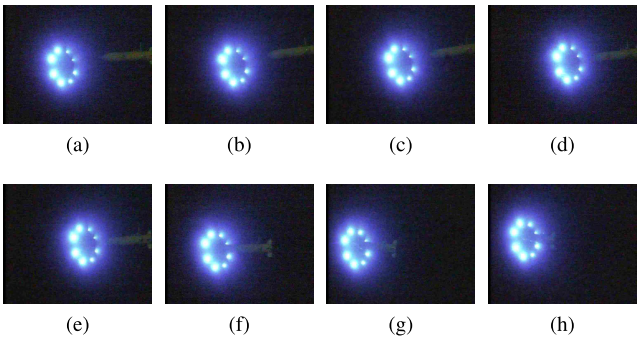


FIGURE 33. Final docking process used in our underwater experiments. (a) $t = 0$. (b) $t = 8$. (c) $t = 16$. (d) $t = 24$. (e) $t = 32$. (f) $t = 40$. (g) $t = 48$. (h) $t = 56$.

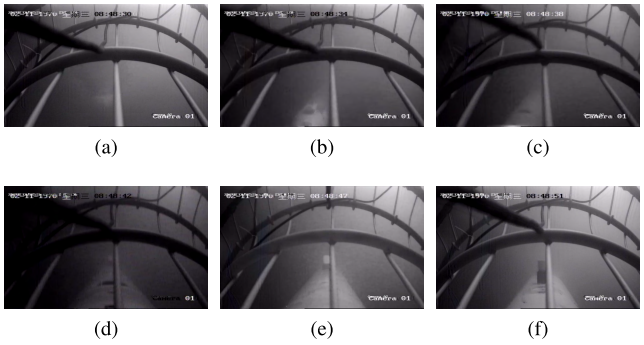


FIGURE 34. Final docking process captured by a forward-looking camera mounted on the docking station. (a) t. (b) t+4. (c) t+8. (d) t+12. (e) t+16. (f) t+20.

and our VBUD method to perform final stage of underwater docking with short-range precision in the experiments. We used the proposed SIA-3 AUV and its corresponding docking station in our field experiments. Their specifications are given in Section II. The docking station was fixed to an anchored experimental ship with underwater 7 m depth, while the station perturbed slightly with oscillation of the ship. Our SIA-3 started underwater docking at a location 50-100 m away from the docking station with random heading in different underwater docking experiments. It approached the docking station at a speed of 0.5 m/s. The SIA-3 only leveraged location information provided by the USBL for navigation when the distance between the SIA-3 and the docking station is greater than 15 m in z_c direction. The SIA-3 switched to using pose information supplied by our VBUD algorithm once our vision-based underwater docking algorithm detected the docking station at a distance of less than 15 m from the docking station in z_c direction. DoNN was implemented by using the framework Darknet [47] written using the C language, and the pose estimation module was also implemented in C in our VBUD algorithm. The average running time of the proposed framework on the on-board PC is 0.12 seconds per frame. Our VBUD algorithm updates the estimated pose in the control computer at a rate of 2Hz.

We performed four consecutive underwater docking experiments in the field. Three of them succeeded while one failed.

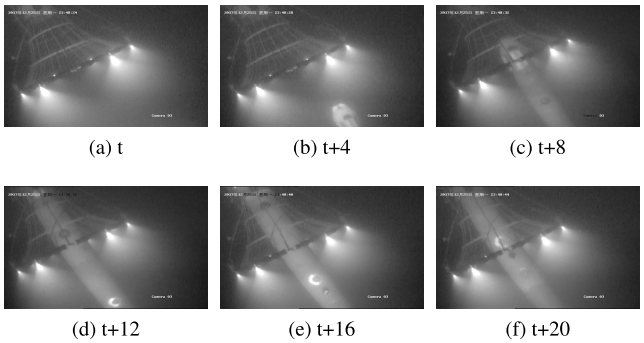


FIGURE 35. Final docking process captured by a downward-looking camera mounted on the docking station. (a) t. (b) t+4. (c) t+8. (d) t+12. (e) t+16. (f) t+20.

