Underwater Vision-Based Gesture Recognition

A Robustness Validation for Safe Human-Robot Interaction

Arturo Gomez Chavez, Andrea Ranieri, Davide Chiarella, and Andreas Birk

Underwater robotics requires very reliable and safe operations. This holds especially for missions in cooperation with divers who are - despite the significant advancements of marine robotics in recent years - still essential for many underwater operations. Possible application cases of underwater human-robot collaboration include marine science, archeology, oil- and gas production (OGP), handling of unexploded ordnance (UXO), e.g., from WWII ammunition dumped in the seas, or inspection and maintenance of marine infrastructure like pipelines, harbors, or renewable energy installations - to name just a few examples.

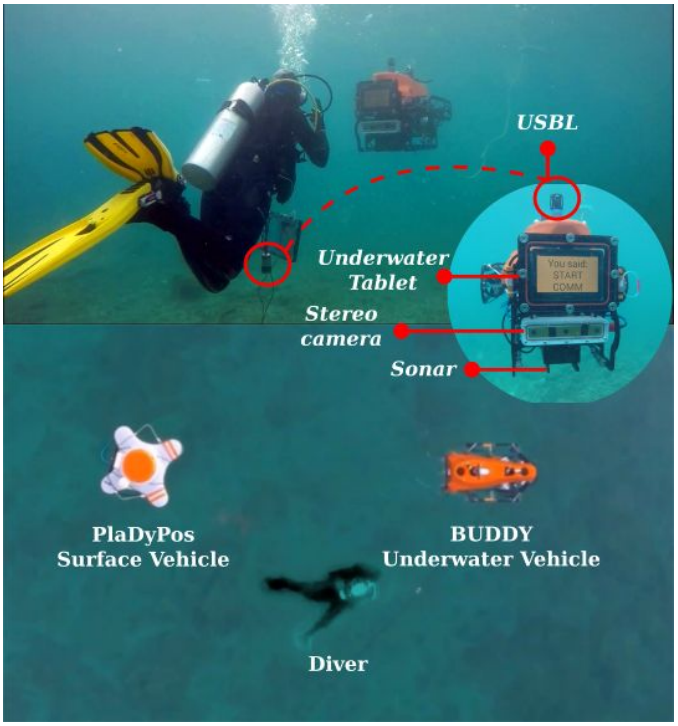


Figure 1: The CADDY system for assistance in diver missions. (Right) The Buddy-AUV is equipped with a Blueprint Subsea X150 USBL, a Underwater Tablet, a BumbleBeeXB3 Stereo Camera, and an ARIS 3000 Imaging Sonar for diver tracking, monitoring and communication. (Top) Diver gesturing a command. (Bottom) Aerial view of the system with a PladyPos surface vehicle for global positioning.

We present a fully integrated approach to Underwater Human Robot Interaction (U-HRI) in form of a front-end for gesture recognition combined with a back-end with a full language interpreter. The gesture-based language is derived from the existing standard gestures for communication between human divers. It enables a diver to issue single commands as well as complex mission specifications to an Autonomous Underwater Vehicle (AUV) as demonstrated in several field trials.

The gesture recognition is an essential component of the overall approach. It requires high reliability under the challenging conditions of the underwater domain. There is especially a high amount of variation in visual data due to various effects in the underwater image formation. We hence investigate in this article different Machine Learning (ML) methods for robust diver gesture recognition. This includes a classical ML approach and four state-of-the-art Deep Learning (DL) methods. Furthermore, we introduce a physically realistic way to use range information for adding underwater haze to produce meaningful additional data from existing real-world data. This can be of interest for creating evaluation data for underwater perception in general or to produce additional training data for ML-based approaches.

I. Related Work

Given the importance of cameras for underwater systems, especially for near-field perception, computer vision is predominantly used for U-HRI. Alternatives are acoustic approaches with pingers or sonars as well as the use of dedicated devices like underwater tablets [1], [2]. The first step towards U-HRI is the detection and tracking of one or multiple divers [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. Given relative localization, different protocols for interaction can be studied and trained, among others also in computer simulation [14].

Relative motions between divers and robots can already be used for a basic, non-verbal form of communication [15], but more capable forms of communication - in terms of expressiveness and reliability - are needed to enable real U-HRI for collaborative missions. Work in that direction is described in [16], where artificial fiducial markers are used that are then interpreted by the robot using grammatical rules. While cards with artificial markers ease the challenges of underwater vision, there are disadvantages like the number of cards that the diver must carry and the effort to handle them.

Gestures are a more natural basis for underwater communication: (a) they are already extensively used by divers and (b) there are among others limitations of water as a medium, e.g., it is impossible to use voice recognition.

Early research on the use of gestures for U-HRI is described in [17], where waving gestures are recognized by differential imaging with a spectral registration method in form of the improved Fourier Mellin Invariant (iFMI). Based on that, trajectories of hand motions are recognized with a Finite State Machine (FSM). The experiments in [17] are done in a pool.

An imaging sonar, also known as acoustic camera, is used in [18] for gesture recognition. Preprocessing stages with cascade classifiers and shape processing are combined with

three different classification approaches, namely a convex hull method, Support Vector Machines (SVM), and the fusion of both. Experiments are conducted in a pool and during field trials with divers in the context of the EU-project "Cognitive autonomous diving buddy (CADDY)". The selection of device parameters within a mission is a known challenge for this type of sensor, which is also reported in [18].

The main type of sensor for U-HRI in CADDY is therefore a (stereo-)camera. To process the visual data, a modification of Nearest Class Mean Forests (NCMF) in form of a Multi-Descriptor extension (MD-NCMF) is introduced. MD-NCMF is used both for diver detection and tracking [8] as well as for the classification of diver gestures [19]. As the name suggests, MD-NCMF is designed to exploit different types of descriptors to achieve high robustness under the challenging conditions of underwater visibility. To this end, MD-NCMF builds on NCMF, which partitions the sample space by comparing the distances between class means instead of comparing values at each feature dimension as in more traditional Random Forests approaches. Therefore, MD-NCMF can treat each feature-object pair as a new class, e.g., SURF-object1, SIFTobject2, SURF-object2, SIFT-background, etc., and MD-NCMF can examine which one provides the best partition of the sample set.

Based on the MD-NCMF gesture recognition [19], a machine interpreter [20] with a phrase parser, syntax checker, and command dispatcher linked to the mission control allows the use of a very expressive language for U-HRI [21]. This Caddian language is based on a context-free grammar, that allows the diver to specify missions with a sequence of tasks. The syntax checker is implemented as a FSM that gives constant feedback to the diver and that allows in situ corrections. The gesture recognition front-end and the machine interpreter back-end are reported in field tests to not only be robust but also useful in complex missions with professional divers [22], [23].

