Vision-Based Computational Methods Towards Effective Underwater Multi-Human-Robot Interaction

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY
Sadman Sakib Enan

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Advisor: Junaed Sattar, Ph.D.

April, 2024
Acknowledgements

First and foremost, I would like to extend my deepest appreciation to my adviser, Professor Junaed Sattar. Without his guidance, none of the accomplishments I have achieved during my doctoral studies would have been possible. I vividly recall October 5, 2018, the day I initially conversed with Junaed. His genuine enthusiasm for robotics deeply resonated with me, compelling me to express my desire to work with him. Junaed helped me learn many things, from how to conduct research and the art of scientific writing, to language refinement (yes, even English!), and proficiency in Linux and robotics systems. I am indebted to him for his willingness to invest in me, despite my limited prior experience in robotics research. Junaed’s continuous challenge and insightful feedback have profoundly shaped my academic pursuits, inspiring me to strive for excellence. I am eternally grateful for his relentless support, encouragement, and guidance throughout my Ph.D. journey.

I am also immensely thankful to my dissertation committee members: Professor Nikolaos Papanikolopoulos, Professor Ryan James Caverly, and Professor Evan Suma Rosenberg. Their invaluable feedback and guidance have been instrumental from my preliminary exams to my Ph.D. proposal exam. I also extend my thanks to Professor Shashi Shekhar for providing his key insights during my oral preliminary exam. Additionally, I am forever grateful to Nikos for being an amazing mentor since my involvement with the Minnesota Robotics Institute. Moreover, I would like to convey my appreciation to Professor Jaideep Srivastava for taking the time to chat with me, understanding my research interests, discussing the scope of my work, and ultimately motivating me to apply to the CS Ph.D. program at the University of Minnesota.

Throughout my doctoral studies, I have had the privilege of collaborating with brilliant minds who have not only helped me acquire new skills but also fueled my
passion for research. I would like to begin by expressing my gratitude to Dr. Md Jahidul Islam, whose guidance played a pivotal role in my transition to field robotics research. His mentorship has been invaluable to me. Field validation is a crucial aspect of my research for which I constantly need help from divers. I owe a debt of gratitude to Cory Ohnsted, Demetrious Kutzke, Chelsey Edge, Dr. Jungseok Hong, and Sakshi Singh for their assistance in this regard. I also want to acknowledge the contributions of my current and former colleagues: Dr. Jiawei Mo, Dr. Michael Fulton, Andrea Walker, Karin de Langis, Corey Knutson, Maryam Kameli, and David Widham. Their constructive feedback and collaborative spirit have been instrumental in refining my research. Lastly, I am thankful for the opportunity to collaborate with undergraduate and high school students, such as Grant Schwidder, Adhithya Anandaraj, Christopher Morse, Zuhayr Nibras Islam, and others. Their enthusiasm and dedication have enriched my research experience.

I am thankful for the generous support provided by the Department of Computer Science & Engineering at the University of Minnesota, the Minnesota Robotics Institute, MnDRIVE, and the National Science Foundation.

I want to conclude by thanking my sister, Asma-ul-Husna, for always supporting me and cheering me up during tough moments. Finally, I want to express my heartfelt appreciation to my son, for loving me so much and, most importantly, for respecting my work computer. Despite his constant desire for playtime, he often showed mature understanding when I explained that I was working.
Dedication

To my wife and my parents.

I dedicate this dissertation to my beloved wife Euna Mehnaz Khan, whose unwavering trust and support have been my guiding light throughout my doctoral studies. And, I dedicate this to my loving parents Amena Begum and Mohammad Ismail. Words fall short in expressing the depth of my affection for you and the significance you hold in my life. Thank you for always pushing me to achieve the best. I love you beyond measure.
Abstract

Numerous important tasks, such as environmental monitoring, cable or wreckage inspection, search-and-rescue, and oil drilling or spillage monitoring, are conducted underwater. A team of human divers typically carries out these challenging and often dangerous tasks, occasionally receiving assistance from a Remotely Operated Vehicle (ROV). However, an ROV is mainly controlled by someone on the surface which leads to inefficient collaboration due to the indirect engagement among the divers and the robot. In contrast, an Autonomous Underwater Vehicle (AUV) does not require a surface operator to operate and can significantly enhance task efficiency by actively engaging with a team of human divers and other AUVs during the tasks. Thus, it is imperative for the AUVs to have robust multi-Human-Robot Interaction (mHRI) capability. In this dissertation, we present a set of vision-based computational methods for AUV perception to facilitate effective underwater mHRI to allow successful collaboration among multiple divers and AUVs. Furthermore, we provide several novel underwater datasets designed to facilitate learning about robot motion, diver identity, their pose information, and multi-human-robot collaborative scenarios. Our proposed methods allow the AUV to enable human-comprehensible interaction between multiple AUVs, identify unique divers for secure interaction and collaboration, reposition itself for interaction by determining whether their human partners are attentive, and identify the current activity of divers to make informed decisions. However, the general operation of AUVs is severely impacted by various factors, such as water turbidity, rapid currents, varying lighting conditions, and signal attenuation. AUVs also have several platform-specific constraints, such as finite battery life, limited on-board processing power, and real-time operational requirements. We have taken these challenges into consideration while designing and implementing our proposed algorithms on-board physical AUV platforms. We have elaborated on the rationale behind the specific design choices made for each system. Experimental validations on proposed datasets as well as through numerous robot trials, performed in both closed- and open-water environments (e.g., swimming pools and oceans), show the efficacy of each proposed system.
Contents

Acknowledgements i

Dedication iii

Abstract iv

List of Tables ix

List of Figures xi

1 Introduction 1

1.1 Research Objectives and Scope 4

1.2 Research Contributions 5

1.2.1 Human-Comprehensible Gestural Language Recognition 5

1.2.2 Visual Diver Face Identification System 5

1.2.3 Diver Identification Using Distance and Photometric Conditions Invariant Features 6

1.2.4 Diver Attention Estimation Framework 6

1.2.5 Diver Activity Recognition Framework 7

1.3 Domain Constraints and Sensor Selection 7

1.4 AUV Platforms Used in the Research 8

1.4.1 Aqua 9

1.4.2 LoCO 10

1.5 Overview of the Manuscript 11
2 Robotic Detection of a Human-Comprehensible Gestural Language for Underwater Multi-Human-Robot Collaboration 12

2.1 Related Work .................................................. 15
2.2 Gestural Recognition Framework .................................. 16
  2.2.1 Communication Message Design .................................. 16
  2.2.2 Gesture Implementation ........................................ 17
  2.2.3 Gestural Message Dataset ........................................ 19
  2.2.4 Robot-To-Robot Communication Network (RRCommNet) .... 20
  2.2.5 Implementation Details .......................................... 22
2.3 Human Transcription Study ........................................ 23
2.4 Evaluation Of RRCommNet .......................................... 24
  2.4.1 Evaluation Procedure and Metrics ............................... 24
  2.4.2 Results ...................................................... 25
  2.4.3 Runtime Performance on Embedded GPU ....................... 30
2.5 Summary ......................................................... 30

3 Towards Visual Diver Face Identification by Autonomous Underwater Robots 32

3.1 Related Work .................................................. 34
3.2 Methodology ..................................................... 36
  3.2.1 Offline Module ............................................. 36
  3.2.2 Online Module ................................................ 39
3.3 Experimental Setup and Evaluation ................................ 41
  3.3.1 Dataset ...................................................... 41
  3.3.2 Evaluation Criteria ........................................... 42
  3.3.3 Model Selection and Evaluation Results ......................... 43
  3.3.4 Practical Feasibility ........................................... 46
3.4 Summary ......................................................... 46

4 Diver Identification Using Anthropometric Data Ratios For Underwater Multi Human-Robot Collaboration 47

4.1 Related Work .................................................. 50
4.2 Methodology ..................................................... 51
4.2.1 Diver Identification Algorithm ........................................... 51
4.2.2 Diver Identification Framework ......................................... 56
4.3 Experiments ........................................................................... 58
  4.3.1 UDI Dataset ................................................................... 58
  4.3.2 Setup ............................................................................. 59
  4.3.3 Experimental Scenarios .................................................... 60
4.4 Results .................................................................................. 60
  4.4.1 Performance of the Offline Framework ............................. 61
  4.4.2 Performance of the Online Framework ............................. 61
4.5 Summary .............................................................................. 65

5 A Diver Attention Estimation Framework for Effective Underwater Human-Robot Interaction 67
  5.1 Related Work ..................................................................... 69
  5.2 DATT Dataset ..................................................................... 70
  5.3 DATT-Net ........................................................................... 73
    5.3.1 Feature Extractor .............................................................. 73
    5.3.2 Facial Anchors ............................................................... 74
    5.3.3 Objective Function Formulation .................................... 74
    5.3.4 Implementation Details .................................................. 76
  5.4 Diver Attention Estimation .................................................... 76
  5.5 Robot Controller ................................................................. 78
  5.6 Experimental Evaluations ...................................................... 78
    5.6.1 Results ......................................................................... 79
    5.6.2 Limitations of the Proposed Framework ......................... 83
  5.7 Summary ........................................................................... 83

6 Semantically-Aware Diver Activity Recognition Framework for Effective Underwater Multi-Human-Robot Collaboration 85
  6.1 Related Work ..................................................................... 88
  6.2 UDA Dataset ..................................................................... 90
  6.3 Diver Action Recognition Framework .................................. 93
    6.3.1 Feature Extraction .......................................................... 93
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Definition, type, and duration of the proposed communication messages where D=Directional, I/C=Information/Command, CC=Conversation Control</td>
<td>17</td>
</tr>
<tr>
<td>2.2</td>
<td>Definitions of the communication messages</td>
<td>18</td>
</tr>
<tr>
<td>2.3</td>
<td>Results of the human transcription study in recognizing different gestural messages. The values represent avg. trans. accuracy (%) / avg. confidence (out of 10)</td>
<td>24</td>
</tr>
<tr>
<td>2.4</td>
<td>Comparison of the gestural message recognition accuracy on both simulated and real data</td>
<td>26</td>
</tr>
<tr>
<td>2.5</td>
<td>Comparison between the performance of RRCommNet and RRCommNet-Skip in recognizing different gestural messages. The values represent avg. recog. probability (%) / avg. inference time (s)</td>
<td>29</td>
</tr>
<tr>
<td>4.1</td>
<td>Our proposed embedding network, which takes 45-d ADR features and project them into 16-d features</td>
<td>54</td>
</tr>
<tr>
<td>4.2</td>
<td>Network structure and online training capabilities of different models used for our Diver Identification Framework</td>
<td>56</td>
</tr>
<tr>
<td>4.3</td>
<td>Comparison of average accuracy for the metric learning embedding network and the 10 models on our UDI dataset. The values are in percentages</td>
<td>59</td>
</tr>
<tr>
<td>4.4</td>
<td>Results of 16 robot trials conducted in closed-water environments for diver identification task where prediction accuracy = ( \frac{(TP+TN)}{(TP+TN+FP+FN)} )</td>
<td>62</td>
</tr>
<tr>
<td>5.1</td>
<td>Diver Face detection and Keypoints regression results on the evaluation set. The values are in percentages</td>
<td>79</td>
</tr>
<tr>
<td>5.2</td>
<td>Diver attention estimation results on the evaluation set. The accuracy values are in percentages</td>
<td>82</td>
</tr>
</tbody>
</table>
6.1 Diver activity recognition performance on the test set. Precision, Recall, and F1-Score are computed as weighted averages. The values are in percentages.
## List of Figures

1.1 A few example underwater tasks that can be benefited from human-robot collaboration: (a) water quality assessment [1], (b) coral reef exploration and monitoring [2], and (c) submarine cable inspection and maintenance [3].

1.2 A multi-human-robot team is conducting a data collection experiment simulated in a closed-water environment, such as a swimming pool. One of the divers is holding an underwater robot that functions as a data collection device. An AUV is observing the experiment to determine if additional assistance can be provided.

1.3 The Aqua AUV swimming over coral reefs in the coastal region of the Caribbean Sea during a data collection experiment.

1.4 The initial trial of the LoCO AUV’s tetherless capability in the Caribbean Sea off the coast of Barbados.

2.1 An underwater gestural communication framework where the speaker robot is communicating back with the listener robot by making a *nodding* motion to mean YES which a human observer is also able to understand.

2.2 The Aqua AUV is performing ASCEND message in both simulated and real underwater environments.

2.3 The description of the configuration file for the START MAPPING message.

2.4 The proposed gestural message recognition framework, RRCommNet. It takes a video clip as input and extracts spatio-temporal features. Self-attention mechanism is performed on the feature representation to get the final classification scores.
2.5 Gestural message recognition using RRCommNet. (a) Prediction is being
performed on the 8th crop. (b) Final prediction is made by averaging the
recognition probabilities on all 10 crops. ................................................. 27

2.6 Confusion matrix for the gestural message recognition. It contains 5
different experiments: RRCommNet in both simulated and real data,
RRCommNet-Skip in both simulated and real data, and human tran-
scription. ................................................................. 28

3.1 An example of underwater human-robot interaction scenario. When an
AUV positions itself at a suitable distance for the interaction, our pro-
posed system performs diver identification. Accurate identification en-
ures AUVs carry out instructions from authorized divers only. ...... 33

3.2 Data augmentation steps for preparing the training dataset for the facial
feature extractor network. (a) *Frontalization*. (b) Automatic crop. (c)
Color correction and keypoints regression. (d) Apply mask and regulator
based on the 15 keypoints. (e) Colorization [4] to incorporate underwa-
ter lighting conditions. (f) Applying underwater effect [5]. (g) Crop to
tighten the boundary. ................................................................. 36

3.3 Reconstructed faces of two participants. Left side of each pair is the
original image, and the right side is the reconstructed one: (left pair)
diver face and (right pair) regular face. ................................. 37

3.4 An example of variance analysis for dimensionality reduction. The blue
and the green values represent variance values for regular and diver faces,
respectively. The variance values for both regular and diver faces from
dimensions 3 and 6 are more than a predefined threshold value (0.25),
making these two dimensions desirable for representing regular and diver
faces. ................................................................. 38

3.5 Complete pipeline of the proposed diver face identification system. The
overall system consists of an online as well as an offline component. In
real-world scenarios, only the online module is executed by AUVs to
recognize scuba divers, *e.g.*, to ensure they are allowed to operate the
AUVs. ................................................................. 40
3.6 A few instances sampled from the proposed dataset. It includes a total number of 22,031 (regular + diver) face images with 184 different subjects.

3.7 Failure cases of the domain transfer techniques. (a) Fails to completely remove masks from the diver faces. (b) Improperly trained data generation model completely destroys the underlying information of the image.

3.8 Diver face identification accuracy results evaluated on real-world data with different approaches. Our approach with VA shows the highest accuracy among all the approaches.

4.1 An example scenario of a robot identifying divers with our proposed framework by using the anthropometric features extracted from the divers.

4.2 Given an RGB image input, our proposed algorithm first extracts features from the diver’s pose. It then predicts the diver’s ID using these highly distinguishable features. The numbers in the second sub-figure represent 17 body keypoints. Additionally, the AD\{1, \ldots, 10\} in the third sub-figure are the extracted features.

4.3 Examples of pose estimation failures caused by rapid movements of divers’ arms underwater. As can be seen, the predictions for wrists, elbows, and shoulder joints locations are inaccurate.

4.4 Results of the metric learning technique: clear separation among classes are seen. The 16-d features are visualized in 2D plots using t-SNE technique. The top and bottom rows show the clustering results on the training and test data, respectively.

4.5 Our proposed Diver Identification Framework. The states in yellow boxes are common between the offline and online frameworks. Whereas, the state in the green box is used only in the online framework to perform model training during mission. The gray colored box shows the DRP estimator which runs continuously in the background.