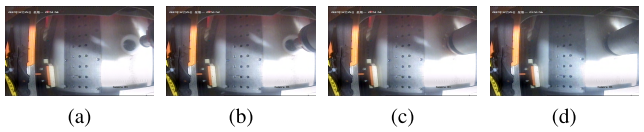


FIGURE 36. Recharge after underwater docking. (a) t+146. (b) t+148. (c) t+150. (d) t+152.

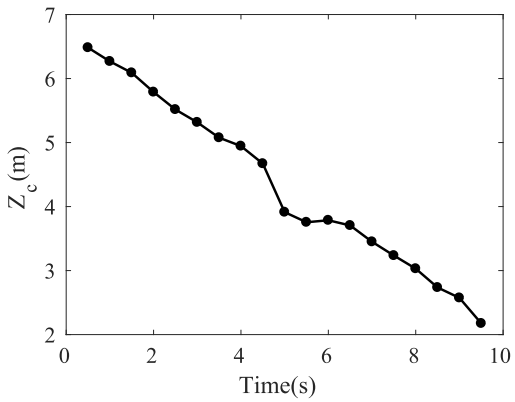


FIGURE 37. An analysis of the estimation of relative distance between the SIA-3 and the docking station in z_c direction using the proposed vision-based underwater docking process.

One of the successful docking processes and the failed docking process are analyzed in this section. We recorded the underwater docking and recharge course using a forward and a downward looking camera mounted on the docking station.

Figure 34 and 35 show a successful docking and recharge course captured using the downward and forward looking camera mounted on the docking station. The SIA-3 collides with the docking station slightly, and then enters into the docking station. We demonstrate our estimated relative pose between the SIA-3, and the docking station by showing estimated distance between them in z_c direction in Figure 37, since it's very difficult to obtain the ground truth of relative pose underwater. Our VBUD algorithm detected the docking station at a distance of 6.48 m for the first time, and entered into the blind area of the camera at a distance of 2.17 m. The distance estimated in z_c direction decreases as

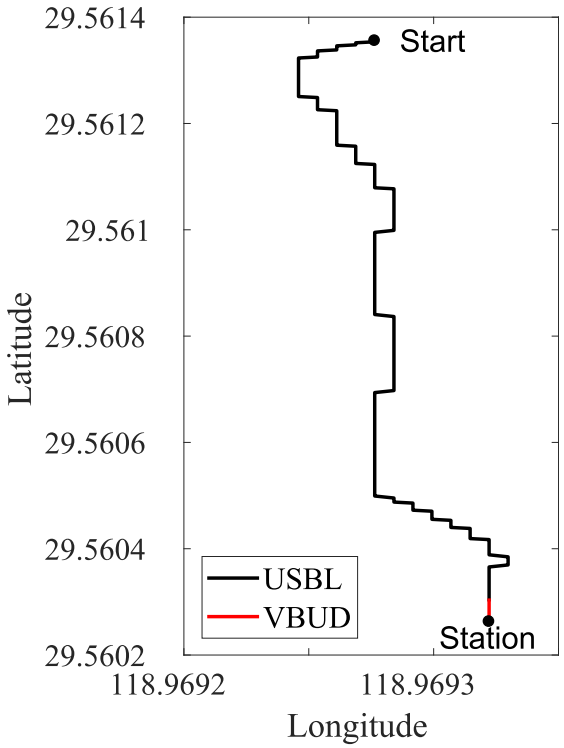


FIGURE 38. The trajectory of the SIA-3 employed in the underwater docking. The black trajectory is obtained by employment of the USBL. The red trajectory is obtained by employment of our VBUD algorithm.

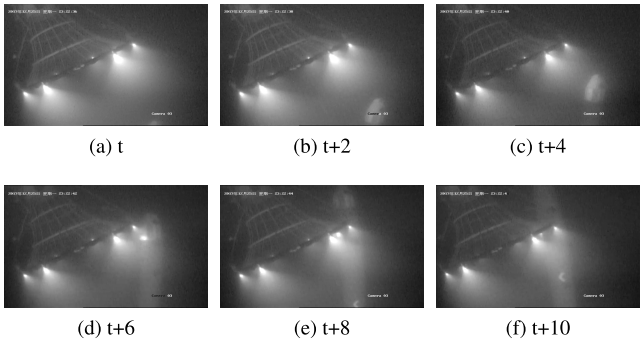


FIGURE 39. A failed underwater docking process. The SIA-3 failed to adjust its heading in this underwater docking experiment since it cannot adjust that much heading offset in such a short distance due to its large size and inherent mobility. (a) t. (b) t+2. (c) t+4. (d) t+6. (e) t+8. (f) t+10.