A full language for U-HRI is also presented in [24]. It is syntactically a bit simpler than Caddian as the FSM in its interpreter is restricted to only one possible transition from state to state, i.e., gesture to gesture, to avoid ambiguities. The gesture recognition front-end in [24] is based on deep learning models. More precisely, Single Shot Detector (SSD) [25] and Faster Region-based Convolutional Neural Networks (Faster R-CNN) [26] are investigated, which achieve above 90% accuracy when being trained with a 50K dataset.

It is assumed in [24] that the diver wears no gloves; this enables the use of skin detection and image contour estimation. In practice, professional divers tend to always wear gloves - both for protection and to avoid heat loss. For the MD-NCMF gesture recognition [19] mentioned above, regular diving gloves are augmented with colored stripes to provide some detectable contrast. First results towards a classification under a wide range of conditions including divers with and without gloves are presented in [27]. Building upon a DL-based approach dubbed SCUBANet to recognize diver body parts [28], MobileNetV2 [29] is trained to recognize 25 image classes using finger count and palm direction - though the authors also state that a significant portion of these classes are unused in most gestures [27].

II. Underwater human-robot interaction with gesture based communication

Our gesture based communication for U-HRI consists of a gesture recognition front-end and an interpreter back-end. In this article, different options for the front-end are investigated, which are described in the following Sec.II-A. A short overview of the actual language and the interpreter back-end is then given below in Sec.II-B. An example from a field trial in Sec.II-C illustrates the use of the complete system and the challenges that occur in practice and that motivate the investigation of different DL-methods.

ML in general and DL in particular typically require high amounts of data for training and evaluation. A physically realistic way to use range information for adding underwater haze is hence introduced in Sec.III. This is used to add artificial degradations to existing real-world images from field trials to produce additional data, which is useful to cover the high amount of variability in the underwater domain without the need of many costly field campaigns.

A. Gesture Detection and Classification

1) MD-NCMF as Classical ML Approach: MD-NCMF is a Multi-Descriptor (MD) extension of Nearest Class Mean Forests (NCMF), which is used for both diver detection/tracking as well as for the classification of diver gestures [8], [19] (Fig. 2). This variant of Random Forests aggregates multiple descriptors (SIFT, SURF, ORB, HoG, etc.) that encode different representations of the objects of interest as we observed that each of these descriptors is robust to different types of underwater image degradations. MD-NCMF can be considered to be a classical ML approach, which forms a comparison basis for the different DL methods described below.

For the hand detection as first step, both 2D monocular images and 2.5D stereo disparity are used. The 2.5D disparity maps are segmented based on distance and density. This provides a reliable hand detection in many cases. However, it fails on texture-rich interferences close to the stereo-camera, e.g., due to air bubbles. Therefore, 2D cascade classifiers are used in a second process running in parallel to filter out the false positive regions. The resulting region proposals, i.e., object candidates, serve as input to the actual classifier. MD-NCMF then filters out further false positives that may still exist and it maps the hand regions to the gestures of the Caddian language described below.

2) Deep Learning (DL) Approaches: State-of-the-art deep models for visual object detection and classification often follow three meta-architectures: Single Shot Detector (SSD) [25], Faster Region-based Convolutional Neural Network (Faster R-CNN) [26], and Region-based Fully Convolutional Neural Network (R-FCN). SSD models offer fast computation speeds since they perform object detection and classification in one single pass of the network. They are hence often preferred for embedded systems. Faster R-CNN has two stages, which are conceptually similar to the described classical ML approach

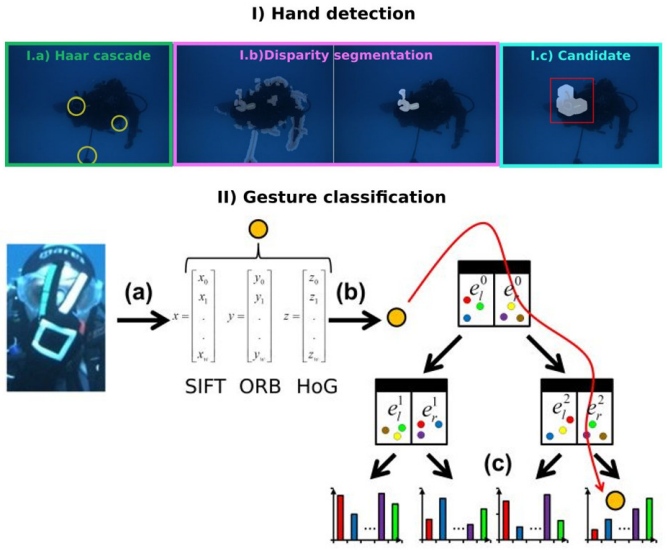


Figure 2: **Hand detection**. Possible regions are detected by two processes running in parallel: (I.a) a Haar cascade model, and (II.b) a disparity map that is thresholded by distance and morphologically transformed to reduce noise. (I.c) A cross-check between the two methods generates the final hand image candidates. **Gesture classification** using a Multi-Descriptor NCM tree (MD-NCM): Each class centroid - marked by a colored dot - traverses a path through the decision tree (II.a). The image is encoded into different types of feature vectors $\vec{x}$ $\vec{y}$ $\vec{z}$ (II.b). The sample passes down the tree following the closest centroid as aggregated similarity measure (II.c).

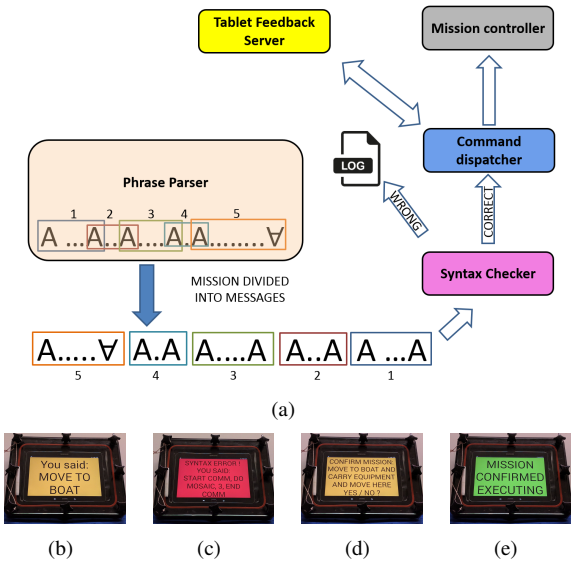


Figure 3: (a) Mission aggregation from single gestures/commands. (b–e) Types of feedback given to the diver through an underwater tablet on the AUV.