4.6 Average accuracy of the offline framework where the training has been performed using the UDI dataset. (Top row): average accuracy for a specific number of test frames across all models, (Bottom row): average accuracy of each model across all frames.
4.7 Average accuracy of the online framework. (Top row): average accuracy for a specific number of test frames across all models, (Bottom row): average accuracy of each model across all frames. 64

4.8 Demonstration of the proposed framework on a physical AUV platform in different environments. (a) An external view of the online identification framework. (b), (c), and (d) depict the identification state of our framework in terrestrial, closed water (swimming pool), and open water (Caribbean Sea) environments. 65

5.1 Demonstration of the proposed framework running on-board the Aqua AUV during our open-water experiments in the Caribbean Sea, off the coast of Barbados. The red dashed line represents the robot’s future trajectory to position itself conveniently for interaction. 68

5.2 (a) A sample image from DATT dataset and the corresponding label containing two bounding box coordinates and 10 facial keypoints. (b) A few additional annotated samples where the divers are wearing different types of scuba gear and are looking at different directions. 72

5.3 Network architecture of the proposed DATT-Net model that employs a multi-scale learning on a feature pyramid and has an extra supervision branch to learn the geometric relation of facial keypoints. This allows the algorithm to work even when the diver is distant and facing at a 90 degree angle with respect to the robot. 73

5.4 The training performance of DATT-Net in terms of the validation loss where the backbone is either (a) ResNet-50, or (b) MobileNet-V2. Here, iterations = (no. of training images/batch size) * epochs. 77

5.5 Qualitative performance of the proposed diver attention estimation framework when run on-board the Aqua AUV in both closed- and open-water environments. Note the robustness of our method as it works at different distances and in challenging lighting conditions. 81

5.6 Two examples of diver attention estimation failures where the algorithm incorrectly detects a diver face in the reflection on the water surface (left image) and in the floor of the pool (right image). 84
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Demonstration of the proposed diver activity recognition framework, making inference on real-world underwater scenarios involving multiple divers and robots. The proposed method is able to learn highly discriminative spatio-temporal features from underwater video clips by focusing on important elements within the scene (as shown in the inset), such as divers, robots, and objects of interest, and their interactions with each other.</td>
</tr>
<tr>
<td>6.2</td>
<td>A few sample images and their semantic labels from the proposed UDA dataset. In our dataset, divers, robots, and objects of interest are annotated with purple, red, and green colors, respectively.</td>
</tr>
<tr>
<td>6.3</td>
<td>An overview of the network architecture of DAR-Net. It takes an underwater diver activity video clip as input and extracts highly discriminative spatio-temporal features by incorporating additional supervision from scene semantics. Intermediary skip connections are used to avoid overfitting.</td>
</tr>
<tr>
<td>6.4</td>
<td>The training performance of DAR-Net. Note the convergence in both the classification and segmentation validation loss graphs.</td>
</tr>
<tr>
<td>6.5</td>
<td>The effect of using semantic labels during the training of DAR-Net. The inclusion of scene semantics directs the model’s focus towards relevant regions in the image for feature learning. In contrast, traditional activity recognition models frequently prioritize irrelevant areas, such as pool lane markings during training. Lighter colors indicate higher attention values.</td>
</tr>
<tr>
<td>6.6</td>
<td>Confusion matrix for diver activity recognition, computed on the 30 video clips from our test set. It highlights the robustness of the proposed framework in accurately identifying the majority of diver activity categories. The matrix entries are normalized.</td>
</tr>
<tr>
<td>6.7</td>
<td>Per-category Precision, Recall, and F1-Score for the proposed diver activity recognition framework. It indicates a similar trend where the model’s performance is notably lower in predicting the <em>Diver Busy</em> and <em>Robot-Diver Interaction</em> categories.</td>
</tr>
<tr>
<td>A.1</td>
<td>Demonstration of underwater image super-resolution technique where high-resolution image can be generated given a lower-resolution image.</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>A.2</td>
<td>USR-248 dataset includes one HR set and three corresponding LR sets of images; hence, there are three possible combinations (i.e., 2×, 4× and 8×) for supervised training of underwater image super-resolution models.</td>
</tr>
<tr>
<td>A.3</td>
<td>Overview of the SUIM dataset.</td>
</tr>
<tr>
<td>A.4</td>
<td>(a) LoCO AUV during deployment in the Caribbean Sea off the coast of Barbados, (b) LoCO’s OLED display showing a sample menu.</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Exploration of and task execution in the underwater environments have traditionally been the domain of human divers. They have gone deep underwater, discovering the unknown, conducting research, and recovering lost artifacts. Yet, the challenges of underwater exploration — limited endurance, physiological constraints, and the inherent risks — have always been part of the diver’s reality. Recent advancements in underwater robotics have paved the way for exciting collaboration between divers and robots (e.g., Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs)).

Although ROVs and AUVs are different types of vehicles (one is operated remotely and the other one is autonomous), in the context of human-robot collaboration, the key principles remain the same where the divers possess the ability to adapt and make complex decisions based on their experience and intuition, while robots bring precision, endurance, and the ability to access challenging or hazardous environments.

There are a number of major applications that can benefit from such collaborations, such as marine ecosystem and archaeological site exploration [2, 7, 8], submarine cables and wreckage inspection [9], offshore oil and gas site maintenance [10], and search-and-rescue missions [11]. Fig. 1.1 presents a few such examples. Here, the AUV can exploit its array of sensors (e.g., cameras, sonar, and grippers) to assist divers in collecting data [12], mapping underwater terrain [13], maintaining underwater infrastructure, performing routine inspection tasks, and handling repetitive operations. Meanwhile, divers can focus on more complex tasks that require human judgment and intervention. Traditionally, ROVs are deployed underwater, as dive buddies, to provide assistance to the
divers performing challenging and dangerous tasks. With the availability of low-cost AUVs and increasing on-board computing power, AUVs can also serve as able companions in such tasks. Typically, these collaborative tasks will involve a multi-human-robot team, consisting of multiple underwater robots and divers (see Fig. 1.2).

For successful collaboration, however, the AUV must possess effective (e.g., robust and secure) multi-Human-Robot Interaction (mHRI) capability. Particularly, it is imperative for the AUV to be able to:

1. communicate with other AUVs using a human-comprehensible language (i.e., human divers can also perceive the AUV-AUV interaction) without carrying any additional equipment,
2. identify the authorized diver to collaborate with or take instructions from,
3. reposition itself for interaction by determining whether their human partners are attentive, and
4. predict the activity of the divers so that it can take proactive actions to help a diver or intervene in the current task of a diver for urgent communication if required. Once AUVs achieve the aforementioned objectives, several advantages become apparent. Multi-human-robot collaborative missions would witness heightened productivity and safety levels. With AUVs possessing the ability to understand and interact with both their robotic counterparts and human divers, tasks can be delegated efficiently, minimizing redundant efforts and maximizing productivity. The ability to identify authorized divers and autonomously position themselves for interaction will enable secure and seamless communication. Moreover, the proactive assistance and intervention capabilities of AUVs contribute to enhanced
Figure 1.2: A multi-human-robot team is conducting a data collection experiment simulated in a closed-water environment, such as a swimming pool. One of the divers is holding an underwater robot that functions as a data collection device. An AUV is observing the experiment to determine if additional assistance can be provided.

safety for divers, particularly in emergencies or challenging underwater conditions. However, there is a significant lack of research in the existing literature regarding these AUV capabilities in an underwater mHRI context. There are two primary contributing factors behind this research gap. First, AUV perception is severely affected in underwater environments due to several adversarial constraints, such as water turbidity and low visibility. AUVs lack high-bandwidth radio communication and localization using the Global Positioning System (GPS) due to significant signal attenuation. Moreover, AUVs encounter specific constraints inherent to their platforms, including finite battery life, limited on-board processing power, and the need for real-time operation. Therefore, careful considerations need to be made while designing and developing algorithms for underwater mHRI. Second, underwater data collection is extremely challenging because of several constraints unique to the domain, such as high pressure, unpredictable currents, extreme temperatures, and reduced situational awareness. Consequently, data-driven approaches encounter substantial limitations during the development phase. This
challenge intensifies in scenarios involving multiple divers and underwater robots within the scene.

In this dissertation, we present various vision-based computational methods for AUV perception to ensure effective underwater mHRI for successful multi-human-robot collaboration. Our proposed methods enable AUV-AUV interaction using a human-comprehensible language and diver identification for secure interaction and collaboration. With our methods, an AUV can also determine when and how to position itself conveniently to facilitate a diver-AUV interaction. In addition, we have also developed solutions that enable AUVs to allow them to analyze and understand diver activities underwater to make informed decisions. Our proposed solutions do not require the AUV or the diver to carry or use acoustic modem [14], Artificial Reality (AR) tag markers [15], digital display device [16], lights [17], and high-speed tethered connection to a topside operator [18]. We have implemented all of our proposed methods on physical AUVs and conducted comprehensive evaluations in diverse environments, encompassing both closed-water settings like swimming pools and open-water environments, such as oceans, in collaboration with human divers.

1.1 Research Objectives and Scope

The main objective of the dissertation is to enable effective underwater mHRI for successful multi-human-robot collaboration. To achieve this goal, the AUV must understand other AUVs and be able to effectively interact with human divers. It must know who to collaborate with or take instructions from (i.e., identify unique divers). The AUV should be able to determine when and how to position itself conveniently to facilitate seamless AUV-human interactions. Finally, the AUV must be able to understand the activity of its dive partners so that it can take proactive actions to assist a diver or intervene in the current task of a diver for urgent communication if required. These capabilities not only enable AUVs to effectively comprehend and reason about their surroundings but also make them intelligent dive partners. Consequently, these advancements significantly enhance the overall efficiency of any mHRI mission.

Towards achieving these objectives, we design and implement several vision-based perception systems for AUVs. We perform thorough experimental evaluations in diverse
water bodies to validate our proposed systems. However, how the AUVs and the divers will collaborate during a task falls outside of the scope of this dissertation. The following section describes our research contributions and Sec. 1.3 discusses some of the domain-related constraints and additional scope of our research. Finally, Sec. 1.4 introduces the AUV platforms used in the research.

1.2 Research Contributions

1.2.1 Human-Comprehensible Gestural Language Recognition

In this work [19], we present a motion-based robotic communication framework that enables non-verbal communication among AUVs and human divers. We design a gestural language for AUV-to-AUV communication which can be easily understood by divers observing the conversation — unlike typical radio frequency, light, or audio-based AUV communication. To allow AUVs to visually understand a gesture from another AUV, we propose a deep network (RRCommNet) which exploits a self-attention mechanism to learn to recognize each message by extracting maximally discriminative spatio-temporal features. We train this network on diverse simulated and real-world data. Our experimental evaluations, both in simulation and in closed-water robot trials, demonstrate that the proposed RRCommNet architecture is able to decipher gesture-based messages with an average accuracy of 88 – 94% on simulated data and 73 – 83% on real data (depending on the version of the model used). Further, by performing a message transcription study with human participants, we also show that the proposed language can be understood by humans with an overall transcription accuracy of 88%. Finally, we discuss the inference runtime of RRCommNet on embedded GPU hardware, for real-time use on board AUVs in the field.

1.2.2 Visual Diver Face Identification System

In this work [20], we present a novel face identification system for underwater robots to identify scuba divers whose faces are heavily obscured by masks and breathing apparatus. The system’s core component is an autoencoder network that can extract features from diver faces. Due to the unavailability of a large-scale diver face dataset, we adopt
a series of data augmentation steps to generate synthetic data. With this augmented dataset, we train the proposed autoencoder network and perform variance analysis on the extracted features to reduce their dimension. Our results demonstrate that the features with reduced dimension are highly discriminative and contribute to uniquely identifying individual divers. We also show that the proposed system outperforms existing face recognition algorithms for diver face identification tasks. Finally, we discuss the practical feasibility of the system for embedded inference on robotic platforms.

1.2.3 Diver Identification Using Distance and Photometric Conditions Invariant Features

Recent advances in efficient design, perception algorithms, and computing hardware have made it possible to create improved human-robot interaction capabilities for AUVs. To conduct secure missions as underwater human-robot teams, AUVs require the ability to accurately identify divers. However, this remains an open problem due to divers’ challenging visual features, mainly caused by similar-looking scuba gear. In this work [21], we present a novel algorithm that can perform diver identification using either pre-trained models or models trained during deployment. We exploit anthropometric data obtained from diver pose estimates to generate robust features invariant to distance and photometric conditions. We also propose an embedding network that maximizes inter-class distances in the feature space and minimizes those for the intra-class features, which significantly improves classification performance. Furthermore, we demonstrate an end-to-end diver identification framework, which runs on an AUV, and evaluate the accuracy of the proposed algorithm. Quantitative results in controlled-water experiments show that our proposed algorithm can perform diver identification with a high degree of accuracy.

1.2.4 Diver Attention Estimation Framework

Many underwater tasks, such as cable-and-wreckage inspection and search-and-rescue, can benefit from robust Human-Robot Interaction (HRI) capabilities. With the recent advancements in vision-based underwater HRI methods, AUVs have the capability to interact with their human partners without requiring assistance from a topside operator.
However, in these methods, the AUV assumes that the diver is ready for interaction, while in reality, the diver may be distracted. In this work [22], we present a diver attention estimation framework for AUVs to autonomously detect the attentiveness of a diver, and develop a robot controller to allow the AUV to navigate and reorient itself with respect to the diver before initiating an interaction. The core element of the framework is a deep convolutional neural network called DATT-Net. It is based on a pyramid structure that can exploit the geometric relations among 10 facial keypoints of a diver to estimate their head orientation, which we use as an indicator of attentiveness. Our on-the-bench experimental evaluations and real-world experiments during both closed- and open-water robot trials confirm the efficacy of the proposed framework.

1.2.5 Diver Activity Recognition Framework

Multi-human-robot collaboration enhances underwater tasks by combining human expertise with robotic capabilities. For effective collaboration, autonomous underwater vehicles need to comprehend the surroundings and recognize the current activity of divers and its significance. In this work, we enable this robotic capability by proposing a novel transformer-based framework that can extract highly discriminative spatio-temporal features to recognize different diver activities. To train the framework, we propose to use supervision from underwater scene semantics and optimize a multi-loss objective function that considers both local and global spatio-temporal information. To facilitate the training process, we present the first-ever underwater diver activity dataset that contains over 2400 semantically segmented images with pixel annotations for divers, robots, and objects of interest. Experimental evaluations in closed-water environments show that the proposed framework can recognize different diver activities with a promising accuracy, outperforming state-of-the-art activity recognition models.

1.3 Domain Constraints and Sensor Selection

Our research is primarily conducted in underwater environments which pose numerous challenges, such as high pressure, low visibility, unpredictable currents, extreme temperatures, and corrosive saltwater for both AUVs and divers. Traditional sensory
 mediums, such as radio and other electromagnetic modalities, suffer from signal attenuation and degradation underwater [23]. Furthermore, due to limited visibility and lack of distinctive landmarks, navigation and localization get extremely challenging [24], often requiring advanced and expensive sensors and navigation systems to overcome these challenges underwater. Although acoustic signals work quite well in underwater settings [25], it often requires transceiver, transponder, etc., to perform communication or localization. Furthermore, they introduce auditory masking which may cause marine animals to alter their behavior. Carrying such specialized and expensive equipment puts additional burden on divers whose physical and cognitive loads are already elevated.

We envision a multi-human-robot collaborative environment where AUVs can understand other AUVs and human divers. And, human divers can also understand what different AUVs are communicating between them, without carrying any specialized and expensive equipment. Therefore, we decided to mainly use the camera sensor of the AUV to perceive the surroundings. Furthermore, visual sensing is unobtrusive and passive in nature. In fact, AUVs are successfully using their visual perception capabilities in littoral regions owing to the success of underwater vision improvements [26] and novel deep learning innovations [27].

Since open-water environments (e.g., oceans) pose significant challenges, we develop our algorithms in slightly less complex closed-water environments (e.g., swimming pools). We also utilized various simulated underwater environments created using platforms, such as Robot Operating System (ROS) Gazebo [28] and Unreal Engine [29], to initially design and develop the methodologies.

1.4 AUV Platforms Used in the Research

A major component of field robotics research (e.g., underwater robotics) is the implementation, integration, and validation of the proposed algorithms and frameworks on real robotic platforms. For our research, we have used the following two AUV platforms.
1.4.1 Aqua

Aqua \[6\] is an autonomous underwater robot, developed primarily by McGill and York University researchers and is currently manufactured by Independent Robotics company\[1\]. It has six flippers and employs a tripod gait mechanism to achieve forward and heave (up and down) movements. It is also equipped with Inertial Measurement Unit (IMU) and depth sensors, and can rotate around $xyz$ axes to achieve roll, pitch, and yaw movements. The Aqua AUV includes a ROS-based software stack and three computing platforms where the first one is responsible for sensors, navigation, and high-level processing; the second one is responsible for controlling the movement of the flippers of the robot; and the final one is responsible for running deep learning inference. The Aqua AUV comes with an autopilot client that can be used to control the movements of the AUV. The body of the AUV is made out of aluminum. The AUV can be controlled either using a tether or without a tether. It includes a display unit that can be operated using AR tags and hand gestures. The robot is equipped with two lithium-ion batteries, and is rated for a maximum depth of 40 meters. As for the vision capabilities, it has

\[1\] https://www.independentrobotics.com
Figure 1.4: The initial trial of the LoCO AUV’s tetherless capability in the Caribbean Sea off the coast of Barbados.

LoCO is a Low-Cost, Open AUV, developed by the Interactive Robotics and Vision Laboratory (IRVLab) at the University of Minnesota. It is built using mostly off-the-shelf and 3D-printed parts. It has two monocular cameras for visual sensing, navigation, and data collection. It includes three thrusters and an IMU sensor that allow the robot to move forward and backward, and rotate around two of its axes that give pitch and yaw motion. LoCO AUV incorporates a ROS-based software stack and is equipped with two computing units. Specifically, it has an Nvidia Jetson TX2 for deep learning inference and a Raspberry Pi 4 for motion control. LoCO is powered by two lithium-polymer batteries and can be toggled on/off using a magnetic power switch. It includes an Organic Light-Emitting Diode (OLED) display unit and a biometric
communication device, named HREyes [31], as human-robot interfaces. Additionally, it includes a sound transducer to communicate using audible sounds. The LoCO AUV can be operated using a tether or without using a tether connection through different AUV-human interaction techniques. Fig. 1.4 shows a picture of the LoCO AUV during its initial trial to test the tetherless capability.