SIA-3 approaches to the docking station. Figure 38 shows the trajectory of the SIA-3 obtained using the USBL and the VBUD by dead reckoning. The SIA-3 started at a location which was approximately 120 m far from the docking station. It first used the USBL, and then switched to using the VBUD for vision based guiding at a distance of 6.48 m in z_c direction. A plug-in unit was used for underwater recharge process performed after the docking process, as shown in Figure 36. After docking and recharge, SIA-3 moved back and continued tasks. We provide the video of the whole process of underwater docking and recharge on the webpage http://vision.is.tohoku.ac.jp/~liushuang/a-vision-based-underwater-docking-system/video.

TABLE 26. List of abbreviations.

Abbreviation	Full Name
AUVs	Autonomous underwater vehicles.
AUC	Area under curve.
CNN	Convolutional neural network.
Conv	Convolutional layer.
DoNN	Docking neural network.
DVL	Doppler velocity log.
DLT	Direct linear transformation.
FC	Fully connected layers.
FN	False negative.
FP	False positive.
FPR	False positive rate.
FrRCNN	Faster RCNN.
PnP	Perspective-n-point.
RPnP	Robust perspective-n-point.
ROC	Receiver operating characteristic.
ROVs	Remotely operated vehicles.
RPN	Region proposal network.
SIA-9	The name of an AUV research platform.
SIA-3	The name of an AUV research platform.
TP	True positive.
TN	True negative.
TPR	True positive rate.
UDID	Underwater docking images dataset.
USBL	Ultra short baseline.
UUVs	Unmanned underwater vehicles.
VBUD	Vision based underwater docking.

Next, we analyze the failed underwater docking process. The whole process is shown in Figure 39. Our VBUD algorithm detected the docking station at a distance of 5.79 m, but with a relative heading estimated by $-31°$ for the first time. As shown in Figure 39, the SIA-3 failed to adjust its heading, since it cannot adjust that much heading offset in such a short distance due to its large size and inherent mobility. This is a problem that hinders performance of underwater docking for heavy AUVs like the SIA-3. In order to increase the visibility of landmarks to solve this problem, we will consider landmarks that combine laser light and LED in our future work.

V. CONCLUSION

We proposed a vision based underwater docking framework to perform final stage of an underwater docking process with short-range precision. The framework consists of (i) a detection module used for localization of docking stations in two dimensional (2D) images, and (ii) a pose estimation module used for estimation of the position and orientation between AUVs and docking stations from the docking stations detected in the images. For credible and robust detection of underwater docking stations, we proposed an algorithm, called DoNN. In order to analyze the performance of DoNN under various conditions, we provided a dataset

TABLE 27. List of frequently used symbols.

Symbol	Description
$\otimes$	Convolution operation.
$\odot$	Pixel-wise multiplication.
B	Number of candidate bounding-boxes predicted by each grid.
$\mathcal{B}_{i,b}$	The b^{th} Bounding-box of the i^{th} grid.
$\mathcal{B}$	Bounding-box.
D_{ltr}	Training set of UDID.
D_{lv}	Original test set of UDID.
$D_{bl\sigma}$	Blurring dataset.
D_{λ_h}, D_{λ_s}, D_{λ_v}	Datasets obtained after hue, saturation and value shift, respectively.
$D_{ct\gamma}$	The dataset obtained after contrast deformation.
D_{mirr}	The dataset containing mirror images.
D_{nu}	The non-uniform illumination dataset.
D_{nlum}	The dataset containing noisy luminaries.
G	Number of grids.
γ	The parameter of gamma transformation which is defined in equation (17).
k_θ	Skew coefficient.
K	The intrinsic matrix of the camera.
k_x, k_y	Scaling factors used in K.
$\lambda_{\mathcal{B}}(\boldsymbol{\theta})$	A hyperparameter defined in equation (3).
$\lambda_d(\boldsymbol{\theta})$	A hyperparameter defined in equation (3).
$\lambda_{\bar{d}}(\boldsymbol{\theta})$	A hyperparameter defined in equation (3).
λ_h, λ_s, λ_v	Hyperparameters defined in equation (15).
$l_{DoNN}(\boldsymbol{\theta})$	Loss function defined in equation (3).
$l_{\mathcal{B}}(\boldsymbol{\theta})$	Sub-loss function defined in equation (4).
$l_d(\boldsymbol{\theta})$	Sub-loss function defined in equation (5).
$l_{\bar{d}(\boldsymbol{\theta})}$	Sub-loss function defined in equation (6).
$S_{i,b}$	Confidence score.
S	Confidence map.
σ	Standard deviation of Gaussian filters.
σ'	Standard deviation of Gaussian noise.
$\mathcal{T}_i$	A feature map computed using equation (13) at the i^{th} convolution layer.
(u_0, v_0)	Principal point.
(u, v)	A coordinate in the image frame.
(x_c, y_c, z_c)	A coordinate in the camera frame.
(x_r, y_r, z_r)	A coordinate in the reference frame.