(Sec. II-A1): a region proposal network generates candidates for object regions and a classifier then verifies and refines the proposals. The R-FCN architecture is a mixture between the previous two meta-architectures. It shares features learned in the initial layers between the region proposal and the actual classifier network.

Visual model	Feature extractor	Software Library	References
FCN-CNN	ResNet-50	Fast.ai/Pytorch	[30]
SSD	MobileNets	Tensorflow	[25], [31]
Faster R-CNN	ResNet-101	Tensorflow	[26], [30]
Deformable Faster R-CNN	[32]	MXNet	[26], [32]

Table I: Overview of the four Deep Learning models and the pre-trained feature extractors.

The DL models are used with pre-trained feature extractors (Table I). A Fully Connected Network like ResNet [30] can be consider the most straightforward approach since it only requires a label per image, no region candidate, which ultimately satisfies our system's requirements. The SSD [25] and Faster R-CNN [26] differ mostly in their architectures among the considered DL methods; the former is tailored towards fast computation when using the MobileNet feature extractor [31]. A Deformable ConvNet [32] allows region proposals with non-uniform boundaries by using a flexible sampling grid on the image. Thus, it is no longer assumed that the object geometry is fixed, which can be beneficial for detecting 6-DoF hands of a free floating diver.

B. The Caddian Language and its Interpretation

The gestures form a language for U-HRI called Caddian [20], which is derived from the routine communication of divers. The Caddian syntax defines boundaries to understand complex commands, i.e., sequences of gestures, which can also be aggregated to form missions composed of several tasks. Two gestures to start a command and to end a communication, denoted as A and $\forall$, are used for this purpose. Commands are sequences of individual gestures delimited by (A, A) that represent a single task. A practical example from field trials dealing is the command "Take a photo at 3 meters altitude". Missions consist of aggregated commands that are delimited by $(A, \forall)$. An example for a mission used in practice is "Take a photo, go to the boat and carry the equipment back".

To handle very frequent tasks or emergencies, there is the special *Slang* group of gestures. They have higher priority and a simpler syntax. Examples include a gesture to instruct the AUV to take a photo at the current location, i.e., without specifying any parameters, or a gesture to signal that the diver is out of air, which triggers emergency response protocols on the AUV and the surface vehicle it is connected to.

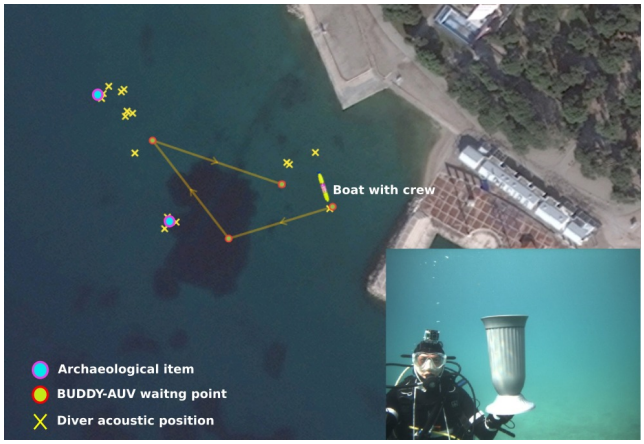
As illustrated in Fig. 3(a), the *Phrase Parser* constantly saves the recognized gestures until it detects one of delimiter pairs (A, A), $(A, \forall)$. It then sends the gesture set to the *Syntax Checker* for validation. If the command is syntactically correct, it is passed to the *Command Dispatcher* where it is saved until a complete mission is received. After the diver confirms, the commands are passed to the *Mission Controller* for execution.

Despite the syntax validation, gestures can be misclassified, i.e., a message can have the correct structure but it represents an infeasible or undesired action. Therefore, the system integrates the diver in a human-in-the-loop approach to identify and correct possible errors as quickly as possible. Five types of feedback are provided to the diver at different times during the communication process through an underwater tablet on the AUV (see Fig. 3):

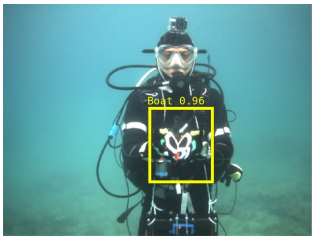
1) Single gesture – Every time a gesture is recognized, the tablet displays the classification label given to that gesture.

2) Syntax error – Whenever a *phrase/command* is detected and analysed by the *Syntax Checker*, an error is displayed if the Caddian grammatical rules are not followed. The received message is shown to the diver for his/her analysis and the communication is reset.
3) Mission confirmation request – When the diver ends communication, the system displays the complete mission and it waits for a confirmation gesture or a gesture to abort.
4) Mission status – When the AUV is executing a mission, the current status of the mission is displayed, e.g., `CARRYING EQUIPMENT`, `MISSION COMPLETED`.

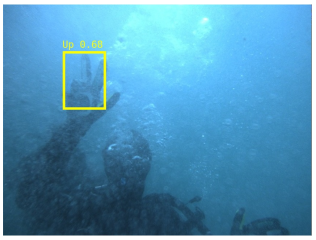
C. Example Use-Case and Challenges



(a) Mission layout and archaeological item to be retrieved.



(b) Gesture `boat` recognized



(c) Gesture `photo` not recognized

Figure 4: A field trial emulating an archaeological underwater mission in Biograd na Moru, Croatia. One task includes the transport of an object found by the diver, here a mock-up amphora, by the AUV to a boat.

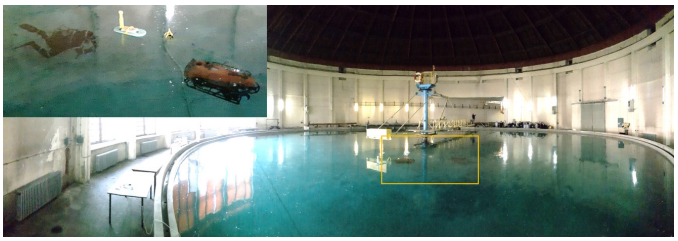


Figure 5: Trials at the the Brodarski Institute in Zagreb, Croatia.