1.5 Overview of the Manuscript

In summary, this dissertation presents a number of perception algorithms for AUVs to enable effective underwater mHRI for successful multi-human-robot collaboration. The rest of the document is organized as follows. Chapter 2 describes a human-comprehensible gestural message recognition framework for AUVs to facilitate mHRI. Chapters 3 and 4 present diver identification systems for secure multi-human-robot collaboration. Chapter 5 introduces a diver attention estimation system for enabling effective diver-AUV interactions. And, Chapter 6 presents a diver activity recognition framework for AUVs to help them analyze and understand diver activities underwater to make informed decisions. Finally, Chapter 7 provides a summary of the presented research, outlines potential future directions, and offers our final remarks. An appendix (Appendix A) is also provided that presents additional contributions on improving the visual perception capability of AUVs and the design and implementation of a low-cost AUV.
Chapter 2

Robotic Detection of a Human-Comprehensible Gestural Language for Underwater Multi-Human-Robot Collaboration

In this chapter, we present a framework for AUVs to use vision-based perception capabilities to understand what other AUVs are communicating. This is particularly important because a number of underwater tasks, such as environmental monitoring [32], mapping [13], submarine cable and wreckage inspection [7], search-and-navigation [33, 34], will greatly benefit from human-robot collaboration. When a team of multiple AUVs and human divers are carrying out a task, they must understand one another to effectively complete the mission. A common language comprehensible to both humans and other AUVs would greatly enhance such underwater multi-Human-Robot Interaction (mHRI) missions (see Fig. 2.1).

1 This work [19] has been a Best Paper Award finalist at the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
Figure 2.1: An underwater gestural communication framework where the speaker robot is communicating back with the listener robot by making a *nodding motion* to mean YES which a human observer is also able to understand.

When designing such a communication protocol, challenges unique to the underwater domain need to be considered. Traditional sensory mediums, such as radio and other electromagnetic modalities, suffer from signal attenuation and degradation under water which limit their use to mostly surface operations. Although acoustic signals work quite well in underwater settings, these types of inter-AUV communication signals are typically incomprehensible to humans. Fulton *et al.* show that robot motion can be used to communicate with divers. Fig. 2.2 shows a sample motion-based gestural message performed by an AUV. Similarly, research on the use of motion for inter-AUV communication has shown the same for AUV-to-AUV communication. We combine these two capabilities and propose a single language for multi-human-robot communication where AUVs can visually perceive a human-comprehensible message.

We design these messages in simulated underwater environments, using Computer-Aided Design (CAD) renderings of a six-legged AUV named Aqua. Additionally, we implement the gestural messages on board the Aqua AUV using the Robot Operating System (ROS). For a robot to interpret these messages, we propose a recognition network, Robot-to-Robot Communication Network (RRCommNet), which learns salient spatio-temporal features from the gestural messages using a self-attention mechanism.
Figure 2.2: The Aqua AUV is performing ASCEND message in both simulated and real underwater environments.

After training RRCommNet on simulated and real-world data, our experiments show the recognition accuracy of RRCommNet to be approximately 94% on simulated data and 83% on real data (closed water pool environment). Based on our previous experience with computer vision-based methods in underwater field environments (e.g., [39, 40]), we are aware of the challenges in transitioning from simulated and controlled to field environments. However, training and evaluation in these environments is a necessary prerequisite to the creation and evaluation of a system in the field. We also show that we can improve the inference time of RRCommNet at a small cost to accuracy (simulated: 88%, real: 73%) by down-sampling the input video by half. Finally, through a transcription experiment, we show that humans can comprehend a conversation between two AUVs using the gestural messages with a transcription accuracy of 88%. Together, these two evaluations demonstrate that our gestural communication system can be used for accurate communication by an AUV in an m/HRI context. Thus, in this work, we:

1. propose a gestural language for AUV-to-AUV communication,
2. create an end-to-end gestural message recognition network, RRCommNet, to interpret underwater communication between AUVs,
3. perform experiments in both simulated and real underwater environments to validate the performance of the proposed gestural message recognition network, and
4. conduct a study which demonstrates that the proposed gestural language can be understood by humans.
2.1 Related Work

Underwater communication has largely focused on human-to-robot communication, i.e., regulating underwater robots based on human inputs. One common approach is to control the robot via high-speed tethered communication [18], however, direct communications are often preferred in missions as opposed to using tethered connections. A number of direct communication techniques do not even require additional robot hardware, e.g., using fiducial markers [41] or hand gestures [40, 42]. For robot-to-human communication, on the other hand, the use of small displays is by far the most popular method [43] although it has significant limitations in readability at distance, at an angle, and under low water quality. There are alternatives, such as the use of a bi-directional communication device [44] and the use of light to represent simple ideas [17], however, they either require the addition of dedicated communication devices or lack functionalities. In comparison, the use of robot motion for information [35] or affective displays for appearance-constrained robots [45] has seen encouraging results in communicating with humans.

In contrast to human-in-the-loop communication, underwater robot-to-robot communication systems are significantly less studied, primarily due to the fact that robots are not as perceptive as humans and therefore need sophisticated algorithms to understand what other robots are trying to communicate. In [46], body markings (helical drawings) are used by AUVs to communicate relative pose information. More recently, Koreitem et al. [37] propose a communication system for underwater robots where the communication messages are represented using full-body gestures and optimal variable-length prefix codes. However, this technique involves multiple steps to devise different messages and the proposed CNN learns indirectly from them. As a result, adding additional messages is not straightforward. In this work, we focus on methods, such as activity recognition that can directly learn from gesture-based messages.

Activity recognition (AR) is a well-studied problem in computer vision and robotics, with research spanning over two decades [47, 48, 49]. The goal is to predict an activity class from a large pool of human activities involving exercise, sport, instrument performance, everyday life, etc. The general solution to this problem is to learn robust spatio-temporal features from different activity classes which are contained in small
video clips. A variety of methods have been used to integrate the temporal information with the spatial, such as pooling [50], fusion [51], recurrent [52], two-stream [53, 54], and 3D architectures [55]. These techniques perform well mainly because of their ability to learn from publicly available large datasets involving human activities. The lack of such datasets is the primary contributing factor behind the absence of activity recognition techniques for robot actions in the literature. Therefore, robotic researchers often use synthetic data to validate their proposed methods before fine-tuning them for real robotic platforms [56, 57]. Furthermore, unlike traditional human activity recognition datasets, the communication messages we have designed for underwater robots have high similarities in terms of background activities and the overall appearance of the robot motions. Additionally, there are cases in which a portion of a gesture’s motion can be common to multiple gestures, leading to a strong resemblance between two gestures when considering small portions of their motion. As a result, AR algorithms that perform well for human activity recognition may not work well for underwater robots. Recent research on natural language processing suggests that one can effectively learn the underlying sequential relations by using a self-attention mechanism [58], which has also shown encouraging results in AR recently [59, 60]. Inspired by such work, we believe that highly robust spatio-temporal features can be learned from our gesture-based communication messages using a similar attention mechanism.

2.2 Gestural Recognition Framework

2.2.1 Communication Message Design

Since the purpose of using motion-based gestural communication is to enable human understanding of robot-to-robot conversations, we draw inspiration from our previous work on robot-to-human underwater communication methods [35, 36], where we have shown that humans are able to identify gestural messages with reasonable accuracy. We create a library of communication messages based on our experience working with AUVs in real underwater environments and discussions with potential end-users (e.g., marine biologists). Our library includes three types of messages: 1. Directional (D): which relate to a notification or command of some directional movement, 2. Information/Command (I/C): which provide information or give direct commands to another
Table 2.1: Definition, type, and duration of the proposed communication messages where D=Directional, I/C=Information/Command, CC=Conversation Control.

<table>
<thead>
<tr>
<th>Communication Message</th>
<th>Description and Type</th>
<th>Average Duration, s</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATTERY LOW</td>
<td>Signal low battery (I/C)</td>
<td>4.28</td>
</tr>
<tr>
<td>START COMMUNICATION</td>
<td>Begin robot-to-robot communication (CC)</td>
<td>16.71</td>
</tr>
<tr>
<td>ASCEND</td>
<td>Go up to a certain depth (D)</td>
<td>3.18</td>
</tr>
<tr>
<td>DESCEND</td>
<td>Go down to a certain depth (D)</td>
<td>3.28</td>
</tr>
<tr>
<td>FOLLOW ME</td>
<td>Instruct another robot to follow it (D)</td>
<td>6.93</td>
</tr>
<tr>
<td>DANGER</td>
<td>Danger nearby (I/C)</td>
<td>7.44</td>
</tr>
<tr>
<td>COLLECT DATA</td>
<td>Start data collection (I/C)</td>
<td>8.32</td>
</tr>
<tr>
<td>START MAPPING</td>
<td>Map the environment (I/C)</td>
<td>4.32</td>
</tr>
<tr>
<td>GO TO LOCATION</td>
<td>Go to a specific location (D)</td>
<td>7.48</td>
</tr>
<tr>
<td>U-TURN</td>
<td>Danger nearby, make a u-turn (D)</td>
<td>4.22</td>
</tr>
<tr>
<td>HELP</td>
<td>Call for help (I/C)</td>
<td>19.95</td>
</tr>
<tr>
<td>EMERGENCY SURFACING</td>
<td>Go to the surface of the water (D)</td>
<td>3.46</td>
</tr>
<tr>
<td>STOP</td>
<td>Stop doing whatever you’re doing (CC)</td>
<td>6.52</td>
</tr>
<tr>
<td>NO</td>
<td>Disagreement (CC)</td>
<td>4.31</td>
</tr>
<tr>
<td>YES</td>
<td>Agreement (CC)</td>
<td>4.26</td>
</tr>
</tbody>
</table>

robot, and 3. Conversation Control (CC): which are responses to questions or commands. Tables 2.1 and 2.2 shows the complete library of communication messages with their description, type, definition, and average duration.

2.2.2 Gesture Implementation

We first implement the gestural messages in simulated underwater environments using Unreal Engine visual programming [29], with CAD renderings of the Aqua robot. We use three self-made environments: pool, lake, ocean, and a separate ready-made ocean environment [61] as the visual environments the gestures will be performed in. To
Table 2.2: Definitions of the communication messages.

<table>
<thead>
<tr>
<th>Communication Message</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATTERY LOW</td>
<td>Roll 360° twice while moving down vertically</td>
</tr>
<tr>
<td>START COMMUNICATION</td>
<td>Roll, pitch, and yaw 45° at the same time twice</td>
</tr>
<tr>
<td>ASCEND</td>
<td>Pitch up 90° vertically and go up</td>
</tr>
<tr>
<td>DESCEND</td>
<td>Pitch down 90° vertically and go down</td>
</tr>
<tr>
<td>FOLLOW ME</td>
<td>Pitch and yaw 35° to the right, comeback to original position; do this twice and then roll 45°, make a 180° turn while unrolling and finally, go forward</td>
</tr>
<tr>
<td>DANGER</td>
<td>Yaw left and right thrice; then roll 45°, make a 180° turn while unrolling</td>
</tr>
<tr>
<td>COLLECT DATA</td>
<td>Pitch and yaw 45° to the right, comeback to original position, do the same to the left; perform this twice</td>
</tr>
<tr>
<td>START MAPPING</td>
<td>Yaw 360° while pitching up and down</td>
</tr>
<tr>
<td>GO TO LOCATION</td>
<td>Roll 45°, move forward and backward twice, then go to a location</td>
</tr>
<tr>
<td>U-TURN</td>
<td>Roll 80° and make a 180° turn while unrolling</td>
</tr>
<tr>
<td>HELP</td>
<td>Roll 45°, circle in a small loop twice</td>
</tr>
<tr>
<td>EMERGENCY SURFACING</td>
<td>Pitch up 90° and roll 360° twice</td>
</tr>
<tr>
<td>STOP</td>
<td>Yaw 360° twice before stopping</td>
</tr>
<tr>
<td>NO</td>
<td>Yaw left and right twice</td>
</tr>
<tr>
<td>YES</td>
<td>Pitch down and up twice</td>
</tr>
</tbody>
</table>
implement the messages in real underwater environments with the Aqua robot, we create a ROS package named \texttt{rrcomm} which operates as a gestural message generator. The generator can produce 15 communication messages using a user-defined configuration file, an example of which can be seen in Fig. 2.3 for the START MAPPING message. The configuration file contains a definition of the gesture as a set of timed motions expressed in roll, pitch, yaw, surge, and heave velocity percentages. Our ROS node parses these configuration files to implement the robot maneuvers using the robot motion controller \cite{62}.

\subsection*{2.2.3 Gestural Message Dataset}

To train a neural network capable of recognizing our gestural messages, we first create a dataset by recording the 15 defined gestural messages (see Tables 2.1 and 2.2) as full-color videos in both simulated and real underwater environments. To improve the robustness of our simulated training data, we introduce a number of variations within the simulated environment and the robot model, giving rise to 25 different environmental conditions. Specifically, we vary the surface texture, object type, hydrodynamics, water visibility, and color of the robot while keeping the message definitions unchanged. For real-world data, we record 5 instances of each message performed in a closed-water underwater environment. For the HELP message, the Aqua robot’s Proportional-Integral-Derivative (PID) controller was overshooting the requested target angles, causing the
robot to stray away from the defined circular path, which made this implementation invalid for use. In summary, there are a total of 445 recordings of the 15 communication messages, of which 375 are synthetic data and 70 are real-world data. We reserve 73% of recordings reserved for training (300 synthetic, 28 real) and the remaining 27% for testing (75 synthetic, 42 real).

2.2.4 Robot-To-Robot Communication Network (RRCommNet)

The RRCommNet network accepts videos of robot gestures as inputs, having dimensions of \( T \times C \times H \times W \), where \( H, W, \) and \( C \) are height, width, and color channels of the input, respectively. \( T \) is the length of the input chunk in frames. We introduce a skipping mechanism for faster training and inference; if enabled, it uses every other frame from the input chunk (\( i.e., \frac{T}{2} \) number of frames). This gives rise to two different models, \( RRCommNet \) and \( RRCommNet-Skip \), both of which look at the same temporal context where the latter simply handles a temporally downsampled version of the input. We encode the input using a highly modularized deep neural network architecture, ResNeXt-101 [63], which outputs a feature representation of shape \( (T' \times C' \times H' \times W') \). We choose ResNeXt-101 as a feature extractor because it can extract highly robust spatio-temporal features by utilizing a \textit{split-transform-aggregation} strategy. With this strategy, the input is split into a few lower-dimensional embeddings, transformed as a group (in a new dimension called the \textit{cardinality} of the architecture), and merged by concatenation. This gives the network the representational power of large and dense layers but at a considerably lower computational complexity.

Now, we need to address two issues unique to the gestural messages while designing our recognition framework. First, the gestural messages have high similarities in terms of background activities and the overall appearance of the robot motions. To address this, we perform average pooling in the spatial dimension while normalizing against the channel dimension on the extracted features to get a narrow but long feature representation, having a dimension of \( (T' \times C') \). This will give less importance to the background and appearance information. The second and more significant issue is the commonality between portions of pairwise gestures; a strong resemblance can confuse the networks when considering small durations of such pairs of gestures. We address
Figure 2.4: The proposed gestural message recognition framework, RRCommNet. It takes a video clip as input and extracts spatio-temporal features. Self-attention mechanism is performed on the feature representation to get the final classification scores.

This by using a self-attention mechanism (proposed in [58]) with bi-directional Transformers (BERT [64]), as was suggested in [59, 60]. With self-attention, the network will be able to focus on the most salient features while BERT enables temporal information from both directions. Taking these together, the network is able to look into the most salient features within a longer temporal limit enhancing the messages’ contexts. With these choices in mind, we design RRCommNet (see Fig. 2.4) as described below.

First, we take a query feature \( (Q) \) from the extracted feature representation (computed using ResNeXt-101) to compare against all other positions in the feature representation (considered as key, \( K \)). For a gestural message, \( Q \) is the small spatial information with a large temporal context. Both query and key are of size \( d_{\text{model}}(= C') \). The comparison gives rise to a weighted sum of values \( (V) \) which are the self-attention scores. Note that to learn the robust feature representations of \( Q \) and \( K \), we use linear projection to project them to a lower dimension \( d_k \). Specifically, we compute the self-attention
as follows:

$$a_{self} = \frac{QK^T}{\sqrt{d_k}}$$

$$a_{weights}^i = \frac{\exp(a_{self}(\cdot, i))}{\sum_{j=1}^{d_k} \exp(a_{self}(\cdot, j))}$$

$$A^i_{self} = a^i_{weights}V$$

where, $A_{self}$ is the weighted self-attention scores.

Note that we use the attention mechanism in a parallel fashion, i.e., using multiple attention heads ($h$), and set $d_k = \text{floor}(d_{model}/h)$. The multi-head attention scores are fed through a two layer position-wise feed-forward network (P-FFN) to get the final self-attention scores to represent the salient features from a gestural message. P-FFN is defined as, $\text{P-FFN}(A_{self}) = \max(0, A_{self}W_1 + b_1)W_2 + b_2$, where $W_i$ and $b_i$ are the weights and biases of the respective layers.