called UDID which was collected in our experimental pool. In the experiments, DoNN achieved 0.99964 performance in terms of AUC on the UDID, and we observed that DoNN was more robust to various deformations, such as blurring, color shift, contrast shift and mirror images, compared to the baseline models in average. A perspective-n-point algorithm called RPnP is integrated to our vision based underwater docking framework for pose estimation. We examined accuracy, speed, and robustness of the algorithm in the experimental analyses. The running time of pose estimation was 0.043 seconds per frame. In the ground experiments, the average error of position and orientation was 5.927 mm and 1.970°, respectively, when no artificial noise was employed. We observed that the average error of position and orientation was 9.432 mm and 2.353°, respectively, when strong artificial noise was added. Underwater docking experiments were performed to validate the effectiveness of pose estimation module in indoor pool. Field experiments conducted in the lake showed that our proposed framework can be used to detect docking stations, and estimate their relative pose efficiently and successfully.

APPENDIX
LIST OF ABBREVIATIONS

Abbreviations and symbols frequently used in this paper are listed in Table 26 and Table 27, respectively.

REFERENCES

[1] M. D. Feezor, F. Y. Sorrell, P. R. Blankinship, and J. G. Bellingham, "Autonomous underwater vehicle homing/docking via electromagnetic guidance," *IEEE J. Ocean. Eng.*, vol. 26, no. 4, pp. 515–521, Oct. 2001.

[2] Y. H. Hong, J. Y. Kim, J. H. Oh, P. M. Lee, B. H. Jeon, and K. H. Oh, "Development of the homing and docking algorithm for AUV," in *Proc. ISOPE*, Honolulu, HI, USA, 2003, pp. 205–212.

[3] J.-Y. Park, B.-H. Jun, P.-M. Lee, and J. Oh, "Experiments on vision guided docking of an autonomous underwater vehicle using one camera," *Ocean Eng.*, vol. 36, no. 1, pp. 48–61, Jan. 2009.

[4] C. Deltheil, L. Didier, E. Hospital, and D. P. Brutzman, "Simulating an optical guidance system for the recovery of an unmanned underwater vehicle," *IEEE J. Ocean. Eng.*, vol. 25, no. 4, pp. 568–574, Oct. 2000.

[5] T. Maki, R. Shiroku, Y. Sato, T. Matsuda, T. Sakamaki, and T. Ura, "Docking method for hovering type AUVs by acoustic and visual positioning," in *Proc. UT*, Tokyo, Japan, Mar. 2013, pp. 1–6.

[6] G. Bianco, A. Gallo, F. Bruno, and M. Muzzupappa, "A comparative analysis between active and passive techniques for underwater 3D reconstruction of close-range objects," *Sensors*, vol. 13, no. 8, pp. 11007–11031, Aug. 2013.

[7] A. Negre, C. Pradalier, and M. Dunbabin, "Robust vision-based underwater homing using self-similar landmarks," *J. Field Robot.*, vol. 25, nos. 6–7, pp. 360–377, Jun. 2008.

[8] Y. Li, Y. Jiang, J. Cao, B. Wang, and Y. Li, "Auv docking experiments based on vision positioning using two cameras," *Ocean Eng.*, vol. 110, pp. 163–173, Dec. 2015.