The following example shortly illustrates a typical use-case and the challenges. The related field trial emulates an archaeological mission. It was designed in cooperation with the *Diver Alert Network Europe (DAN Europe)* to take safety and ergonomic aspects in the evaluation into account. The test was done in Biograd na Moru, Croatia in 2016. Fig. 4 depicts the overall mission. The field trial tested many system functionalities and not only the gesture-based communication. Therefore, only the `photo` and `boat` command happened to be used, along with the gestures for the start and the end of communication. More commands and also full missions were tested in other trials, among others in 2017 at the Brodarski Institute in Zagreb, Croatia (Fig.5). But the Biograd field trial nicely illustrates the challenges: there is for example reduced visibility compared to experiments in a pool, and the sea was rough on several days causing motion blur as neither the divers nor the AUV could keep still on the spot.

Fig. 4(c) shows an example where the `photo` command is incorrectly classified by the gesture recognition front-end. Note that due to the interpreter back-end, the error was captured and remedied, which indicates the importance of a complete human-robot interaction and collaboration framework to ensure correct operation and diver safety. Also, accuracy is boosted in the real system by fusing the classification results of multiple consecutive frames. Nonetheless, the higher the accuracy in the gesture recognition, the more convenient is the system to use. Even more importantly, it is of high interest to check that the recognition does not fail in previously untested environment conditions, i.e., that the accuracy does not suddenly drop to unexpectedly low rates. This evaluation across a wide range of environment conditions is a non-trivial chore for underwater vision in general.

III. Physically Realistic Underwater Image Degradation

Underwater image formation is influenced by many factors [33]. Given in addition the complexity of underwater field trials, it can be challenging to provide sufficient data for the evaluation of underwater vision methods. This holds especially when ML and in particular when DL is employed. There, clear and immutable design assumptions can not be assumed. In addition, there is the need for large amounts of training data for ML and especially DL algorithms. One option is to produce synthetic images in simulations [34], [35]. An other option is the use of Generative Adversarial Networks (GAN) as successfully demonstrated in the context of underwater image enhancement [36], [37], [38].

We use here insights from underwater image formation for artificial image degradation to produce additional data from real-world data. The existing real-world data covers the relevant scenarios, i.e., underwater scenes with divers carrying out realistic tasks including situations with the use of hand gestures. The artificial degeneration allows to evaluate the different options for the gesture recognition under a very wide range of possible environment conditions, which are unfeasible to cover in this broadness with real field trials. In addition, the artificial degeneration allows an evaluation under controllable conditions. Among others, we introduce in Sec.III-B a method based on depth information that allows a physically very realistic reduction of visibility conditions. The image degradation can also be used in other applications

of underwater vision. This also includes the production of additional training data for ML methods including especially DL methods.

Several different image degradation methods are considered. The first group of transformations, named *pixel-based perturbations*, only requires information from a single monocular image and it transforms only pixel values, i.e., all operations are constrained to the image domain. The second group of *geometry-contextual perturbations* uses the 3D scene geometry information obtained from stereo imagery [39] to compute the depth relative to the camera and, in turn, to render a more detailed simulation of underwater light backscattering effects. Fig. 6 shows examples of each type of distortion. The code for the degradations is available at https://github.com/arturokkboss33/caddy-underwater-diver-classification.

A. Pixel-Based Perturbations

1) Gaussian blur: The image is blurred to approximate effects caused by moving objects, sediment clouds, material on the lens/housing, misalignment of the camera with respect to the housing window, or a wrong focus caused by light forward-scattering. This is done using a Gaussian kernel with standard deviation σ and size k_s pairs: $\{(1.5, 9)(3, 17)\}$.

2) Brightness shift: For shallow water operations (depth$<$ $15\,\mathrm{m}$), the ambient light can drastically change the brightness of the image depending on weather conditions and on the time of the day. To simulate this, a scaling factor b is applied to each image channel, respecting saturation values, with $b = \{0.5, 2\}$.

3) White Balance: White balance is considered as it can lead to unexpected image artifacts. White balancing methods are typically based on the assumption that there is a minimum range of colors in the scene including neutral (white) colors. But when there are large regions with uniform color in the scene (water medium), this can shift the color correction to more blueish or reddish colors. Thus, if the white balance is not properly configured, respectively if a standard in-air method is used, it can degrade the quality of the image. To reflect this, a gray-world (GW) white balance is applied that assumes that the average of all channels should result in a gray image. It requires a saturation threshold $t_{GW} = 0.7$. All normalized pixels above this value are not used during the color correction process. Another method, denoted as simple white balance (SWB), just stretches each input channel to generate similar ranges for each channel. It uses a threshold $t_{SWB} = 5\%$ to ignore the according top and bottom percent of pixels.

4) Underwater Alpha Blend: The image I is blended with a background image of uniform color H that represents simple underwater haze effects. This image operation known as alpha blend is defined as $A = H \times \alpha + I \times (1 - \alpha)$. The blending coefficients used here are $\alpha = \{0.25, 0.5\}$. Typically, a gray color for H is used based on the color of fog on land. However, to emulate underwater haze with higher fidelity, the Jerlov water types [33] are used with their associated light downwelling and back-scattering attenuation factors to tune H to a more realistic color. For our experiments, ambient light at depth d of $10\,\mathrm{m}$ is assumed. Jerlov water types $w = II, 1C$ are considered, i.e., murky oceanic water and coastal water with low amounts of sediments. Based on our experience, they offer challenging visibility conditions but they are within the operational range for divers. With these values of α, w and d, values for H are computed based on the Jerlov classification.

5) Image/Video Compression: Underwater robots are employed in practice in a wide-range of applications, respectively for a wide range of different tasks even within a particular application. Hence, different bandwidth values or CPU resources are typically allocated to each system component depending on the mission, respectively a task within the mission. It is hence common practice that compression algorithms are applied to the images, respectively video frames during real missions to free resources for other processes as well as for data storage, respectively for data transmission in the case of Remotely Operated Vehicles (ROV). Often, *motion JPEG* is used to optimize for coding speed and frame-by-frame quality over bitrate. To study image degradation effects, compression quality values of $q = \{60, 20\}$ are used here.

B. Geometry-Contextual Perturbations

In underwater environments, ambient light attenuates exponentially with depth d and even further with the distance z between the target object (here, the diver) and the observer (here, the camera on the AUV). However, the attenuation factor $K(d)$ due to the depth can typically be ignored because the attenuation factor $\beta(\lambda, z)$ due to the distance z and the wavelength λ is 2 to 5 times greater [33]. Note that we can typically assume an observer-object viewing direction of approximately $\theta = 90°$ (Fig. 1).