Finally, a single linear layer is used to get the final classification scores which has exactly $N$ neurons, where $N$ is the total number of communication messages. Note that we perform the final prediction using the classification embedding which we include on top of the feature representation for better classification, as suggested in [64, 60]. Additionally, we include a location embedding to all the locations in the feature representation in order to incorporate positional information in the attention scores. These embeddings are set as learnable parameters. Moreover, while performing the self-attention mechanism, we randomly mask 10% of the feature locations and set their attention scores to zeros which incorporates the bi-directional context learning from BERT.

### 2.2.5 Implementation Details

We use PyTorch [65] libraries to implement RRCommNet, with input spatial resolution of $320 \times 256$ and temporal resolution of 64 or 32 (with input skipping). We follow various data augmentation schemes, e.g., multi-scale cropping, random horizontal flipping, and normalization, before training the model as described in [66]. For training the overall network, we use similar settings as described in [60] with the following modifications. We choose batch size of 8, learning rate of $10^{-4}$ with a scheduler, and ADAMW as the optimizer [67]. We use ResNeXt-101 with a cardinality of 32 without the last
two layers. The P-FFN uses ReLU non-linearity and has 2 layers with 2048 and 512 neurons, respectively. The dropouts are set to 0.1. We use one transformer block with four attention heads for the self-attention mechanism. We use $d_{\text{model}} = 512$ and $d_{k} = 128$. A classification embedding is added on top of the feature representation and a location embedding is added to all the locations in the feature representation in order to incorporate positional information. Both of which are initialized using the normal distribution $\mathcal{N}(0, 0.02^2)$. We have trained the network on an Nvidia GeForce RTX 2080 GPU for 200 epochs with cross-entropy loss and noticed convergence in validation loss, top-1 accuracy, and top-3 accuracy. The network is evaluated on an Intel® Xeon® E5-2650 CPU for realism of inference time results.

### 2.3 Human Transcription Study

To demonstrate that the proposed gestural language can be effectively understood by humans in an mHRI mission, we undertake a study of humans transcribing the conversations of robots using our gestural language, using the QualtricsXM survey platform. First, the participants are taught the messages in a random order. Then, they are asked to transcribe the conversation shown to them in a simulated video of two robots conversing. A total of 10 such conversations are shown from one of three different viewpoints: head-on, rotated by 90 degrees of yaw, and rotated by 90 degrees of pitch. Our population of 34 participants are randomly assigned to one of these viewpoints, asked to select the displayed gesture’s meaning from a drop-down list for each message, and rate their confidence in their answers for each transcription from zero to 10. Table 2.3 shows the results of the study. As we can see from the table, the participants are able to correctly transcribe the conversation with an average accuracy of 88.20% and confidence of 7.9 (out of 10). Here, avg. accuracy $= \frac{1}{34} \sum_{p=1}^{34} \frac{\text{correct selections}_p}{\text{total shown}_p}$. The participants seem to struggle the most with the messages which RRCommNet performs poorly on (as discussed in Sec. 2.4.2). Closer inspection of such messages indicates that they include gestures which have high visual similarity. The results from this study show that humans can comprehend our proposed gestural language, which, along with the results

\footnote{Study (reference no. 00012959) has been reviewed and approved by the University of Minnesota’s Institutional Review Board.}
Table 2.3: Results of the human transcription study in recognizing different gestural messages. The values represent $\text{avg. trans. accuracy (\%)} / \text{avg. confidence (out of 10)}$.

<table>
<thead>
<tr>
<th>Gest. Message</th>
<th>BATTERY START</th>
<th>COMM. ASCEND DESCEND</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Trans.</td>
<td>94.10/8.30</td>
<td>94.10/7.30 97.10/8.30</td>
</tr>
<tr>
<td>Gest. Message</td>
<td>FOLLOW COLLECT START</td>
<td>ME DANGER DATA MAPPING</td>
</tr>
<tr>
<td>Human Trans.</td>
<td>97.10/8.60</td>
<td>72.10/7.80 85.30/8.30</td>
</tr>
<tr>
<td>Gest. Message</td>
<td>GO TO EMERGENCY LOCATION U-TURN HELP SURFACING</td>
<td></td>
</tr>
<tr>
<td>Human Trans.</td>
<td>82.40/8.10</td>
<td>87.10/6.40 91.20/8.00</td>
</tr>
<tr>
<td>Gest. Message</td>
<td>STOP NO YES Overall</td>
<td></td>
</tr>
<tr>
<td>Human Trans.</td>
<td>73.50/6.40</td>
<td>94.10/8.20 96.30/8.20</td>
</tr>
</tbody>
</table>

in the following section, fulfills our goals of a gestural language that can be understood by both robots and humans.

2.4 Evaluation Of RRCommNet

2.4.1 Evaluation Procedure and Metrics

The test videos are processed as chunks of either 64 or 32 frames (with skipping). We make the final prediction on 10 different cropped versions of the input video clip. We follow the same cropping mechanism as described in [66], i.e., four crops from the corners, one from the center, and the same on a flipped version. The crops have a spatial dimension of $112 \times 112$. Therefore, a single RGB test batch is a 5-dimensional tensor of shape $[10, 64, 3, 112, 112]$ or $[10, 32, 3, 112, 112]$ (with skipping).

The test batch is fed to the trained RRCommNet model to get 10 prediction scores.

---

3 In this work, we use prediction and recognition interchangeably.
as a tensor ($x_{\text{preds}}$) of shape (10, 15). We average the scores to get our final prediction vector, $x_{\text{mean}}$. Finally, we calculate the predicted class probability using the softmax function, as shown below:

$$P(x_{\text{mean}}^i) = \frac{\exp(x_{\text{mean}}^i)}{\sum_{j=1}^{N} \exp(x_{\text{mean}}^j)}$$  \hspace{1cm} (2.1)

where $i$ refers to the $i$-th class, $N$ is the total number of communication messages, and $i = 1, 2, \ldots, N$. Each prediction class is found by selecting the index of the maximum class probability.

We consider three metrics for quantitative evaluation of RRCommNet:

1. **Recognition Accuracy**: it is the ratio between correct predictions and total instances (for each message).

2. **Recognition Probability**: it is the softmax probability, as defined in Eq. 2.1, that shows the confidence for a correct prediction (for each message).

3. **Inference Time**: it is defined as the time it takes for each prediction (reported on CPU unless otherwise specified).

### 2.4.2 Results

**RRCommNet Outperforms SOTA For AUV Gestures**

First, we compare the performance of the RRCommNets against the State-Of-The-Art (SOTA) action recognition models in terms of average recognition accuracy. From Table 2.4, we see that for simulated data, RRCommNet achieves an average recognition accuracy of 94.67% which is exactly same as the SOTA model, LateTemporal3DBert [60]. In comparison, the SlowFast [54] model, which is another robust action recognition model, does not achieve comparable performance (accuracy 74.67%) against our method. As for real data, RRCommNet achieves an average recognition accuracy of 83.33% which is significantly better than the rest of the methods. In contrast, RRCommNet-Skip achieves an average recognition accuracy of 88% on simulated data which is higher than the SlowFast method but only comparable against the SOTA or RRCommNet. For real data, however, RRCommNet-Skip shows superior performance than the SOTA. As a
Table 2.4: Comparison of the gestural message recognition accuracy on both simulated and real data.

<table>
<thead>
<tr>
<th>Method</th>
<th>Avg. Recognition Accuracy (%)</th>
<th>Simulated</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td>SlowFast</td>
<td>74.67</td>
<td>64.29</td>
<td></td>
</tr>
<tr>
<td>LateTemporal3DBert</td>
<td>94.67</td>
<td>71.42</td>
<td></td>
</tr>
<tr>
<td>RRCommNet</td>
<td>94.67</td>
<td>83.33</td>
<td></td>
</tr>
<tr>
<td>RRCommNet-Skip</td>
<td>88.00</td>
<td>73.81</td>
<td></td>
</tr>
</tbody>
</table>

Note, we were unable to make a comparison with the underwater inter-robot communication framework presented in [37] because the authors in that paper use a 3D pose regressor as the visual decoder whereas we use a self-attention based classifier as our visual decoder.

RRCommNet Confusion Matches Human Confusion

First, we evaluate the message recognition performance of both RRCommNet and RRCommNet-Skip by feeding our test data (as described in Sec. 2.2.3) to both networks and analyzing the outputs. Fig. 2.5 shows a snapshot of RRCommNet predicting the U-TURN message, performed by the Aqua AUV in actual underwater environment. The predictions are accumulated in a confusion matrix and further augmented by the human transcription of robotic conversation results (as described in Sec. 2.3). Fig. 2.6 shows the complete confusion matrix where the values in different cells are normalized (refer to the color bar to see the non-normalized values). From the figure, we see that DANGER and START MAPPING are relatively difficult to recognize for humans and also by the RRCommNets. Analyzing the gesture implementations, we notice that DANGER and START MAPPING have a resemblance to NO and STOP, respectively. Noting this, we suggest that pre-deployment analysis of spatial similarity of gestures could optimize both robot and human comprehension of gestures by avoiding overlaps in spatio-temporal fragments.
Figure 2.5: Gestural message recognition using RRCommNet. (a) Prediction is being performed on the 8th crop. (b) Final prediction is made by averaging the recognition probabilities on all 10 crops.

**RRCommNet-Skip Outperforms RRCommNet in Speed**

Finally, we evaluate the performance of the RRCommNets in terms of their prediction confidence and speed. From Table 2.5, we see that both RRCommNets are fairly confident while making gestural message predictions, having overall average recognition probability of 78.97% and 78.88%, respectively. As for recognition speed, we see that RRCommNet-Skip is notably faster than RRCommNet. As a matter of fact, both the networks display fast inference times on CPU. For example, inference times for BATTERY LOW and STOP messages are 0.55s and 1.32s, respectively using RRCommNet and 0.34s and 0.90s, respectively using RRCommNet-Skip. Therefore, we can choose RRCommNet-Skip where inference time is important, and choose RRCommNet where accuracy is the main requirement.
Figure 2.6: Confusion matrix for the gestural message recognition. It contains 5 different experiments: RRCommNet in both simulated and real data, RRCommNet-Skip in both simulated and real data, and human transcription.
Table 2.5: Comparison between the performance of RRCommNet and RRCommNet-Skip in recognizing different gestural messages. The values represent *avg. recog. probability (%)*/avg. inference time (s).

<table>
<thead>
<tr>
<th>Gest. Message</th>
<th>BATTERY</th>
<th>START</th>
<th>LOW</th>
<th>COMM.</th>
<th>ASCEND</th>
<th>DESCEND</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRCommNet</td>
<td>95.53/0.55</td>
<td>78.16/3.48</td>
<td>97.5/0.51</td>
<td>96.94/0.59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RRCommNet-Skip</td>
<td>95.78/0.34</td>
<td>80.93/2.37</td>
<td>93.74/0.34</td>
<td>98.16/0.35</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gest. Message</th>
<th>FOLLOW</th>
<th>COLLECT</th>
<th>ME</th>
<th>DANGER</th>
<th>DATA</th>
<th>MAPPING</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRCommNet</td>
<td>63.65/1.61</td>
<td>47.93/2.37</td>
<td>96.85/1.68</td>
<td>65.00/0.51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RRCommNet-Skip</td>
<td>37.68/0.94</td>
<td>-</td>
<td>98.48/1.01</td>
<td>75.63/0.34</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gest. Message</th>
<th>GO TO</th>
<th>EMERGENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCATION</td>
<td>U-TURN</td>
<td>HELP</td>
</tr>
<tr>
<td>RRCommNet</td>
<td>86.68/1.67</td>
<td>64.06/0.75</td>
</tr>
<tr>
<td>RRCommNet-Skip</td>
<td>78.81/1.03</td>
<td>55.61/0.55</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gest. Message</th>
<th>STOP</th>
<th>NO</th>
<th>YES</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRCommNet</td>
<td>66.9/1.32</td>
<td>73.1/0.67</td>
<td>94.3/0.66</td>
<td>78.97/1.45</td>
</tr>
<tr>
<td>RRCommNet-Skip</td>
<td>61.08/0.90</td>
<td>63.14/0.41</td>
<td>98.07/0.47</td>
<td>78.88/0.80</td>
</tr>
</tbody>
</table>
2.4.3 Runtime Performance on Embedded GPU

As established in the previous section, there is a trade-off between inference time and accuracy with RRCommNet and RRCommNet-Skip, as shown in Table 2.5. However, these inference times are on a powerful CPU (an Intel® Xeon® E5-2650), processing power that most AUVs will not have available on-board. It is, therefore, worthwhile to consider runtime performance on a processor more likely to be used in an AUV, such as the Nvidia TX2 (already in use in a number of AUVs, Aqua among them). To test on the TX2, we simply run our evaluation experiments a second time, recording the inference time for each video, calculating the per-frame inference time for both networks, and averaging across all videos. RRCommNet operates at an average inference speed of 10 FPS and RRCommNet-Skip performs inference at 12.5 FPS. Total inference time for any gesture depends on the length of the gesture, but this rate of inference allows the recognition of gestures at sufficient speeds to be useful in the field.

2.5 Summary

In this chapter, we have presented the design and implementation of a unified gestural language for underwater mHRI missions and shown that an attention mechanism-based deep network can recognize a gestural language for AUVs in both simulated and real environments. Our networks, RRCommNet and RRCommNet-Skip, demonstrated recognition accuracy which outperforms SOTA activity recognition methods. Both networks also have inference speeds on embedded GPU which are sufficient for on board robot deployments, though RRCommNet-Skip is somewhat faster. Through a human transcription study, we have also demonstrated that the proposed gestural language is understandable to humans. Having shown our framework to be comprehensible to both humans and robots, we also suggest that our framework is ideal for integration into existing diver interaction workflows. The majority of underwater communication between divers happens via gestures (with some exceptions). In much the same way, human-to-robot control methods underwater often utilize gestures [40,42]. Thus, a gesture-based robot-to-robot language fits nicely into the overall workflow of underwater communication. On the technical side, our ROS-based implementation makes the process of integrating our method into any ROS-powered AUV relatively straightforward. Lastly,
implementation of the gestures in our language via simple configuration files makes the transition from designing a gesture to implementing it as simple as writing out the desired movements. While programming experience will still be required to fine-tune the movements, this implementation will allow for the easy creation of new gestures when they are required. Taken together, our method’s SOTA recognition performance, reasonable inference times, human comprehensibility, conformity with existing robot and human underwater communication methods, and easy generation/reconfiguration of gestures make it an excellent candidate for deployment in the field. This enables the opportunity to use our gestural language and language recognition network for AUV-to-AUV communication in underwater mHRI missions, so that every party to the interaction, whether human or robot, can understand every part of the conversation.

In the next chapter, we will present a diver identification framework that extracts unique facial features from diver faces by synthetically removing the mask and breathing apparatus. This framework will enable AUVs to take instructions from and collaborate with correct individuals.
Chapter 3

Towards Visual Diver Face Identification by Autonomous Underwater Robots

Underwater human-robot communication remains an open challenge, particularly when robots are required to identify specific individuals within a group of divers and communicate with them. Unable to uniquely identify divers may cause mission failures and compromise secure Human-Robot Interaction (HRI). The latter concern is significant in most underwater missions (e.g., pipeline and cable inspections, search and rescue) but is particularly crucial to maritime defense applications. Recent work [68, 69] have made some inroads in uniquely identifying underwater scuba divers from their swimming gaits. Still, these algorithms are inapplicable for close-proximity interactions as they require the entire body to be visible and in motion.

In this work[^1], we propose a framework that can correctly identify scuba divers in close-proximity interactions using facial cues only (see Fig. 3.1). Despite recent facial recognition methods (e.g., [70, 71]) achieving a high degree of accuracy, we are yet to see a study addressing the underwater facial recognition problem. A number of significant challenges contribute to this situation. First, unlike traditional facial recognition tasks, distinctive components (e.g., eyes, nose, and mouth) of the diver faces are heavily

[^1]: In this project, I have been the joint-lead student investigator with another Ph.D. student (during the collaboration) named Jungseok Hong.
Figure 3.1: An example of underwater human-robot interaction scenario. When an AUV positions itself at a suitable distance for the interaction, our proposed system performs diver identification. Accurate identification ensures AUVs carry out instructions from authorized divers only.

obscured by their masks and regulators. As a result, the State-Of-The-Art (SOTA) algorithms, despite their efficiency in learning discriminative features from regular faces, fail to learn these highly occluded diver faces. Second, the underwater domain poses significant challenges to visual perception due to its unique optical properties \cite{72} (e.g., distortion, back-scatter, color-degradation, etc). Third, there is no publicly available dataset of diver faces; this limits the possibility of using supervised end-to-end Deep Neural Networks (DNNs) to recognize scuba divers. Furthermore, it is impossible to create such a dataset using the publicly available images of scuba divers because the identity information (i.e., corresponding regular faces without dive apparatus) of the divers are mostly unknown. However, such information is essential to train a face identification DNN.