[9] J. W. Kaeli and H. Singh. *Illumination and Attenuation Correction Techniques for Underwater Robotic Optical Imaging Platforms*. Accessed: Nov. 2018. [Online]. Available: https://www.whoi.edu/cms/files/kaeli_joe14_InReview_183164.pdf

[10] C. D. Mobley, *Light and Water: Radiative Transfer in Natural Waters*. New York, NY, USA: Academic, 1994, pp. 86–100.

[11] M. Bryson, M. Johnson-Roberson, O. Pizarro, and S. B. Williams, "True color correction of autonomous underwater vehicle imagery," *J. Field Robot.*, vol. 33, no. 6, pp. 853–874, Sep. 2013.

[12] S. Cowen, S. Briest, and J. Dombrowski, "Underwater docking of autonomous undersea vehicles using optical terminal guidance," in *Proc. MTS/IEEE Conf. OCEANS*, vol. 2, Oct. 1997, pp. 1143–1147.
[13] S. Ghosh, R. Ray, S. R. K. Vadali, S. N. Shome, and S. Nandy, "Reliable pose estimation of underwater dock using single camera: A scene invariant approach," *Mach. Vis. Appl.*, vol. 27, no. 2, pp. 221–236, Feb. 2016.
[14] M. Myint, K. Yonemori, A. Yanou, K. N. Lwin, M. Minami, and S. Ishiyama, "Visual-based deep sea docking simulation of underwater vehicle using dual-eyes cameras with lighting adaptation," in *Proc. OCEANS*, Shanghai, China, Apr. 2016, pp. 1–8.
[15] M. R. Benjamin, H. Schmidt, P. M. Newman, and J. J. Leonard, "Nested autonomy for unmanned marine vehicles with MOOS-IvP," *J. Field Robot.*, vol. 27, no. 6, pp. 834–875, Nov. 2010.
[16] Q. Jia, H. Xu, and G. Chen, "The development of a MOOS-IvP-based control system for a small autonomous underwater vehicle," in *Proc. OCEANS*, Shanghai, Shanghai, China, Apr. 2016, pp. 1–5.
[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.
[18] T. N. Wiesel and D. H. Hubel, "Receptive fields of single neurones in the cat's striate cortex," *J. Physiol.*, vol. 148, no. 3, pp. 574–591, 1959.
[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, San Diego, CA, USA, 2015, pp. 1–14.
[20] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. CVPR*, Boston, MA, USA, Jun. 2015, pp. 1–9.
[21] G. L. Oliveira, C. Bollen, W. Burgard, and T. Brox, "Efficient and robust deep networks for semantic segmentation," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 472–491, Apr. 2018.
[22] M. Schwarz, A. Milan, A. S. Periyasamy, and S. Behnke, "RGB-D object detection and semantic segmentation for autonomous manipulation in clutter," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 51–437, Apr. 2018.
[23] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 421–436, Apr. 2018.
[24] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. ICML*, Haifa, Israel, 2010, pp. 807–814.
[25] Y.-T. Zhou and R. Chellappa, "Computation of optical flow using a neural network," in *Proc. ICNN*, San Diego, CA, USA, Jul. 1988, pp. 71–78.
[26] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: Towards accurate region proposal generation and joint object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 845–853.
[27] R. Girshick, "Fast R-CNN," in *Proc. IEEE ICCV*, Santiago, Chile, Dec. 2015, pp. 1440–1448.
[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[29] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.
[30] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jan. 2010.
[31] D. Bradley and G. Roth, "Adaptive thresholding using the integral image," *J. Graph. Tools*, vol. 12, no. 2, pp. 13–21, Jan. 2007.
[32] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proc. ACM-SIAM SODA*, New Orleans, LA, USA, Jan. 2007, pp. 1027–1035.
[33] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
[34] Y. I. Abdel-Aziz, H. M. Karara, and M. Hauck, "Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry," *Photogramm. Eng. Remote Sens.*, vol. 81, no. 2, pp. 103–107, Feb. 2015.
[35] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 22, no. 11, pp. 1330–1334, Dec. 2000.
[36] C.-P. Lu, G. D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 610–622, Jun. 2000.
[37] S. Li, C. Xu, and M. Xie, "A robust o(n) solution to the perspective-n-point problem," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 34, no. 7, pp. 1444–1450, Jul. 2012.
[38] D. Oberkampf, D. F. DeMenthon, and L. S. Davis, "Iterative pose estimation using coplanar feature points," *Comput. Vis. Image Understand*, vol. 63, no. 3, pp. 495–511, May 1996.
[39] G. Schweighofer and A. Pinz, "Robust pose estimation from a planar target," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 28, no. 12, pp. 2024–2030, Dec. 2006.
[40] H. Tanaka, Y. Sumi, and Y. Matsumoto, "A solution to pose ambiguity of visual markers using moire patterns," in *Proc. IEEE/RSJ Int. Conf. IROS*, Chicago, IL, USA, Sep. 2014, pp. 3129–3134.
[41] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, Zurich, Switzerland, 2014, pp. 818–833.
[42] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, Jul. 1997.
[43] L. Shapiro and G. C. Stockman, *Computer Vision*. Upper Saddle River, NJ, USA: Prentice-Hall, 2001, pp. 166–170.
[44] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, Jul. 2003.
[45] I. Kavasidis, S. Palazzo, R. Di Salvo, D. Giordano, and C. Spampinato, "An innovative Web-based collaborative platform for video annotation," *Multimedia Tools Appl.*, vol. 70, no. 1, pp. 413–432, May 2014.
[46] T. Tasdizen, E. Jurrus, and R. T. Whitaker, "Non-uniform illumination correction in transmission electron microscopy," in *Proc. MICCAI WS MIAAB*, New York, NY, USA, Sep. 2008, pp. 5–6.
[47] J. Redmon. *DarkNet: Open Source Neural Networks in C*. Accessed: Nov. 2018. [Online]. Available: http://pjreddie.com/darknet/