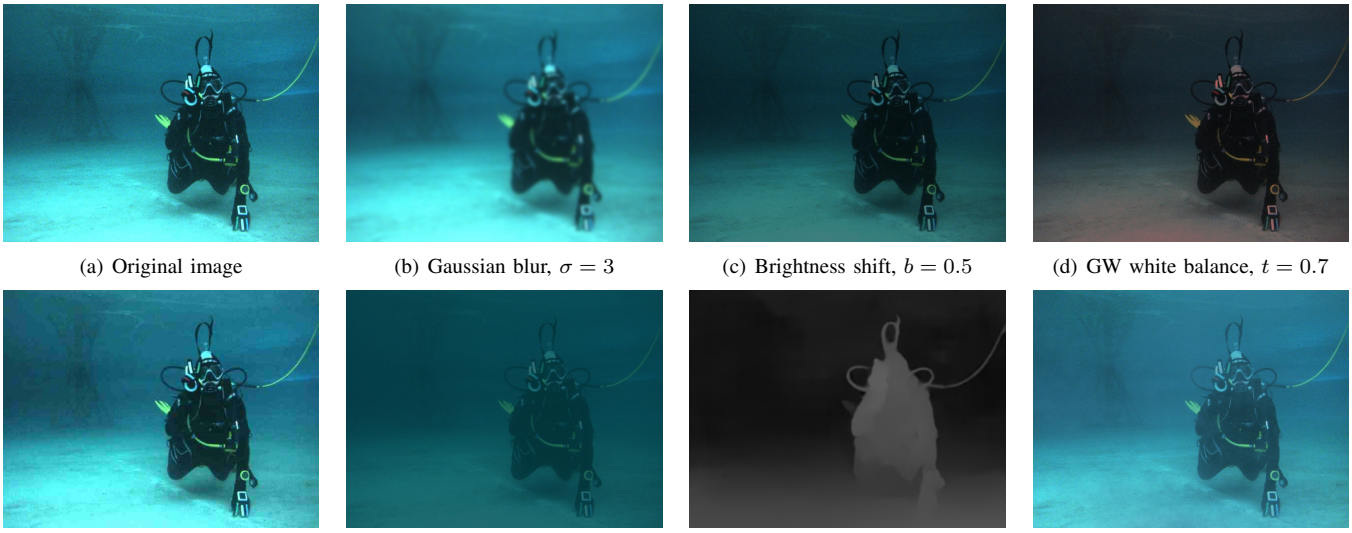
To obtain z or the "depth" relative to the image, DispnetC [40] is used. It is a 100% dense disparity estimator with $\approx 4\%$ error in the KITTI Stereo 2015 benchmark.This accuracy is more than enough for our purposes, and the method has proven to perform well in underwater scenarios [41]. Nonetheless, the estimated z is refined through a bilateral filter to keep the image edge consistency (Fig. 6(g)). Based on this value, the geometry-contextual image transformations presented in the following sections can be applied.

1) Underwater Haze: A popular haze model used in terrestrial robotics is based on the following equation:

$$I(x) = J(x)t(x) + B(1 - t(x)) \tag{1}$$

$$t(x) = e^{-\beta(\lambda)z(x)} \tag{2}$$

where $I(x)$ is the image received by the camera sensor, $J(x)$ is the original image (scene radiance), which is exponentially attenuated by the transmission matrix $t(x)$ at every pixel x as range (distance to object) increases and depending on the wavelength λ. B is the ambient light. As mentioned, $z(x)$ is computed here by DispnetC. It is refined with a bilateral and a Gaussian filter to avoid discontinuity effects. But a physically realistic haze model is more complex in the underwater case [33]. In summary, a different transmission matrix is needed for each $J(x)$ and B.



(a) Original image (b) Gaussian blur, $\sigma = 3$ (c) Brightness shift, $b = 0.5$ (d) GW white balance, $t = 0.7$

(e) JPEG compression, $q = 20\%$ (f) Alpha blend, $w = 1C, \alpha = 0.25$ (g) Depth map from stereo (h) Underwater haze

Figure 6: Examples of underwater image perturbations: (a) Original image (b–f) Pixel-based (g–h) Geometry-contextual.

$$I(x) = J(x)t_J(x) + B(1 - t_B(x)) \quad (3)$$

$$t_J(x) = e^{-\beta(\lambda)z(x)} \quad (4)$$

$$t_B(x) = e^{-\beta(w,d)z(x)} \quad (5)$$

The underwater haze model from Eq. 3 is hence combined with the range map $z(x)$. This allows to apply systematic and controlled image degradations to the real world data in form of physically realistic underwater haze.

For the values of B and its corresponding $t_B(x)$, the same values as for the underwater alpha blend $\alpha = 0.25$ are used (Sec. III-A). For $t_J(x)$, attenuation coefficients $\beta(\lambda)$ are chosen to allow visibility in a distance of approximately $10\,\mathrm{m}$, which can be considered to be a reasonable maximum operational distance in underwater human-robot interaction. In terms of Jerlov water types, this corresponds to $\beta = [0.5, 0.15, 0.90]$ for the red, green and blue channel respectively.

IV. Experiments and Results

A. Dataset and Setup

The presented approach to U-HRI originated within the EU-project "Cognitive autonomous diving buddy (CADDY)". Major efforts were devoted in this project to the collection of data including experiments on the use of underwater gestures. Concretely, data was recorded in open sea as well as in indoor and outdoor pools in three different locations, namely in Biograd na Moru (Croatia), at the Brodarski Institute in Zagreb (Croatia), and in Genova (Italy). The data is divided into 8 scenarios representing different diver missions and field experiments. The scenarios named *Biograd-A*, *Biograd-B*, and *Genova-A* represent trials that were mainly organized for data collection; they hence feature a high number of samples. The other scenarios *Biograd-C, Brodarski-A* to *D* cover experimental or real diver missions. A detailed discussion of the number of samples and of the environmental conditions of each scenario is provided in [39]. For the evaluation of the different ML-methods here, they are trained according to the partition of the data shown in Table II.

	Model A	**Model B**	**Model C**	**Model F**
Training Sets	Biograd A,B	Genova A	Brodarski A,C	All scenarios
Samples mean	338	415	222	1156
Samples median	151	294	206	792

Table II: Partioning of the data for training (samples mean and median are provided per class).

Each method described in Sec. II-A has four Model X versions. The partition is made to gather samples with similar environmental conditions (location, light, etc.) and to observe how the methods perform against unseen types of data. Samples from *Biograd C*, *Brodarski B & D* are only used as test sets.

The complete dataset contains 18,478 images (9,239 stereo pairs) that represent 16 gesture classes. A split of 80%-20% for training and validation sets is used for each model, except for Model F. This splitting criterion is applied to each gesture class. The test sets comprise all scenarios that are not included in the training. All classifiers with the exception of Model C have samples of all gestures. Model C, being trained only on the *Brodarski A* and *C* scenarios, is trained and tested only on 9 gestures. Then, Model F (F stands for "full") is trained with samples from all scenarios according to a standard split of 70%-20%-10% following the data distribution per scenario. As mentioned, the data and its distribution is described in detail in [39].