We present a face identification system which is able to uniquely recognize scuba divers in spite of diving apparatus obscuring their facial features. Once an AUV properly positions itself with respect to a diver for interaction, our proposed system performs
diver identification. To benefit a diver face feature extractor, we formulate a synthetic diver face (i.e., faces with mask and breathing apparatus) dataset through a series of data augmentation steps, taking the unique underwater effects into account. First, we use a convolutional autoencoder \cite{73} to extract the facial features from a person’s regular (i.e., faces without mask and breathing apparatus) and diver faces. Since the features of a diver face contain additional mask, regulators (or snorkels and other breathing apparatus), and information from the underwater environment, we perform a dimensionality reduction technique that automatically selects features which contribute to facial cues only. Our experiments on 27 real-world scuba diver faces show an identification accuracy of around 51% which is substantial because we do not use any real-world data to train our algorithm. Furthermore, unlike some recent masked face recognition algorithms \cite{74,75} which use synthetically created datasets to report the test accuracy, we perform the evaluation solely on real-world data. Our work is the first to address this unique challenge of diver recognition through facial features for underwater HRI tasks.

In this work, we:

1. propose a diver face identification system that can identify scuba divers using facial cues only,

2. develop a data augmentation technique to synthetically create diver faces from regular faces and construct a dataset that includes both regular and diver faces of same identities, and

3. perform several quantitative and qualitative experiments that validate the efficacy of the proposed system in correctly identifying scuba divers, and analyze the practical feasibility of this framework for mobile platforms.

### 3.1 Related Work

Recent advances (e.g., hand gestures \cite{76}, fiducial tags \cite{77}) in underwater HRI have allowed divers to communicate with robots without using a tethered underwater control device \cite{44}. With the advances in vision-based communication between underwater robots and divers, underwater human-robot joint missions (specifically, missions requiring multi-Human-Robot Interaction or mHRI) have become feasible. Thus, the ability
of robots to identify individual divers in such teams has become imperative. Existing vision-based diver identification work \cite{68,69} use motion characteristics to recognize divers, which would work neither in close proximity nor in the case of immobile divers. A possible solution would be to use facial cues for identification. In the literature, human subject identification using faces has been widely studied for terrestrial robots \cite{78,79}. In contrast, diver identification using faces has received little attention due to the unique challenges present in the underwater domain.

In recent years, deep convolutional neural networks have yielded high-accuracy in face recognition tasks \cite{80,81,71}. In general, most of these methods include the following steps: 1. **Face detection**: by learning feature pyramid \cite{82}, using a “proposal and refinement” mechanism \cite{83}, or densely sampling locations across multiple scales \cite{84}; 2. **Face classification**: achieved by either training a multi-class classifier to classify different identities in the training set using softmax loss \cite{85} or directly learning the face representation (i.e., embeddings) to classify different identities which may not be present in the training set \cite{86}. For the face recognition task, the primary objective is to ensure compactness in *intra-class distances* (i.e., the same faces) while maximizing the *inter-class distances* (i.e., different faces) \cite{87,88}.

Generally, traditional methods, discussed above, exhibit poor performance on partially-occluded faces. The work in \cite{89} proposed to extract local face descriptors only from the non-occluded facial areas. In \cite{90}, a deep network is used as a feature extractor to classify faces from partial data. However, the discriminative power of these methods is limited. In \cite{91}, the authors code the occluded face image as a sparse linear combination of the training samples and the occlusion dictionary. But, it fails to generalize because of the assumption that test samples have identical subjects as the training samples. In contrast, the work in \cite{92} trains a deep network to detect facial keypoints from partial face images and then the angles between the keypoints are used to perform face identification. However, none of these methods show how to tackle significant facial occlusions as those occurring from underwater breathing apparatus. In addition, the work in \cite{93} shows that the network trained on a large dataset is unable to achieve 50% accuracy when faces are cropped by more than 40%. This research gap is mostly due to difficulty in collecting real-world diver face datasets, distortions present in underwater imagery, and the complex facial occlusions occurring on diver faces.
Figure 3.2: Data augmentation steps for preparing the training dataset for the facial feature extractor network. (a) Frontalization. (b) Automatic crop. (c) Color correction and keypoints regression. (d) Apply mask and regulator based on the 15 keypoints. (e) Colorization [4] to incorporate underwater lighting conditions. (f) Applying underwater effect [5]. (g) Crop to tighten the boundary.

3.2 Methodology

The overall pipeline of the proposed approach consists of both online and offline components. Once trained, the inference process runs entirely on-board the AUV.

3.2.1 Offline Module

Data Augmentation

A significant challenge in the diver face identification task is the unavailability of diver face datasets. To circumvent this issue, we resort to using publicly available traditional regular face datasets and perform a series of operations to transform them into diver faces. Our augmented face dataset thus includes both regular and diver face pairs. First, we perform frontalization to the faces, as described in [94]. That is, given a query photo, we detect facial features on both the query photo and a rendered 3D face model. The 2D coordinates on the query image and the corresponding 3D coordinates on the 3D model enable us to estimate a projection matrix to project the 2D keypoints from the query photo onto a reference 3D coordinate system. Finally, the missing pixels’ color intensities are filled in with the original query image’s symmetric regions. We perform an automatic crop on the processed image to retrieve the original composition of the image. The motivation behind the frontalization step stems from the fact that the scuba divers will likely be in a frontal-looking position while directly interacting
Next, we use a facial keypoint prediction network, which uses four convolutional layers and three fully connected layers to output 15 \((x, y)\) keypoint coordinates. These 15 keypoints are used to put different kinds of masks and regulators on top of the faces to make them look like diver faces. We process the masked images further to achieve the underwater green/blue hue [4]. In addition, to account for the distortions, such as barrel distortions [95] and longer focal lengths caused by underwater optics, we add some underwater effects [5] to the face images so that they closely resemble the pictures taken underwater. Fig. 3.2 demonstrates all the steps of the data augmentation process.

**Feature Extraction Method**

To extract discriminative features from facial images, we use a convolutional autoencoder [73]. The autoencoder first encodes the face as a compact hyperdimensional feature vector from which it decodes the feature vector to reconstruct the face. The convolutional layers of the autoencoder help to capture features from the images. Essentially, the network learns to obtain a highly discriminative 512-dimensional feature embedding to represent a face. We vary the composition of the training dataset (e.g., regular + diver faces and diver faces only) to see the impact on the reconstructed faces. Moreover, we vary the size of the encoder output to create different dimensional feature embeddings for comparison purpose. Fig. 3.3 shows two sample outputs from the autoencoder where it tries to reconstruct the faces of two participants.

Figure 3.3: Reconstructed faces of two participants. Left side of each pair is the original image, and the right side is the reconstructed one: (left pair) diver face and (right pair) regular face.

with an AUV.
Figure 3.4: An example of variance analysis for dimensionality reduction. The blue and the green values represent variance values for regular and diver faces, respectively. The variance values for both regular and diver faces from dimensions 3 and 6 are more than a predefined threshold value (0.25), making these two dimensions desirable for representing regular and diver faces.

**Variance Analysis for Dimensionality Reduction**

To correctly match a person’s diver face with their regular face, the diver’s face without the mask and the breathing apparatus would be ideal to have. We achieve this by performing a variance analysis (VA) on both regular and diver faces to find the feature dimensions that contribute to highly discriminative facial cues only (i.e., highly variant dimensions). First, we use the autoencoder to extract 512-dimensional feature embeddings from the faces (regular + diver) in our dataset. With this embedding, we compute the variance of all 512 feature values across all the regular faces which gives us a 512-dimensional variance vector. We perform a similar computation on diver faces to yield a similar length variance vector. Second, we compare these two 512-dimensional variance vectors to find the highly variant dimensions. Specifically, we use a predefined...
threshold to find these dimensions. To find the threshold, we start at the initial value of 0.009 and increment by 0.005 to generate a set of thresholds. Then, we use values from this set to lower the feature dimensions of the faces from the training dataset and calculate the identification accuracy with the reduced features. We pick the threshold value which gives the highest identification accuracy. This way, we find a lower-dimensional \((R-d, R << 512)\) feature representation for both diver and regular faces which contain highly variant feature values. Our hypothesis is that VA converts the original feature embedding (i.e., encoded output from the autoencoder) to a lower-dimensional feature representation in a way that discards the feature dimensions which correspond to mask, snorkel, and underwater artifacts. This will also require less memory to store the feature representations. Fig. 3.4 shows an example of the VA method for 8-dimensional feature representations. From the figure we see that only the third and the sixth dimensions appear to be highly variant based on the set threshold value. Therefore, the final feature representations of diver and regular faces should have only these two dimensions.

**Diver Face Identification Database**

To correctly identify a scuba diver for a secure HRI session, an AUV must know beforehand who the particular individuals are to interact with. To serve this purpose, we construct a diver face identification database that stores feature embeddings for regular faces of the *authorized* divers. Specifically, we store the lower-dimensional embeddings, computed as described in Sec. 3.2.1, of the regular faces. We store at least two facial feature embeddings for each subject to minimize the false negative rate. Note that we do not store any information of the diver faces in the identification database.

**3.2.2 Online Module**

**Diver Face Localization**

In the mHRI context, an AUV needs to authenticate a diver before they are authorized to issue commands. First, the AUV needs to localize or detect diver faces. To this end, we found that RetinaFace [96] performed the best in localizing diver faces in underwater images compared to other face detection algorithms (e.g., [97, 98]). RetinaFace is a single-stage pixel-wise face detector that employs a joint extra-supervised
Figure 3.5: Complete pipeline of the proposed diver face identification system. The overall system consists of an online as well as an offline component. In real-world scenarios, only the online module is executed by AUVs to recognize scuba divers, e.g., to ensure they are allowed to operate the AUVs.

and self-supervised multi-task learning strategy to localize faces at various scales.

**Diver Face Feature Extraction and Matching**

Once the AUV has detected a diver face from the captured image, it will extract facial feature embedding from the diver face using the method described in Sec. 3.2.1. Since we already know what the important dimensions are (from Sec. 3.2.1) which ignore mask, snorkel, and underwater information, we represent the diver face feature embedding using the feature values in those dimensions only. Then, this embedding is compared against all the stored embeddings in the diver face identification database. If a match is found, the AUV would interact with the diver.

Fig. 3.5 shows the complete pipeline of the proposed diver face identification framework.
3.3 Experimental Setup and Evaluation

3.3.1 Dataset

Training Set

There are 184 different subjects in the dataset, collected from the following sources [99, 100]. Each subject has multiple regular face images. We augment the dataset by putting four different types of masks and two different types of regulators on top of the regular faces. This gives us a total number of 22,031 (diver + regular) samples of 184 different subjects. We use different combinations of masks and regulators so that our feature extraction model learns generalized discriminative face features. Finally, we carry out the steps described in Sec. 3.2.1 to prepare the final dataset. We show a few sample images from the prepared dataset in Fig. 3.6.
Evaluation Set

We conducted a human study\footnote{The study (reference no. 00013497) was reviewed and approved by the University of Minnesota’s Institutional Review Board.} with 27 participants to create an evaluation dataset for the proposed method. We took their facial photos while they were in the following conditions: 1) fully submerged in the pool wearing a mask and a snorkel and 2) outside of the pool without dive apparatus. Diver face images are collected from three different pool environments, under different lighting conditions, and at various water depths. Even though our augmented dataset has regulators on faces while the participants were wearing snorkels, our proposed method appears to be invariant to the type of the breathing apparatus.

3.3.2 Evaluation Criteria

We evaluate the performance of the proposed approach on real-world data by comparing a query diver face against the stored regular faces in a feature space. Specifically, a query embedding is compared against the stored embeddings by calculating the cosine similarity (CS) \cite{cosine_similarity} between them. The CS metric provides better distinction between our test images than other similarity measures (Euclidean and Manhattan distance). The CS between two feature vectors $f_q$ (embedding of the query face) and $f_a$ (embedding of the stored face) is defined as,

$$CS(f_q, f_a) = \frac{f_q^T f_a}{\|f_q\| \|f_a\|}$$

Our algorithm uses the CS values to classify the ID of the query embedding by selecting the ID from stored embeddings which gives the highest CS. We check if they are the same subject. If they are the same, the prediction is considered correct. To calculate the accuracy of the predictions, we take a number of test samples (i.e., real-world diver faces) to match them against the stored subjects and keep a count of the correct predictions. Finally, we divide this number by the total number of test samples to get the final prediction accuracy.
3.3.3 Model Selection and Evaluation Results

To evaluate the performance of our approach against other methods, we conduct two baseline experiments in addition to our approach on the evaluation datasets. In the first experiment, we test SOTA face recognition algorithms to extract features. In the second experiment, we perform domain transfer (diver-to-regular and vice versa) on the evaluation dataset first and extract the features with the face recognition algorithm. Lastly, we use our method to extract features from the dataset. We provide details in this section how we perform feature extraction for each case.

Face Recognition Algorithms

We select ArcFace [70] and FaceNet [71] as baseline face recognition models because they are two of the most accurate face recognition algorithms. We train both models with our augmented dataset (see Sec. 3.3.1). We choose the following parameters for the networks. For ArcFace, we use a batch size of 128 and an input size of $112 \times 112$. As the optimizer, we choose stochastic gradient descent with a learning rate of 0.01 and a momentum of 0.9. For FaceNet, we select a batch size of 128 and an input size of $160 \times 160$. We train the network using Adagrad optimizer with a learning rate of 0.1 and a learning rate decay factor of 1.0. From our tests, ArcFace and FaceNet show 14.8% and 18.5% accuracy, respectively. During our tests, we see that facial embeddings of size 512 converges to the lowest validation loss and therefore, we use 512-dimensional feature embeddings to represent faces. The number of training parameters (ArcFace: 40M and FaceNet: 140M) of both networks enable them to learn facial features from large datasets (e.g., VGGFace2 [102] (3.31M) and MS-Celeb-1M [103] (1M)), but they overfit with our training dataset since it is much smaller compared to the large datasets.

Domain transfer and Face Recognition Algorithm

To minimize the error coming from the discrepancy between stored regular faces (ground domain) and query diver faces (underwater domain), we apply domain transfer from both sides: 1) Mask Off: domain transfer from diver faces to regular faces and 2) Mask On: domain transfer from regular faces to diver faces. We choose CUT [104], which
is a generative model to perform domain transfer. With this approach, we can extract features from faces within the same domain (i.e., underwater vs underwater, ground vs ground). We use the following parameters to train CUT: batch size of 1, Adam optimizer with learning rate of 0.0002, $\beta_1 = 0.5$, and $\beta_2 = 0.999$. While it shows improved performance compared to the former experiment in the case of “Mask On” (25.9%), it is prone to failure when the generation is not reliable. Fig. 3.7 shows a few instances when generation from CUT failed and worsens the identification process. “Mask Off” shows much worse performance (7.4%) compared to other experiments since the generation process of regular faces were susceptible to many factors (i.e., type of faces, mask color, snorkel shape, etc). Furthermore, the generated faces often have smudged facial features (e.g., nose and mouth), and this degrades the performance of this approach.
Figure 3.8: Diver face identification accuracy results evaluated on real-world data with different approaches. Our approach with VA shows the highest accuracy among all the approaches.

**Proposed approach**

For our feature extractor, we use a convolutional autoencoder [73] with both regular and diver faces from our augmented dataset. The model parameters are 19M and 20M for the encoder and the decoder, respectively. We use a batch size of 128, and we train the network with Adam optimizer, having a learning rate of 0.001. We perform VA to find a highly variant 194-dimensional feature embedding (instead of 512) to represent faces. As can be seen from Fig. 3.8, our approach achieves an accuracy of 51.8% in correctly identifying divers, which is higher than all other comparing methods. Our experiments with smaller initial embeddings, e.g., 128- and 256-dimensional, show their best accuracies of 25.93% and 37.04% even after performing VA.
3.3.4 Practical Feasibility

The accuracy of the proposed method is around three times higher than the SOTA face recognition algorithms for a diver identification task (see Fig. 3.8). It shows that our approach is effective in capturing discriminative features from diver faces even though they are heavily obscured with mask and breathing apparatus. Furthermore, it only requires 194-dimensional feature vectors for each individual information, which uses minimal disk space to store them. Our model only requires 19M parameters, which is much less than the SOTA facial algorithms, such as ArcFace(40M) and FaceNet(140M). This helps reduce the computational load on embedded robotic computing hardware and will lead to faster inference time. From our testing on an Nvidia Jetson TX2 embedded GPU, it took $\approx 433 \text{ ms}$ to load the model and $\approx 133 \text{ ms}$ to make the inference. The results lead us to believe that the proposed framework can run in real-time. However, the model loading time needs to be improved by reducing its size in case multiple deep learning models are being simultaneously used on board the AUV.

3.4 Summary

This chapter presents a novel diver face identification system that leverages a series of data augmentation steps and a dimensionality reduction method to learn discriminative features from scuba diver faces. Although the proposed approach outperforms the SOTA face recognition algorithms for the task of diver identification (using facial cues), it still suffers from low accuracy. This is mostly due to the inferior quality of diver’s facial features as they are heavily obscured by mask and breathing apparatus.