SHUANG LIU was born in Shenyang, China, in 1988. He received the B.S. and M.S. degrees in computer science and technology from the Shenyang University of Technology, in 2011 and 2014, respectively. He is currently pursuing the Ph.D. degree in mechatronic engineering with the University of Chinese Academy of Sciences, China.

His research interest includes autonomous underwater vehicles, underwater image processing, and computer vision.

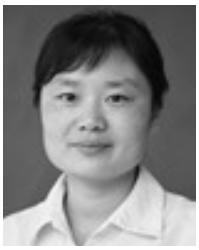
METE OZAY (M'09) received the Ph.D. degree in computer engineering from Middle East Technical University, Ankara, Turkey, in 2013. He is currently an Assistant Professor with the Graduate School of Information Sciences, Tohoku University, Sendai, Japan.

His current research interests include pure and applied mathematics, theoretical computer science, information theory, and machine learning.

TAKAYUKI OKATANI was born in Kainan, Japan, in 1971. He received the B.Sc., M.Sc., and Ph.D. degrees in mathematical engineering and information physics from the Graduate School of Engineering, Tokyo University, in 1994, 1996, and 1999, respectively. He is currently a Professor in computer vision with Tohoku University. He served as a Leader of the Infrastructure Management Robotics Team, RIKEN Center for Advanced Intelligence Project, in 2016.

He has more than 100 publications in refereed journals and conferences. These include about 20 papers in the top three conferences in computer vision. His research interests are in the field of computer vision and machine learning.

He is a member of the IEEE Computer Society, the Information Processing Society of Japan, The Institute of Electronics, Information and Communication Engineers, and The Society of Instrument and Control Engineers. He is an Advisory Board Member of the Japan Association for Medical Artificial Engineering and the Director of the Japan Deep Learning Association.



HONGLI XU was born in Faku, China, in 1978. She received the B.S. degree from the Taiyuan University of Science and Technology, in 2001, and the Ph.D. degree from the Shenyang Institute of Automation, Chinese Academy of Sciences, in 2008. She is currently a Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences.

Her research interests include autonomous underwater vehicles and robot collaboration.

KAI SUN was born in Ji'an, China, in 1979. He received the B.S. and M.S. degrees from Northeastern University, in 2005 and 2007, respectively, and the Ph.D. degree from the University of Chinese Academy of Sciences, in 2017. He is currently a Vice Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences.

His research interests include autonomous underwater vehicles and underwater observation network.

YANG LIN was born in Shenyang, China, in 1962. He received the B.S. and M.S. degrees from the Dalian University of Technology, in 1983 and 1988, respectively. He is currently a Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences. He is also an Editorial Board Member of the *Journal of Unmmaned Undersea Systems*.

His research interests include autonomous underwater vehicles.