B. Setup of the ML-Methods

Following settings are used for the ML-methods that are evaluated here as possible approaches for underwater gesture recognition.

For the classical machine learning approach, i.e., MD-NCMF [42], 15 trees are used for the ensemble forest. The tree branches stop splitting when the number of samples is 20 or less to avoid overfitting. Each node has a subset of feature centroids of 3. The engineered features used are ORB, Harris Corners, Edge Based Regions (EBR), Difference of Gaussians (DoG), Harris Affine Laplace and DAISY.

For the FC-CNN w/ ResNet-50, the default parameters of [30] are not strictly followed. The reason is to train the network using a cyclic learning rate implemented in the Fast.ai library, which has been found in the literature to yield better results. The other parameters are set as follows: $epochs = 10$, maximum learning rate $m\_lr = 10e^{-2}$, and batch size $bs = 32$.

For SSD w/ MobileNets, Faster R-CNN w/ ResNet-101 and Deformable Faster R-CNN, the default parameters of their respective publications and of the related source code are used. Only the training convergence is monitored in order to choose the training iteration with the best validation performance. Likewise, a minimum Intersection over Union IoU of 0.5 is set.

All methods are real-time capable. Their run-times are so small that the differences among them are completely negligible compared to the computation needs of all the other processes running on the system during a mission.

C. Baseline Performance with Only Real-World Data Training

In this experiment, the models trained according to Table II are evaluated using the original data without artificial image perturbations. The results with respect to accuracy are shown in Table III. Deformable Faster R-CNN and Faster R-CNN have the lead when the complete dataset is used (Model-F), followed by FC-CNN with an accuracy of 95%. This is an indication that if the amount and the variance of data is high, direct classifiers offer top performance, which can save efforts and time dedicated to manually segmenting object regions on the images. SSD MobileNet still has a better performance than the MD-NCMF as a classical ML approach, but it drops below 90%. Note that SSD is mainly known for its superior speed and suitedness for embedded systems. MD-NCMF ranks last with an accuracy below 80%.

For the Models A to C, which are trained with specific scenario data, it can be seen that the performance drastically changes. More precisely, following observations can be made.

Deformable and standard Faster R-CNN still have the lead (except Model B), but MD-NCMF as classical method offers competitive results and it outperforms FC-CNN and SSD MobileNets. Thus, deep visual models suffer a great performance drop, namely $\approx 40\%$, while MD-NCMF drops only $\approx 20\%$. This strongly indicates that DL techniques are highly dependent on the amount of data and how representative it is of the real-world class distribution.

For Model B versions, MD-NCMF performs better than the rest. A reasonable explanation for this cannot be done without a close examination of the data and a visualization of the learned features by the deep models. It can be assumed that data used to train Model B, i.e., from *Genova-A*, is not sufficient for the deep models to learn strong features despite providing more samples per class than Model A and B (see Table II), and that the human-engineered features used for MD-NCMF are simply more representative.

The classical visual model provides a more stable performance across the test sets. The most representative example is when Model-C versions are benchmarked against Genova-A samples, then accuracy goes down for all methods but especially for the deep learning based ones. So, deep models have strong performance drops for particular tests; this holds especially for FC-CNN. Our hypothesis is that this is the case as FC-CNN is the only method without a region proposal step within its architecture that helps refining the classification process.

D. Robustness under Artificial Image Perturbations

Table IV shows the performance of all visual models tested with samples on which the image perturbations described in Sec. III are applied. Only Model F versions are evaluated, i.e., visual models trained with the complete dataset. Their baseline accuracy from the previous experiment in Sec. IV-C is shown for comparison. Based on this, Table IV shows the numerical values of the absolute accuracies as well as the normalized accuracies with respect to the baseline performance in form of a color code.

Note that the models are trained with the original sensor images and none of the image degradations is used to augment the training data. Deformable and standard Faster R-CNN show good robustness against the majority of the degradations, except high levels of Gaussian blur, which affects all other models as well. They also both exhibit similar performance drops for every image degradation.

The performance of the rest of the models degrades more substantially, especially from haze effects, which are emulated by alpha blend and our proposed method for producing artificial underwater haze using range information. MD-NCMF is completely ineffective at high levels of alpha blend. We can conclude that haze effects, which are the most typical natural underwater phenomena, are really important to consider when designing an underwater object detector.

For the deep visual models, JPEG compression has almost no effect. This holds even at a very low quality level of 10%. Grayworld white balance especially affects FC-CNN and SSD, indicating that the grayworld assumption is tailored towards terrestrial robotics and users have to pay attention to camera presets for underwater applications. Increasing brightness has a bigger effect than lowering it, as saturation levels may be reached quicker. As mentioned, significant blur affects all models. MD-NCMF is affected by almost all image perturbations, which supports the idea that DL approaches learn important strong features given enough data that may be hard for a human to mathematically and algorithmically conceptualize.

V. Conclusions

A fully integrated approach to Underwater Human Robot Interaction (U-HRI) was presented. It features a front-end

	MD-NCMF				FC-CNN w/ ResNet-50			
	Mod-A	Mod-B	Mod-C	Mod-F	Mod-A	Mod-B	Mod-C	Mod-F
Biograd-A	0 0	0.72	0.52	0.81	0 0	0.42	0.5	0.99
Biograd-B	0 0	0.71	0.51	0.84	0 0	0.21	0.51	0.99
Biograd-C	0.74	0.75	0.68	0.85	0.53	0.45	0.51	0.97
Brodarski-A	0.76	0.76	0 0	0.78	0.52	0.24	0 0	0.95
Brodarski-B	0.81	0.79	0.71	0.73	0.63	0.12	0.68	0.86
Brodarski-C	0.7	0.65	0 0	0.77	0.57	0.48	0 0	0.98
Brodarski-D	0.69	0.61	0.55	0.71	0.71	0.48	0.53	1
Genova-A	0.52	0 0	0.48	0.69	0.34	0 0	0.24	0.89
All scenarios	0.56	0.64	0.46	0.77	0.45	0.36	0.43	0.95