In the next chapter, we aim to address this particular shortcoming by proposing a diver identification method which exploits features that are distance and photometric condition invariant.
Chapter 4

Diver Identification Using Anthropometric Data Ratios For Underwater Multi Human-Robot Collaboration

In the previous chapter, we introduced a diver identification method that uses facial cues of the divers to identify them. However, as we discussed, the method is significantly affected by the inferior quality of the facial features since diver faces are heavily obscured by masks and breathing apparatus. In addition, the faces of divers must be acquired in ground settings before deployment, which will bar the AUV from identifying a diver whose data is not known a priori.

To this end, in this work

1

, we propose a diver identification framework that utilizes distance and photometric invariant anthropometric features of the divers, and is functional in data-scarce condition. Fig. 4.1 depicts an example scenario where an AUV is using anthropometric features of divers to identify them. To address the diver identification problem, researchers have proposed various solutions, including fiducial

1 In this project, I have been the joint-lead student investigator with another Ph.D. student (during the collaboration) named Jungseok Hong.
Figure 4.1: An example scenario of a robot identifying divers with our proposed framework by using the anthropometric features extracted from the divers.

markers [41], unique hand signs [40], and acoustic communication devices [105]. However, with these methods, the diver must carry additional items or memorize unique hand signals, which can add to their already-elevated cognitive and physical burdens. Losing or forgetting such means of identification will compromise the entire mission. In contrast, our framework only requires the divers to take a standing-upright posture in front of the AUV which will use its vision sensor to automatically collect the necessary anthropometric features.

Our method can be deployed with either pre-trained models (offline framework) or models that can be trained during deployment (online framework). This allows the AUV to identify divers even if their data is not present in our dataset. This enables efficient human-robot collaborative missions with robust diver identification by addressing the domain-specific challenges underwater and constraints introduced above. Anthropometric Data (AD) refers to the study of human body measurements, such as the width of shoulders, lengths of the lower and upper arms, and measurement of other limbs or limb sections. AD can be captured even if divers’ bodies and faces are heavily obscured. However, the AD can change during the identification process, since the distance between the robot and the diver can vary. Therefore, we propose to use Anthropometric
Data Ratios (ADRs) instead, to represent divers uniquely, exploiting the property that ADRs will be invariant to the robot-diver distance. Through experiments, we demonstrate that these ADR features can be projected into a 16-d hyperspace where they show better separation. We achieve this by using our proposed embedding network that maximizes inter-class feature distance and minimizes intra-class feature distance, which significantly improves the classification performance. With that, we present a complete diver identification framework for underwater robots to identify divers using only ADR features.

In our framework, a robot starts by searching for and then approaching divers for identification. For the offline framework, the robot uses pre-trained models to identify the divers using ADR features generated by pose estimation and feature extraction. For the online framework, the robot collects data first to train models on-the-fly and identifies the divers upon completing the training. Our framework enables the robot to: (1) identify the divers with their known identities using the offline framework when prior data is available, and (2) distinguish the divers with the online framework even if their data is not available to the robot prior to deployment. We evaluate both frameworks with our proposed identification module in a closed-water environment onboard a physical AUV and demonstrate promising results. We make the following contributions in this work:

1. We propose distance and photometric invariant ADR features to create unique representations of scuba divers.

2. We introduce an embedding network that can project the ADR features to a 16-d hyperspace where they form highly separable inter-class clusters, resulting in superior classification performance.

3. We use anthropometric data statistics to filter out erroneous diver pose estimations.

4. We propose a diver identification algorithm for underwater robots to use the collected features to train a number of models both offline and online.

5. We measure the efficacy of the system on a physical AUV in a closed-water environment.
4.1 Related Work

One of the preliminary works on person identification was described in [106] where the proposed method used AD to identify different individuals. As the technology evolved, AD-based identification methods were replaced by systems that use biometric information, such as fingerprints, voice, and iris scans [107]. Spectral information extracted from electroencephalogram signals can also be used to identify a person [108]. Furthermore, person identification using facial cues has seen incredible success [109, 70], especially with the availability of large-scale human face datasets [110, 103, 111], superior computing power, and advances in deep convolutional neural networks [112]. Alternatively, gait detection can also be used to uniquely identify a person, as it has been found that walking patterns have a high correlation to a subject’s identity [113].

The above-mentioned techniques were developed for terrestrial applications. Person identification techniques in an underwater environment (i.e., diver identification), on the other hand, are significantly less studied. In a recent work [68], the authors present a diver identification framework by leveraging both spatial and frequency domain feature descriptors to uniquely represent different divers. In a similar line of work [69], the authors use appearance and location metrics to track divers for re-identification. However, in both of these papers, the divers need to be in motion to extract feature representations. In contrast, the work in [20] proposes to use facial cues to identify divers for Human-Robot Interaction (HRI), but achieves sub-par results because of the difficulty in extracting good facial features from heavily obscured diver faces.

Recently, Munsell et al. [114] show that AD can be used to identify different individuals. They demonstrate that these data are less sensitive to photometric differences and more robust to obstructions, e.g., glasses, and hats. Similarly, the work in [115] shows that anthropometric and gait features can be used to uniquely identify different people. In this work, we also choose to use AD to represent and identify different scuba divers. To compute them, we rely on 2D pose estimation methods which can find the location of different human body joints. There are a number of 2D pose estimation approaches (e.g., OpenPose [116], trt_pose [117], MediaPipe [118], DeepLabCut [119] to name a few) which achieve high accuracy in terrestrial applications. For our purpose, these methods generated a high number of incorrect pose estimations for the divers we
Figure 4.2: Given an RGB image input, our proposed algorithm first extracts features from the diver’s pose. It then predicts the diver’s ID using these highly distinguishable features. The numbers in the second sub-figure represent 17 body keypoints. Additionally, the AD\{1, \ldots, 10\} in the third sub-figure are the extracted features.

used in our study, possibly due to water turbidity and challenging lighting conditions. A recent work [120] uses high-resolution representation learning, by employing several high-to-low resolution convolutional branches in parallel, to achieve semantically rich and spatially precise pose estimations. We exploit this method in our work to estimate diver poses.

4.2 Methodology

4.2.1 Diver Identification Algorithm

It consists of four major components: (1) Pose Estimation and Filtering, (2) Feature Extraction, (3) Metric Learning, and (4) Classifier Training. We perform pose estimation on the image frames containing a diver and filter them using anthropometric data statistics. Then, we compute features from the filtered pose estimations to train different models to identify divers. Fig. 4.2 visualizes this process.

Pose Estimation and Filtering

To maximize the probability of getting stable pose estimation, our system requires that the divers: (1) take a standing-upright and frontal-facing position; and (2) ensure they do not occlude each other from the robot’s perspective. Divers usually pause to discuss the mission plan before transitioning to horizontal swimming postures. Therefore, it is reasonable to assume that an AUV observes scuba divers in an upright position for a
short duration for identification before the mission starts. We use HRNet \cite{120}, a robust 2D pose estimation method, to predict the joint locations of divers. While such posture requirements can help to produce good predictions, they cannot completely prevent the pose estimation method from making false predictions. Especially, rapid movements of scuba divers’ arms often make it extremely difficult for the pose estimation algorithm to accurately predict the poses. Fig. 4.3 demonstrates two such instances where the pose estimation algorithm fails to compute the locations of elbows, wrists, or shoulder joints correctly. Originally HRNet estimates 17 body keypoints, from which we select the following 10 keypoints: *left shoulder, right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee*. We include lower body joints, such as hips and knees, based on findings from our prior study \cite{121}, which demonstrated that the upper body lacks distinctive features for unique diver identification. It is only enough to provide an approximate distance to the divers. Additionally, upper body features are likely to be convoluted and obstructed by dive gear (*e.g.*, BCD, mask, and regulator). While including the remaining seven keypoints (*i.e.*, *nose, left eye, right eye, left ear, right ear, left ankle, right ankle*) could make our feature set richer, we observed that their predictions tend to be inconsistent across frames compared to the selected 10 keypoints. Additionally, the unselected seven keypoints require the underwater robot to constantly capture the whole body of the diver, which may not be realistic during underwater missions due to environmental factors (*e.g.*, waves).

To create robust ADR features from the predicted pose, erroneous data must be removed from the dataset. We achieve this by formulating a number of filtering conditions motivated by anthropometric data statistics \cite{122}. Specifically, we use the following filtering conditions.

- hips are located lower than shoulders.
- knees are located lower than hips.
- hip-width, shoulder-width, and distance between the knees need to be larger than 10 px.
- ratio of both hip-to-knee distances needs to be within (0.8, 1.2).
- ratio of both elbow-to-wrist distances needs to be within (0.6, 1.3).
Figure 4.3: Examples of pose estimation failures caused by rapid movements of divers’ arms underwater. As can be seen, the predictions for wrists, elbows, and shoulder joints locations are inaccurate.

- ratio of upper arm to lower arm needs to be within (1, 1.2).
- ratio of shoulder-to-hip distance to shoulder-width needs to be within (1.3, 1.6).
- shoulder-to-hip distance cannot be twice as large as hip-to-knee distance.
- shoulder-to-hip distance must be larger than the sizes of lower and upper arms.

We only keep the pose estimations that meet all the filtering conditions.

**Feature Extraction**

Once we obtain the filtered pose estimations, we calculate 10 AD values using the 10 body keypoints, as shown in the third sub-figure of Fig. 4.2. While the keypoint predictions are relatively stable, their locations in each frame constantly change as the robot moves. Therefore, keypoints and also the AD values are distance-variant. To address this issue and make the features distance-invariant, we propose to use ADRs as features to represent the divers. We compute the ADRs by taking the ratios of two different AD values, generating a total of 45 ratios from unique pairs, excluding the pairs formed by identical ADs. We observed that the ADRs stay relatively consistent for each diver even if the distance between the robot and the diver changes.
Table 4.1: Our proposed embedding network, which takes 45-d ADR features and project them into 16-d features.

<table>
<thead>
<tr>
<th>Input:</th>
<th>ADRs ($F \times 45$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 1</td>
<td>Linear (45, 1024); Leaky ReLU; BN</td>
</tr>
<tr>
<td>Layer 2</td>
<td>Linear (1024, 512); Leaky ReLU; BN</td>
</tr>
<tr>
<td>Layer 3</td>
<td>Linear (512, 256); Leaky ReLU; BN</td>
</tr>
<tr>
<td>Layer 4</td>
<td>Linear (256, 16)</td>
</tr>
</tbody>
</table>

**Metric Learning**

Metric learning [123] is a method to learn a distance measure for maximally separating inter-class distances in the feature space, using an appropriate distance metric. Examples of classical approaches are the K-nearest neighbor (KNN) [124] and support vector machine (SVM) [125]. We apply metric learning when we train our proposed embedding network (as shown in Table 4.1). Since the input feature vectors have only 45 dimensions, we use the embedding network with four hidden layers, where four was determined to be the minimum number of layers that yielded acceptable inter-class feature separation. We select the triplet loss [71] $L$ as our loss function due to its simplicity and effectiveness. Our loss function is formulated as follows:

$$L(A, P, N) = \max(0, D(A, P) - D(A, N) + m)$$

where $A$ is an anchor point (reference data), $P$ is a positive point, $N$ is a negative point, and $m$ is a predefined margin. By training our model with this loss function, we enforce intra-class data points (i.e., $A$ and $P$) to stay close while increasing the distance between inter-class data points (i.e., $A$ and $N$). In [71], Euclidean distance was used for the distance function $D$. While this is one of the common distance metrics, it cannot capture nonlinearity in the data. Thus, we use cosine similarity as a distance metric in our work.

Additionally, we created the following datasets for training the embedding network: (1) All-class dataset (four divers and four swimmers) and (2) Diver dataset (four divers only). The datasets are described in more detail in Sec. 4.3. Our embedding network
Figure 4.4: Results of the metric learning technique: clear separation among classes are seen. The 16-d features are visualized in 2D plots using t-SNE technique. The top and bottom rows show the clustering results on the training and test data, respectively.

takes 45-d features and outputs 16-d feature vectors. Interestingly, upon plotting these 16-d features from both training and test sets, they are observed to form well-defined clusters (see Fig. 4.4). We use the t-SNE [126] technique to plot the 16-d features in a 2D plot.

**Classifier Training**

We use 10 models to evaluate the performance of the diver identification algorithm (as shown in Table 4.2). The model names starting with All are trained with the All-class dataset, and those starting with Diver are trained with the Diver dataset. For All_KNN, Diver_KNN, All_SVM, and Diver_SVM models, we use only either KNN or SVM to train for classification without using an embedding network. For All_NN_KNN, Diver_NN_KNN, All_NN_SVM, Diver_NN_SVM, All_NN, and Diver_NN models, we first pass the raw 45-d ADR features through our embedding network to generate highly distinguishable 16-d features. We then train classifiers with the 16-d features. In the case of the All_NN and Diver_NN models, we use a two-layer neural network, attached with a final softmax layer to perform the multi-class classification using the 16-d features.
Table 4.2: Network structure and online training capabilities of different models used for our Diver Identification Framework.

<table>
<thead>
<tr>
<th>Model Name*</th>
<th>Embedding</th>
<th>Classification</th>
<th>Offline</th>
<th>Online</th>
</tr>
</thead>
<tbody>
<tr>
<td>All_KNN</td>
<td>-</td>
<td>KNN</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>Diver_KNN</td>
<td>-</td>
<td>KNN</td>
<td>✓  ✓</td>
<td>✓</td>
</tr>
<tr>
<td>All_SVM</td>
<td>-</td>
<td>SVM</td>
<td>✓  ✓</td>
<td>-</td>
</tr>
<tr>
<td>Diver_SVM</td>
<td>-</td>
<td>SVM</td>
<td>✓  ✓</td>
<td>✓</td>
</tr>
<tr>
<td>All_NN_KNN</td>
<td>NN</td>
<td>KNN</td>
<td>✓  ✓</td>
<td>✓</td>
</tr>
<tr>
<td>Diver_NN_KNN</td>
<td>NN</td>
<td>KNN</td>
<td>✓  ✓</td>
<td>✓</td>
</tr>
<tr>
<td>All_NN_SVM</td>
<td>NN</td>
<td>SVM</td>
<td>✓  ✓</td>
<td>✓</td>
</tr>
<tr>
<td>Diver_NN_SVM</td>
<td>NN</td>
<td>SVM</td>
<td>✓  ✓</td>
<td>✓</td>
</tr>
<tr>
<td>All_NN</td>
<td>NN</td>
<td>NN</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>Diver_NN</td>
<td>NN</td>
<td>NN</td>
<td>✓</td>
<td>-</td>
</tr>
</tbody>
</table>

*All_KNN and Diver_KNN refer to the same model for online training. Same goes for (All_SVM, Diver_SVM) and (All_NN, Diver_NN).

4.2.2 Diver Identification Framework

With our proposed diver identification algorithm, we develop a diver identification framework (see Fig. 4.5) using ROS [38]. It enables a robot to find divers and identify them, and it is designed to be compatible with both offline (models are trained before deployment) and online (models are trained during deployment) frameworks. The goal is to find a ‘target’ diver where the target diver is either pre-assigned (offline framework) or assigned during mission (online framework).

We implement these frameworks using a state machine that utilizes a Diver-Relative Position (DRP) estimator and a robot controller. These components are designed based on the Autonomous Diver-Relative Operator Configuration (ADROC) system introduced in [121]. Our state machine has the following eight states: INIT, SEARCH,
Figure 4.5: Our proposed Diver Identification Framework. The states in yellow boxes are common between the offline and online frameworks. Whereas, the state in the green box is used only in the online framework to perform model training during mission. The gray colored box shows the DRP estimator which runs continuously in the background.

**APPROACH, DATA COLLECTION, ANGLE YAW, MODEL TRAINING, IDENTIFICATION, and CONCLUSION.** The DRP estimator approximates the distance from the robot to the diver using bounding box detection (at a further distance) or pose estimation (at a closer distance) of the diver. When the distance information is available, i.e., when the robot detects a diver, its motion is governed through a Proportional-Integral-Derivative (PID) controller [127].

**Offline Framework**

The state machine starts with the INIT state and immediately makes a transition to the SEARCH state until the a distance information is available from the DRP estimator. Once available, the robot enters the APPROACH state. Now, the robot attempts to find and maintain a preset distance between itself and the diver. After reaching the desired distance, the robot enters the DATA COLLECTION state, where it computes the 45-d ADR features from $F$ (a predefined number) filtered pose estimations. With the $(F \times 45)$ ADR features, the robot goes into the IDENTIFICATION state since the models are already trained in this framework. If the target diver is found, the robot
enters the CONCLUSION state. Otherwise, it enters the SEARCH state again after staying briefly in the ANGLE YAW state. We included the ANGLE YAW state to make the robot intelligently yaw away from the current diver to prevent performing identification on the same diver again.