	SSD w/ MobileNets				Faster R-CNN w/ Resnet 101				Deformable Faster R-CNN			
	Mod-A	Mod-B	Mod-C	Mod-F	Mod-A	Mod-B	Mod-C	Mod-F	Mod-A	Mod-B	Mod-C	Mod-F
Biograd-A	0 0	0.35	0.38	0.84	0 0	0.63	0.65	0.99	0 0	0.65	0.64	0.99
Biograd-B	0 0	0.29	0.44	0.88	0 0	0.51	0.71	0.99	0 0	0.54	0.7	1
Biograd-C	0.36	0.31	0.4	0.82	0.74	0.58	0.67	0.98	0.74	0.57	0.67	0.98
Brodarski-A	0.38	0.3	0 0	0.87	0.72	0.48	0 0	0.97	0.73	0.49	0 0	0.97
Brodarski-B	0.33	0.29	0.48	0.84	0.72	0.52	0.85	0.96	0.74	0.5	0.87	0.97
Brodarski-C	0.32	0.28	0 0	0.86	0.79	0.56	0 0	0.99	0.78	0.6	0 0	0.99
Brodarski-D	0.29	0.26	0.36	0.79	0.82	0.55	0.68	0.99	0.84	0.54	0.72	0.99
Genova-A	0.25	0 0	0.23	0.75	0.69	0 0	0.44	0.94	0.66	0 0	0.41	0.96
All scenarios	0.28	0.361	0.29	0.85	0.59	0.49	0.52	0.98	0.61	0.5	0.53	0.98

Table III: The accuracy (0 1) of the visual models in all scenarios according to Table II.

	MD-NCMF	FC-CNN w/ ResNet-50	SSD w/ MobileNets	Faster R-CNN w/ Resnet 101	Deformable Faster R-CNN
Baseline	**0.77**	**0.95**	**0.85**	**0.98**	**0.98**
Blur ($\sigma = 1.5$)	0.61	0.8	0.63	0.85	0.95
Blur ($\sigma = 3$)	0.19	0.61	0.28	0.65	0.7
Brightness ($b = 0.5$)	0.63	0.76	0.69	0.93	0.95
Brightness ($b = 2$)	0.49	0.47	0.4	0.77	0.88
White balance (GW,$t = 0.7$)	0.11	0.48	0.46	0.79	0.82
White balance (SB,$t = .05$)	0.73	0.86	0.79	0.94	0.96
JPEG compression ($q = 60$)	0.7	0.92	0.81	0.96	0.98
JPEG compression ($q = 20$)	0.4	0.83	0.73	0.87	0.91
UW alpha blend ($w = II, d = 10, \alpha = 0.5$)	0.29	0.38	0.31	0.91	0.96
UW alpha blend ($w = II, d = 10, \alpha = 0.25$)	0.07	0.3	0.28	0.82	0.89
UW alpha blend ($w = 1C, d = 10, \alpha = 0.5$)	0.24	0.33	0.26	0.85	0.95
UW alpha blend ($w = 1C, d = 10, \alpha = 0.25$)	0.03	0.17	0.17	0.76	0.87
Haze ($w = II, \beta_{R,G,B} = [0.5, 0.15, 0.90]$)	0.42	0.49	0.4	0.85	0.95
Haze ($w = 1C, \beta_{R,G,B} = [0.5, 0.15, 0.90]$)	0.15	0.33	0.26	0.79	0.89

Table IV: The accuracy of each visual model under all image perturbations. In addition to the numerical value of accuracy shown in each cell, the cell color (0 1) illustrates the normalized value relative to the baseline accuracy to highlight performance variations and robustness.

for gesture recognition combined with a back-end with an interpreter for a language derived from the existing standard gestures for communication between human divers. The approach enables a diver to communicate commands as well as complex mission specifications via gestures to an underwater robot.

The wide range of environment conditions, especially with respect to visibility conditions, pose a severe challenge for underwater vision in general and the gesture recognition in particular. Hence, different Machine Learning (ML) methods in form of four Deep Learning (DL) approaches and a more classical ML method are investigated with respect to their robustness for the gesture recognition. In addition to the exhaustive test with real-world data from different tests in pools and during field trials, artificially degraded image data is used. To this end, we presented among others a physically realistic way to use range information for adding underwater haze in controlled ways.

References

[1] B. Verzijlenberg and M. Jenkin, "Swimming with robots: Human robot communication at depth," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, Conference Proceedings, pp. 4023–4028.

[2] A. Speers, P. M. Forooshani, M. Dicke, and M. Jenkin, "Lightweight tablet devices for command and control of ros-enabled robots," in *2013 16th International Conference on Advanced Robotics (ICAR)*, 2013, Conference Proceedings, pp. 1–6.

[3] J. Sattar and G. Dudek, "Where is your dive buddy: tracking humans underwater using spatio-temporal features," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 2007, Conference Proceedings, pp. 3654–3659.

[4] ——, "Underwater human-robot interaction via biological motion identification," in *Robotics: Science and Systems (RSS)*, 2009, Conference Proceedings.

[5] H. Bülow and A. Birk, "Diver detection by motion-segmentation and shape-analysis from a moving vehicle," in *IEEE Oceans*, 2011, Conference Proceedings.

[6] K. J. DeMarco, M. E. West, and A. M. Howard, "Sonar-based detection and tracking of a diver for underwater human-robot interaction scenarios," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2013, Conference Proceedings, pp. 2378–2383.

[7] ——, "Autonomous robot-diver assistance through joint intention theory," in *Oceans*. IEEE, 2014, Conference Proceedings, pp. 1–5.

[8] A. G. Chavez, M. Pfingsthorn, A. Birk, I. Rendulic, and N. Miskovic, "Visual diver detection using multi-descriptor nearest-class-mean random forests in the context of underwater human robot interaction (hri)," in *IEEE Oceans*, 2015, Conference Proceedings.

[9] A. G. Chavez, C. A. Mueller, A. Birk, A. Babic, and N. Miskovic, "Stereo-vision based diver pose estimation using lstm recurrent neural networks for auv navigation guidance," in *IEEE Oceans*. IEEE press, 2017, Conference Proceedings.

[10] Y. Xia and J. Sattar, "Visual diver recognition for underwater human-robot collaboration," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, Conference Proceedings, pp. 6839–6845.

[11] M. J. Islam, M. Fulton, and J. Sattar, "Toward a generic diver-following algorithm: Balancing robustness and efficiency in deep visual detection," *IEEE Robotics and Automation Letters (RAL)*, vol. 4, no. 1, pp. 113–120, 2019.

[12] W. Remmas, A. Chemori, and M. Kruusmaa, "Diver tracking in open waters: A low-cost approach based on visual and acoustic sensor fusion," *Journal of Field Robotics*, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21999

[13] K. d. Langis and J. Sattar, "Realtime multi-diver tracking and re-identification for underwater human-robot collaboration," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, Conference Proceedings, pp. 11 140–11 146.

[14] K. J. DeMarco, M. E. West, and A. M. Howard, "A simulator for underwater human-robot interaction scenarios," in *OCEANS*. IEEE, 2013, Conference Proceedings, pp. 1–10.

[15] M. Fulton, C. Edge, and J. Sattar, "Robot communication via motion: Closing the underwater human-robot interaction loop," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, Conference Proceedings, pp. 4660–4666.

[16] G. Dudek, J. Sattar, and A. Xu, "A visual language for robot control and programming: A human-interface study," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, April 2007, pp. 2507–2513.

[17] H. Bülow and A. Birk, "Gesture-recognition as basis for a human robot interface (hri) on a auv," in *IEEE Oceans*, 2011, Conference Proceedings.

[18] F. Gustin, I. Rendulic, N. Miskovic, and Z. Vukic, "Hand gesture recognition from multibeam sonar imagery," in *10th IFAC Conference on Control Applications in Marine Systems (CAMS)*, V. Hassan, Ed., vol. 49. IFAC PapersOnLine, 2016, Conference Proceedings, pp. 470–475. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896316320389

[19] A. G. Chavez and A. Birk, "Underwater gesture recognition based on multi-descriptor random forests (md-ncmf)," 2015.

[20] D. Chiarella, M. Bibuli, G. Bruzzone, M. Caccia, A. Ranieri, E. Zereik, L. Marconi, and P. Cutugno, "A novel gesture-based language for underwater human–robot interaction," *Journal of Marine Science and Engineering*, vol. 6, no. 3, 2018.

[21] D. Chiarella, M. Bibuli, G. Bruzzone, M. Caccia, A. Ranieri, E. Zereik, L. Marconi, and P. Cutugno, "Gesture-based language for diver-robot underwater interaction," in *OCEANS 2015 - Genova*, May 2015, pp. 1–9.

[22] N. Miskovic, A. Pascoal, M. Bibuli, M. Caccia, J. A. Neasham, A. Birk, M. Egi, K. Grammer, A. Marroni, A. Vasilijevic, N. Đ, and Z. Vukic, "Caddy project, year 3: The final validation trials," in *OCEANS*. IEEE, 2017, Conference Proceedings, pp. 1–5.

[23] N. Miskovic, A. Pascoal, M. Bibuli, M. Caccia, J. A. Neasham, A. Birk, M. Egi, K. Grammer, A. Marroni, A. Vasilijevic, and Z. Vukic, "Caddy project, year 2: The first validation trials," in *10th IFAC Conference on Control Applications in Marine Systems (CAMS)*. International Federation of Automatic Control, 2016, Conference Proceedings.

[24] M. J. Islam, M. Ho, and J. Sattar, "Understanding human motion and gestures for underwater human–robot collaboration," *Journal of Field Robotics*, vol. 36, no. 5, pp. 851–873, 2019.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2017. [Online]. Available: https://doi.org/10.1109/TPAMI.2016.2577031

[27] R. Codd-Downey and M. Jenkin, "Human robot interaction using diver hand signals," in *14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2019, Conference Proceedings, pp. 550–551.

[28] ——, "Finding divers with scubanet," in *International Conference on Robotics and Automation (ICRA)*, 2019, Conference Proceedings, pp. 5746–5751.

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, Conference Proceedings, pp. 4510–4520.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. [Online]. Available: https://doi.org/10.1109/cvpr.2016.90

[31] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[32] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *CoRR*, vol. abs/1703.06211, 2017. [Online]. Available: http://arxiv.org/abs/1703.06211

[33] D. Akkaynak and T. Treibitz, "A revised underwater image formation model," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 6723–6732.

[34] M. O'Byrne, V. Pakrashi, F. Schoefs, and B. Ghosh, "Semantic segmentation of underwater imagery using deep networks trained on synthetic imagery," *Journal of Marine Science and Engineering*, vol. 6, no. 3, p. 93, 2018. [Online]. Available: https://www.mdpi.com/2077-1312/6/3/93

[35] Y. Hu, K. Wang, X. Zhao, H. Wang, and Y. Li, "Underwater image restoration based on convolutional neural network," in *Proceedings of The 10th Asian Conference on Machine Learning*, Z. Jun and T. Ichiro, Eds., vol. 95. PMLR, 2018, Conference Paper, pp. 296–311. [Online]. Available: http://proceedings.mlr.press

[36] J. Li, K. A. Skinner, R. M. Eustice, and M. Johnson-Roberson, "Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 387–394, 2018.

[37] C. Fabbri, M. J. Islam, and J. Sattar, "Enhancing underwater imagery using generative adversarial networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 7159–7165.

[38] D. G. Kim and S. M. Kim, "Single image-based enhancement techniques for underwater optical imaging," *Journal of Ocean Engineering and Technology*, vol. 34, no. 6, pp. 442–453, 2020. [Online]. Available: https://doi.org/10.26748/KSOE.2020.030http://www.joet.org/journal/view.php?number=2997

[39] A. Gomez Chavez, A. Ranieri, D. Chiarella, E. Zereik, A. Babić, and A. Birk, "Caddy underwater stereo-vision dataset for human–robot interaction HRI in the context of diver activities," *Journal of Marine Science and Engineering*, vol. 7, no. 1, 2019.

[40] N.Mayer, E.Ilg, P.Häusser, P.Fischer, D.Cremers, A.Dosovitskiy, and T.Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, arXiv:1512.02134.

[41] A. Gomez Chavez, Q. Xu, C. A. Mueller, S. Schwertfeger, and A. Birk, "Adaptive navigation scheme for optimal deep-sea localization using multimodal perception cues," *arXiv preprint*, 2019, [Accepted at IROS 2019]. arXiv: 1906.04888 [cs.RO].

[42] A. G. Chavez, M. Pfingsthorn, A. Birk, I. Rendulić, and N. Misković, "Visual diver detection using multi-descriptor nearest-class-mean random forests in the context of underwater human robot interaction (HRI)," in *OCEANS 2015 - Genova*, May 2015, pp. 1–7.

Arturo Gomez Chavez : a.gomezchavez@jacobs-university.de, Jacobs University Bremen, Germany

Andrea Ranieri : andrea.ranieri@cnr.it, Institute of Applied Mathematics and Information Technology - National Research Council of Italy

Davide Chiarella : davide.chiarella@cnr.it, Institute for Computational Linguistics A. Zampolli - National Research Council of Italy

Andreas Birk : a.birk@jacobs-university.de, Jacobs University Bremen, Germany