**Online Framework**

It operates the same way as the offline framework up to the DATA COLLECTION state. Since the classification models are not pre-trained in this framework, the state machine enters the MODEL TRAINING state. Here, the models (e.g., SVM and KNN) are trained using $(F \times 45)$ ADR features collected from $X$ number of divers. Since we only train SVM and KNN classifiers in this state, the training time is acceptable on typical robotic hardware (e.g., a mobile GPU or CPU). During our robot trials, for instance, the online training time for SVM and KNN models was just approximately 117 ms. Once the training is done, a target diver will randomly be assigned. However, our framework has the flexibility to update the target diver as needed, e.g., through specific hand gestures, or human-to-robot interaction. Then, the robot goes to the INIT state, and the state machine subsequently follows the same logic as the offline framework since the models are now trained.

### 4.3 Experiments

This section describes experimental setups for evaluating our proposed algorithm including dataset creation, model building and training, and robot trials with divers. Human data collection and trials were conducted with the approval of the University of Minnesota’s Institutional Review Board (study reference no. 00013497).

#### 4.3.1 UDI Dataset

To facilitate the learning of our proposed algorithm, we need to create a dataset that contains AD of divers. Since it is extremely challenging to find and recruit many certified divers to collect such data in a closed-water environment (e.g., pool), we collect the data from both swimmers and divers. In total, we have data from four swimmers and
Table 4.3: Comparison of average accuracy for the metric learning embedding network and the 10 models on our UDI dataset. The values are in percentages.

<table>
<thead>
<tr>
<th>Dataset Type</th>
<th>Embedding Network</th>
<th>Models</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All-Class</td>
<td>Diver</td>
</tr>
<tr>
<td>Training Accuracy</td>
<td>97.78</td>
<td>97.89</td>
</tr>
<tr>
<td>Testing Accuracy</td>
<td>96.39</td>
<td>96.65</td>
</tr>
</tbody>
</table>

four divers. We ask the participants to take frontal-facing and standing-upright postures towards the camera, and we capture their full body images using GoPro cameras at varying distances. We collect a total of 16,557 images across the eight participants. These images are extracted from diver videos at three frames per second to avoid repetitive information. We then perform pose estimation on these images, filter out erroneous poses, and compute the ADR features. This gives us a total of 13,994 ADR features and their corresponding labels, which we use to formulate the Underwater Diver Identification (UDI) dataset. We employ a 80/20 data split for training and testing.

### 4.3.2 Setup

We use two versions of the UDI dataset to train our models: (1) All-class dataset (whole UDI dataset) and (2) Diver dataset (a subset of the UDI dataset containing only divers). The first four models in Table 4.2 do not include the embedding network in their structures and hence are trained directly with 45-d raw ADR features. In contrast, the last six models include pre-trained embedding network and therefore are trained with 16-d features. We use scikit-learn libraries [128] to implement the KNN and SVM models, and, PyTorch libraries [65] to implement the neural networks. For the KNN model, we set the starting value of the neighbor size as \( \sqrt{30} \approx 5 \), following discussions in [129] [130]. Here, 30 represents the average number of data samples observed during an inference. Our metric learning embedding network is trained for 1,000 epochs with a batch size of 512, learning rate of \( 5 \times 10^{-4} \), and margin, \( m = 0.3 \). We use Stochastic Gradient Descent (SGD) as the optimizer [131]. For the All_NN and Diver_NN models, the classification network is a two-layer neural network.
For offline framework, both embedding and classification networks are trained before deployment. In contrast, the classification network is trained during mission in online framework although the embedding network is still pre-trained. Table 4.3 presents the average training and testing accuracy for both the embedding network and 10 models presented in Table 4.2. Additionally, we train the models with features computed from either 50 or 100 frames to evaluate the correlation between the number of training data and the performance of each model. To keep the online training time short, we do not consider training the Diver_NN model.

### 4.3.3 Experimental Scenarios

We use three divers in closed-water environments to perform our experiments using the offline and online frameworks. In the offline framework, our AUV collects ADR features from a diver and then feeds the features through pre-trained models and gets a prediction. If the prediction is accurate, then the algorithm concludes and enables the robot to initiate an interaction with the diver. Otherwise, the robot turns to the next diver. The robot repeats this for all the divers present in the scene. We assume that the robot knows a priori the total number of divers present in the scene.

In the online framework, the robot aims to identify a diver without using pre-trained classification models. To achieve this, first, the robot collects the ADR features from all the observable divers. It then trains six models online, to create highly separable clusters of divers. Once a target diver is assigned, the robot uses these newly trained models to identify the diver. Whenever there is a correct match, the algorithm concludes without checking the rest of the divers. If the robot is unable to find a correct match after checking all divers, it is considered as a failure case.

### 4.4 Results

We deploy the system onboard the LoCO AUV [30] in closed-water environments to conduct 16 robot trials. During each of these trials, the AUV starts by searching for a diver, approaching them for data collection, and then performing identification. The AUV performs these actions completely autonomously using our proposed diver identification framework. To our knowledge, the only other work that attempts to address
the problem of diver identification is presented in [20] where the authors propose to use facial cues to identify different divers. We consider this work to be the baseline method for evaluating the performance of our proposed system. During the trials, we compute true positive, TP, true negative, TN, false positive, FP, and false negative, FN. Table 4.4 summarizes the results of these trials. From the table, we notice that our proposed method accurately identifies divers in 18 out of 23 instances, yielding an average prediction accuracy of 78.26%. Although ADR features are highly discriminative (see Fig. 4.4), we observed that the sensitivity of the PID controller led to an overshooting effect in the AUV’s response to a diver’s position. This unintended overshoot compromised the accuracy of pose estimations, resulting in the false identifications. In comparison, the baseline method achieves only a 33.33% average prediction accuracy and fails to detect any diver faces during robot trial no. 16. This demonstrates that our proposed method not only reliably identifies divers but also outperforms the only other state-of-the-art method in the diver identification task.

4.4.1 Performance of the Offline Framework

We test the performance of all 10 models pre-trained with the All-class and Diver dataset. Fig. 4.6 presents the quantitative results of the experiments performed for the offline framework. The bottom plot in Fig. 4.6 shows the average accuracy of each model across all numbers of test frames. From the figure, the superior performance of All_NN_KNN and All_NN_SVM is evident. Finally, we calculate the average accuracy across all the models (the top plot in Fig. 4.6) and notice that the accuracy remains almost the same. This suggests that All_NN_KNN and All_NN_SVM can even make faster predictions using a lower number of frames while maintaining their high accuracy.

4.4.2 Performance of the Online Framework

We first use 50 frames per diver to train the models and evaluate the diver identification performance. Fig. 4.7 shows the quantitative results of the experiments performed for the online framework. From Fig. 4.7a (bottom), we notice that All_NN_SVM performs the best and All_NN_KNN shows comparable performance. That means the metric
Table 4.4: Results of 16 robot trials conducted in closed-water environments for diver identification task where prediction accuracy $= \frac{(TP+TN)}{(TP+TN+FP+FN)}$.

<table>
<thead>
<tr>
<th>Trial No.</th>
<th>Our Method</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TP  TN  FP  FN</td>
<td>TP  TN  FP  FN</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>no face detection</td>
</tr>
</tbody>
</table>

Sum 7 11 1 4 5 3 7 9

Pred. Acc. 78.26% 33.33%
learning technique is indeed benefiting the identification task. Even if we change the number of frames per diver to train the models to 100, we see similar trends. From Fig. 4.7b (bottom), we see that both the All_NN_SVM and All_NN_KNN achieve the highest accuracy. Finally, from Fig. 4.7a (top) and 4.7b (top), we notice that the number of frames used for making inferences does not have any significant effect on the system’s performance for the online framework, as was the case for the offline framework.

Moreover, we have conducted experiments to evaluate the robustness of the proposed ADR features outside closed-water environment. As shown in Fig. 4.8, our method successfully performs across various environmental settings, including terrestrial and open water scenarios. Note the different photometric conditions and distances of the persons positioned in front of the AUV in Fig. 4.8b-4.8d.

As these results suggest, our diver identification framework can successfully identify divers within both the offline and online frameworks. In both cases, the All_NN_KNN and All_NN_SVM models outperform the other models. Especially, the All_NN_SVM
Figure 4.7: Average accuracy of the online framework. (Top row): average accuracy for a specific number of test frames across all models, (Bottom row): average accuracy of each model across all frames.

(a) Trained online with 50 data points.

(b) Trained online with 100 data points.
model shows consistently accurate identification performance regardless of the type of the framework and the number of frames for training and inference. Additionally, we see that the models with our embedding network achieve higher identification accuracy than the models without the network. More importantly, our results demonstrate that the ADR features are distance and photometric-invariant considering the fact that (1) our data has been captured at varying distances, and (2) images from the camera (0.9-megapixel, 720p resolution) that we used to build our training dataset with and the robot camera (2-megapixel, 1080p), are significantly different.

As we have described in Sec. 4.2, our approach relies on visual features. This limits the approach to only work when AUV can see divers. However, when water turbidity is not severe, supplementary algorithms (e.g., FUnIE-GAN [39]) can be used to improve visibility.

4.5 Summary

In this chapter, we present the design and implementation of a diver identification algorithm for AUVs using ADRs as features, which are invariant to distance and photometric conditions. With experiments performed during closed-water trials, we show that our proposed algorithm enables an AUV to extract the ADR features from divers.
using a pose estimation technique and use them to identify a target diver. We incorpo-
rate our diver identification algorithm within both offline and online frameworks. This
enables the AUV to identify divers even if we do not have their data in our dataset
prior to deployments. This technique significantly overcomes the shortcomings of the
diver identification method that uses facial cues. The key advantages of the presented
method are: it only requires a monocular camera to identify a diver, and it can be
deployed on-board a physical AUV. We believe our proposed method will enable secure
human-robot collaborative missions by successfully identifying divers.

In the next chapter, we present a diver attention estimation system. Once an AUV
knows who to work with, it will try to interact with fellow divers before starting or
during a multi-human-robot collaborative mission. So, it is important for the AUV to
determine if a diver is paying enough attention to show the intent to begin or continue
an interaction. We attempt to address this problem in the following chapter.
Chapter 5

A Diver Attention Estimation Framework for Effective Underwater Human-Robot Interaction

In recent years, Autonomous Underwater Vehicles (AUVs) have been used in standalone missions, such as underwater oil spill survey system [10] and oceanographic survey system [12]. They are also increasingly utilized in collaborative settings involving both humans and other AUVs (e.g., [76, 19]). The ever-increasing on-board computing power, availability of low-cost and affordable AUVs, and improved Human-Robot Interaction (HRI) capabilities are some of the contributing factors behind the surge in AUV usage. Currently, humans and AUVs can interact with each other using a number of different methods, such as high-speed tethered communication [18], fiducial markers [41], gesture-based framework [40], small displays and lighting schemes [17], and motion-based methods [36]. But these interaction systems usually require both the diver and the AUV to be oriented properly and to be close to each other. In addition, it is mostly the diver who is required to swim towards the AUV, orient themselves correctly with respect to the AUV, and begin the interaction. However, this puts additional physical and cognitive burdens on the divers. In contrast, AUVs do not currently have this capability
Figure 5.1: Demonstration of the proposed framework running on-board the Aqua AUV [6] during our open-water experiments in the Caribbean Sea, off the coast of Barbados. The red dashed line represents the robot’s future trajectory to position itself conveniently for interaction.

as it is extremely difficult for a machine to intelligently perceive human orientation and attention underwater, localize itself and the target, navigate towards the target, and reorient itself all at the same time. This is mainly due to the domain-specific challenges underwater, e.g., degraded visuals, reduced sensing capability arising from signal attenuation, less reliable localization and motion planning, and lack of high-bandwidth wireless communication.

To this end, we present a novel diver attention estimation framework that enables AUVs to determine the attentiveness of a diver. In the context of the work, we use the phrases “diver attention” and “attentiveness of a diver” interchangeably, which indicates whether or not a diver is paying attention to the robot (i.e., their head orientation is straight towards the robot) with an intention to interact and/or collaborate. The core element of the proposed framework is a deep convolutional neural network called DATT-Net (Diver Attention Network). It is based on a pyramid structure that can detect diver faces and the spatial locations of 10 facial keypoints. We introduce a novel geometric loss in the training process that enforces the network to maintain the geometric relations among the keypoints. This enables the network to robustly detect the facial keypoints
even when a diver is looking at an angle. Furthermore, we design an algorithm that uses
the detected keypoints to estimate the attentiveness of a scuba diver. Additionally, we
develop a robot controller for the Aqua AUV [6] that enables it to navigate and reorient
for interaction based on the diver’s attention (see Fig. 5.1). We make the following
contributions in this work:

1. We propose an end-to-end deep convolutional neural network called DATT-Net
to detect diver faces and their associated facial keypoints even when the diver
is facing the robot at a 90-degree angle. The training pipeline includes a multi-
loss objective function which gives emphasis on maintaining the actual geometric
relations among the facial keypoints.

2. We present the DATT dataset containing 3,314 annotated diver images, collected
during multiple closed-water robot trials, that facilitate the training of DATT-Net.
The images are carefully annotated to include diver face bounding boxes and 10
facial keypoints of divers.

3. We perform spread analysis on the facial keypoints to determine the attentiveness
of a diver and define a pseudo_angle variable to quantify the direction of the
diver’s attention.

4. We design a robot controller that enables the AUV to position itself conveniently
based on the diver’s attention.

5. We deploy the proposed framework on a physical AUV in both closed- and open-
water environments to measure the efficacy of the system in real-world use cases.

5.1 Related Work

AUVs are successfully using their visual perception capabilities in both littoral and off-
shore regions owing to the success of underwater vision improvements [20] and novel
deep learning innovations [132]. This success contributes to the development of the
work in [121], which proposes to use monocular cameras to approach a human diver by
leveraging the pose of the body and biological priors to facilitate human-robot interac-
tion. This method does not require the use of expensive sensors, such as stereo cameras,
sonar, acoustic sensors (e.g., Doppler Velocity Logs), and localization devices, and yet achieves promising results. However, the proposed solution always assumes that a diver is ready to interact. There are times when a diver may be engaged in a different task or interaction with a fellow dive buddy but is still willing to interact with a robot only if the robot positions itself conveniently. In such scenarios, it is imperative for the AUVs to understand the intent of the diver, more specifically their attentiveness.

To determine the attentiveness of the diver, the AUV needs to know the gaze direction or the head pose of the diver. In terrestrial applications, researchers solve 3D (head) pose estimation problems using various learning algorithms (e.g., [133, 134, 135]), which use datasets that contain 3D locations of the pose keypoints with respect to a global frame. However, there are no datasets that contain 3D head pose data of scuba divers, partly arising from the difficulty and logistical challenges in collecting such datasets. In contrast, some researchers model the 3D pose estimation problem as a 2D problem and have used keypoints regression to address the 3D pose estimation challenge [136, 137]. It is almost impossible to estimate the 3D pose from a 2D image if the corresponding 2D keypoints are occluded in the image space. Consequently, the unseen keypoints are not even annotated in the datasets and are not used in establishing the 2D-3D correspondence. The works in [138] and [139] tackle the diver pose estimation problem to determine diver heading (i.e., orientation), however, they either require stereo image pair or use a computationally expensive model that is not suitable for real-world deployments.

We propose to tackle these issues by creating the DATT dataset where we carefully annotate images captured by monocular cameras with diver face bounding boxes and 10 facial keypoints. We annotate the facial keypoints even when the frontal face is occluded, to promote learning of keypoints’ geometric structure rather than just their individual locations.

## 5.2 DATT Dataset

To facilitate the training and testing of the proposed diver attention estimation system, we prepare a diver attention dataset called DATT, which contains a wide variety of annotated underwater diver images. The images have a resolution of 1920 × 1080
pixels and are collected from multiple closed-water pool trials using GoPro action cameras [140]. There are a total of 3,314 annotated diver images (train set: 2,652 and test set: 662). To increase the diversity of the dataset, the divers were instructed to appear differently underwater. Some divers wore scuba masks while others simply put on pairs of goggles. Some used their snorkels while a few others did not. The head orientation of the divers were also varied by instructing them to look directly at the camera or to look elsewhere at different angles (both in the left and right directions). As a result, there are even instances where the faces of the divers are completely obscured from the camera’s perspective, i.e., the diver is looking to their left or right at a 90 degree angle with respect to the camera.

Since we want to detect the gaze direction of the divers directly from the 2D images, we carefully annotate the images to reflect as many 3D perspective properties as possible. We start by annotating the diver face bounding boxes. To determine the head orientation of the diver, a full set of facial keypoints and their underlying geometrical relationships need to be used. Traditionally, researchers select eyes and two corners of the lips, among others, as three of the vital facial keypoints for terrestrial applications. However, for a diver underwater, their eyes are usually obscured by the scuba mask or goggles and the shape of their lips are mostly distorted by the snorkels or regulators. So, instead of selecting facial keypoints in the traditional manner, we leverage the following facts. Since the breathing apparatus on the face (e.g., masks and regulators) are always visible underwater and are rigid types of objects, we select most of our facial keypoints on those and focus mainly on finding the geometry of the facial keypoints to mimic the shape of those in 3D space. Specifically, we select 10 facial keypoints which are easier to identify underwater compared to the traditional facial keypoints, where eight keypoints are on the mask, one is on the snorkel or regulator, and the remaining keypoint is on the diver’s chin. We always annotate all 10 facial keypoints even when some of them are occluded from the camera’s perspective so that a downstream algorithm can regress the keypoints based on the actual geometric relation among the keypoints. Fig. 5.2 shows a sample image and a few annotations from the DATT dataset. We make it available at https://irvlab.cs.umn.edu/datt.
Figure 5.2: (a) A sample image from DATT dataset and the corresponding label containing two bounding box coordinates and 10 facial keypoints. (b) A few additional annotated samples where the divers are wearing different types of scuba gear and are looking at different directions.
5.3 DATT-Net

The primary goal of the proposed diver attention estimation framework is to calculate the head orientation (i.e., gaze direction) of a scuba diver to determine if they are attentive to a companion robot. To achieve this, we have to: 1. localize diver faces, 2. determine the facial keypoints of the diver, and 3. estimate the gaze direction (i.e., attentiveness) from the geometry of the keypoints. In this work, we propose to solve the first two of these tasks with a unified architecture, called DATT-Net, in a supervised manner. Inspired by the robust single-stage face detector named RetinaFace [96], we design DATT-Net as follows where we employ the loss functions of RetinaFace as well as add a specialized geometric loss function to exploit the geometric relation among the 10 facial keypoints of the divers to determine their head orientation. Fig. 5.3 shows the complete architecture of DATT-Net.

5.3.1 Feature Extractor

DATT-Net uses a ResNet-101 block [141], pretrained on ImageNet dataset [142], to extract the initial features from the input image of 640 × 640 resolution. Since we want to identify the diver faces and consequently the facial keypoints of the divers regardless of their physical distance from the robot, we employ a multi-scale learning approach as was used in [143]. Therefore, we extract the initial features at three different scales (i.e., from three different layers of ResNet-101). These features are further refined individually
using a backbone network (we use either ResNet-50 [141] or MobileNet-V2 [144] as the backbone). We refrain from using any pretrained weights for the backbone network to eliminate any learning biases from the ImageNet dataset. We then create a three-level ($P_1$, $P_2$, $P_3$) feature pyramid [145] with these extracted features using both top-down and lateral connections, as described in [96]. Instead of performing the final predictions directly on the feature pyramid, we use extra context learning modules to leverage information from the surrounding pixels. It is shown in [146] that using such post-context learning blocks increases the receptive fields of the pixel locations, which enables knowledge from the surroundings and hence gives better inference. The outputs from the final three context modules have feature dimensions of ($80 \times 80 \times 256$), ($40 \times 40 \times 256$), and ($20 \times 20 \times 256$), respectively.

### 5.3.2 Facial Anchors

To capture the facial keypoints of the divers at different scales, we perform the network supervision at patch levels on the feature pyramid. These patches, called *anchors*, were introduced in [147] for enabling multi-scale face detection. Since underwater human-robot interaction takes place at a reasonably close distance, we select 16 as the smallest anchor size to capture distant divers. As for the rest, we use anchors of sizes 32, 64, 128, and 256. Anchors 16 and 32 work on level $P_1$, 64 and 128 on level $P_2$, and finally the anchor 256 works on level $P_3$. All the anchors have square dimensions. We consider an anchor to have a diver face if it matches with the ground truth bounding box with an Intersection over Union (IoU) value of at least 60% and to have background if the IoU is less than 0.4. All other anchors are ignored.

### 5.3.3 Objective Function Formulation

One of the primary goals of this work is to accurately identify the facial keypoints even when they are occluded from the camera’s perspective. Therefore, we formulate the objective function as a multi-loss optimization problem (as was used in [148]) where we not only want to detect diver faces and their facial keypoints but also maintain the underlying geometric relation among the 10 keypoints. The individual loss components are described below:
1. $\mathcal{L}_{fc}(f, f_{gt})$: This is a cross-entropy loss which governs whether a particular image patch (anchor) is a face or background. Here, $f$ is the probability of an anchor being a face. Additionally, if the ground truth anchor also contains a face, then the value of $f_{gt}$ is 1, otherwise it is 0. Here, $gt$ refers to the ground truth values.

2. $\mathcal{L}_{fb}(\vec{b}, \vec{b}_{gt})$: This is the diver face bounding box regression loss where $\vec{b} = [x_{min}, y_{min}, x_{max}, y_{max}]$ is a vector containing the predicted face bounding box coordinates. Here, $\mathcal{L}_{fb}$ is a smooth-L$_1$ loss which was introduced in [148] and is defined as,

$$\text{smooth-L}_1(x) = \begin{cases} 0.5x^2 & \text{if } |x| \leq 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

(5.1)

where $x$ is fed as the difference between the predicted and the ground truth bounding box coordinates. Finally, the loss is averaged for all four coordinates.

3. $\mathcal{L}_{kp}(\vec{p}, \vec{p}_{gt})$: This is the diver face keypoints regression loss where $\vec{p} = [x_1, y_1, \cdots, x_{10}, y_{10}]$. The loss is also calculated as a smooth-L$_1$ loss (as shown in Eq. [5.1]). Essentially, $\mathcal{L}_{kp}$ supervises the algorithm to predict the 10 facial keypoints closer to the ground truth values. For both $\mathcal{L}_{fb}$ and $\mathcal{L}_{kp}$ losses, the four bounding box coordinates and the 10 facial keypoints values were normalized.

4. $\mathcal{L}_{gm}(\vec{p})$: Finally, we introduce an additional geometric loss term which enforces the algorithm to maintain the geometric relation among the 10 keypoints. This should allow accurate keypoints regression even when all 10 keypoints are occluded from the camera’s perspective. This is a vital criterion because we want our algorithm to infer accurate facial keypoints even when the diver is looking at a 90 degree angle with respect to the robot’s camera. Since scuba masks are rigid object, the points on a mask should never deviate from their relative positions (with respect to one another) in a 3D space regardless of the head orientation of the diver. We leverage this idea on image space and ensure that the outer six keypoints $(p_1, p_3, p_4, p_5, p_6, p_8)$ on the scuba masks are symmetric with respect to the middle two keypoints $(p_2, p_7)$. We define the loss term as,

$$\mathcal{L}_{gm}(-\vec{p}) = \sum_{j=2,7} l(p_{1j}, p_{3j}) + l(p_{4j}, p_{5j}) + l(p_{6j}, p_{8j})$$
where

\[ l(p_{ab}, p_{cb}) = |d_{p_{ab}} - d_{p_{cb}}| \]

and,

\[ d_{pmn} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2} \]

In total, we minimize the following multi-loss objective function:

\[ \mathcal{L} = \mathcal{L}_{fc} + \alpha f_{gt} \mathcal{L}_{fb} + \beta f_{gt} \mathcal{L}_{kp} + \gamma f_{gt} \mathcal{L}_{gm} \] (5.2)

where \( \alpha, \beta, \gamma \) are set to 0.2, 0.15, and 0.1, respectively. We use non-uniform loss weights to prevent overfitting [149]. A higher loss weight is assigned to \( L_{fb} \) to ensure accurate diver face bounding box detection, which is crucial for robust keypoints regression. The last three loss terms in Eq. 5.2 are considered only if an anchor contains a face, i.e., \( f_{gt} = 1 \). Fig. 5.3 shows the complete architecture of DATT-Net.

### 5.3.4 Implementation Details

We use TensorFlow [150] libraries to implement DATT-Net. All the input images are resized to have 640 × 640 resolution. To augment the DATT dataset, we utilize the following schemes (as suggested in [151, 84]): random cropping, image flipping, normalization, image distortion (varying the brightness, hue, saturation, contrast, and sharpness), etc. During training, we use a batch size of 8, learning rate of \( 10^{-2} \) with a scheduler that lowers the value down to \( 10^{-5} \), and Stochastic Gradient Descent (SGD) as the optimizer with a momentum of 0.9. We train the network using 2,652 training images on an Nvidia GeForce RTX 2080 GPU for 300 epochs where we notice convergence in the validation loss for both ResNet-50 and MobileNet-V2 backbones (see Fig. 5.4). For on-the-bench evaluations, we have used an Intel® Xeon® E5-2650 CPU to run DATT-Net. And, during real-world deployments in both closed- and open-water environments, we have used the on-board computing platform (Nvidia Jetson® TX2) of the Aqua AUV.

### 5.4 Diver Attention Estimation

We estimate the attentiveness of a scuba diver using DATT-Net’s prediction of the 10 facial keypoints. First, we measure the \( x \)-spread of the outer keypoints on the scuba
Figure 5.4: The training performance of DATT-Net in terms of the validation loss where the backbone is either (a) ResNet-50, or (b) MobileNet-V2. Here, iterations = (no. of training images/batch size) * epochs.

mask \((p_1, p_3, p_4, p_5, p_6, p_8)\) with respect to the nose \((p_7)\) of the diver. However, instead of directly comparing against the nose coordinate, we use the mean \((x_{\text{mean}})\) location of the inner keypoints \((p_2, p_7, p_9, p_{10})\) because these points should always lie on the same line in a 3D space. We use,

\[
x\text{-spread} = \frac{1}{6} \sum_{i=1, 3, 4, 5, 6, 8} \sqrt{(x_{p_i} - x_{\text{mean}})^2}
\]

to compute the spread. We normalize the \(x\)-spread value with respect to the width of the predicted diver face bounding box. Second, we leverage the positions of the top three keypoints \((p_1, p_2, p_3)\) on the diver’s mask with respect to center \((x_{\text{center}})\) of the face bounding box to define a pseudo-angle variable to quantify the angular deviation
of a diver’s head from the center; \textit{pseudo}-angle is defined as

\[ \text{pseudo-angle} = x_{\text{mask}} - x_{\text{center}} \]

where \( x_{\text{mask}} \) is the mean \( x \) location of the keypoints \( p_1, p_2, p_3 \).

Finally, we determine the attentiveness (\( att \)) of a diver as follows,

\[ att = \begin{cases} 
\text{True} & \text{if } x\text{-spread} > \lambda_1 \text{ and } |\text{pseudo-angle}| < \lambda_2 \\
\text{False} & \text{otherwise}
\end{cases} \]

where the values of \( \lambda_1 \) and \( \lambda_2 \) are empirically found as 0.27 and 10, respectively.

Also, we conclude that a diver is looking to their left if \( \text{pseudo-angle} > 0 \) and right if \( \text{pseudo-angle} < 0 \).

### 5.5 Robot Controller

We use the six-legged Aqua AUV platform [6] to design a controller to leverage the estimated \textit{pseudo}-angle value to navigate and reorient itself for interaction with a diver. We create a Robot Operating System (ROS) [38] package consisting of two nodes; one of these continuously predicts the facial keypoints to estimate the diver attention, and the other node keeps track of the attentiveness of the diver for the past 3 seconds. Based on the recent attentiveness (\textit{Attentive, Looking Right}, or \textit{Looking left}) score, we design the controller to make the Aqua AUV try to catch the attention of the diver with a unique movement. If unsuccessful, it autonomously navigates and reorients itself to interact with the diver (\textit{i.e.}, if the diver is looking right, the AUV must do a left maneuver, and vice-versa). We implement the maneuvers using a motion controller introduced in [62].

During the navigation-and-reorientation phase, the initial turn, the circular movement, and the final turn are governed by the \textit{pseudo_angle} value, which is computed in the previous section. The ROS package and the implementation code for the controller are available at [https://github.com/IRVLab/datt_controller](https://github.com/IRVLab/datt_controller).

### 5.6 Experimental Evaluations

To evaluate the performance of the diver attention estimation framework, we prepare an evaluation set containing 812 diver images. Of these, 662 images are from the DATT
Table 5.1: Diver Face detection and Keypoints regression results on the evaluation set. The values are in percentages.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AP (↑)</td>
<td>mAP (↑)</td>
</tr>
<tr>
<td>PyramidBox [82]</td>
<td>61.45</td>
<td>44.83</td>
</tr>
<tr>
<td>BlazeFace [152]</td>
<td>66.95</td>
<td>45.86</td>
</tr>
<tr>
<td>DSFD [97]</td>
<td>77.22</td>
<td>54.70</td>
</tr>
<tr>
<td>RetinaFace [96]</td>
<td>70.81</td>
<td>48.33</td>
</tr>
<tr>
<td><strong>DATT-Net (+ ResNet-50)</strong></td>
<td><strong>91.75</strong></td>
<td><strong>72.56</strong></td>
</tr>
<tr>
<td>Same w/o (L_{gm})</td>
<td>82.64</td>
<td>55.22</td>
</tr>
<tr>
<td>DATT-Net (+ MobileNet-V2)</td>
<td>87.93</td>
<td>55.33</td>
</tr>
<tr>
<td>Same w/o (L_{gm})</td>
<td>75.37</td>
<td>50.22</td>
</tr>
</tbody>
</table>

test set and 150 images are from multiple open-water environments. Furthermore, we deploy the proposed system on-board the Aqua AUV in real-world scenarios (robot trials carried out in a swimming pool and the Caribbean Sea) and evaluate the efficacy of the framework with scuba divers present in the field of view of the robot. To test the performance of the proposed robot controller, we check if the Aqua AUV successfully navigates and reorients itself with respect to the diver if they are found inattentive.

5.6.1 Results

**Diver Face Detection and Keypoints Regression**

Since a major component of the proposed framework is diver face detection, we start by comparing the diver face detection performance of DATT-Net against the State-Of-The-Art (SOTA) face detection algorithms. *Metrics:* we compute the Average Precision (AP) for IoU=0.5 and Mean Average Precision (mAP) for IoU=[0.5:0.05:0.95]. The results are presented in Table 5.1. From the table, it is evident that DATT-Net achieves significantly better performance than the SOTA methods, regardless of the backbone network. While we also considered LFFD [153], errors present in the available GitHub
code prevented us from training the model for quantitative comparisons. Even though RetinaFace [96] and BlazeFace [152] use facial landmarks to robustly detect faces, their accuracy suffer on diver faces. This might be due to divers wearing different types of scuba gear because these methods worked well for divers who did not wear masks, snorkels, or regulators (e.g., wearing only pairs of goggles). Furthermore, it is also evident from the table that the use of the additional supervision in the form of geometric loss contributes to better diver face detection because the AP value decreases when the geometric loss is excluded during training. DATT-Net achieves the best performance (AP=91.75%, mAP=72.56%) when ResNet-50 is used as the backbone network.

For the keypoints regression task, we are only able to evaluate the performance of DATT-Net models because we do not have ground truth values for BlazeFace and RetinaFace, whereas PyramidBox [82] and DSFD [97] do not output facial keypoints. Metric: we compute the Percentage of Correct Keypoints (PCK) [154] by measuring if the predicted keypoint and the ground truth are within a certain distance threshold. For our purpose, we set the threshold as 10% of the mask width (distance between the keypoints #4 and #5). From Table 5.1, we see that DATT-Net with ResNet-50 backbone achieves the best PCK value of 86.21%. As before, we notice similar phenomenon where the PCK value decreases when the geometric loss is excluded.

**Diver Attention Estimation**

First, we perform qualitative evaluations on the performance of the proposed diver attention estimation technique. We test our algorithm on the evaluation set and also in two open-water trials (Square Lake, MN, USA, and Caribbean Sea, off the coast of Barbados). Fig. 5.5 presents the qualitative results. From the figure we see that the proposed algorithm accurately determines the attentiveness of divers irrespective of the type of scuba gear they are wearing. It is also evident that the proposed solution works at different distances and in various lighting conditions. This is understandable because we have used multi-scale learning (using different sizes of anchors) on a feature pyramid and image distortion as a data augmentation scheme, respectively, during the training of DATT-Net (as described in Sec. 5.3.2 and Sec. 5.3.4). It is important to note that even though DATT-Net was trained using images collected from closed-water environments, e.g., swimming pools, it performs well on both lake and sea images captured by the
Figure 5.5: Qualitative performance of the proposed diver attention estimation framework when run on-board the Aqua AUV in both closed- and open-water environments. Note the robustness of our method as it works at different distances and in challenging lighting conditions.
robot’s camera. The framework also performs well even when the majority of the diver’s face is not visible (e.g., because they are facing the AUV at a 90-degree angle). This points to DATT-Net indeed benefiting from learning the geometry of the keypoints.

Second, we quantitatively evaluate the performance of the proposed framework. To our knowledge, there is no other research tackling the problem of diver attention estimation that we can compare against. Consequently, we decide to use terrestrial facial keypoints regression techniques as our baselines, such as a CNN-based keypoints regressor [155], BlazeFace, and the Perspective-n-Point (PnP) algorithm [156]. To determine the attentiveness of divers using these techniques, we do the following. First, we compute the facial keypoints from diver images in our evaluation set using the CNN-based keypoints regressor, BlazeFace, and DATT-Net. Then, we use spread analysis (as discussed in Sec. 5.4) to determine the attentiveness of divers. We had to modify which keypoints to consider during the calculation of the spread analysis for the first two techniques as the keypoints’ locations are defined differently for them. In contrast, we determine the diver attention for the PnP technique as follows. We add depth information to the facial keypoints to prepare a 3D model of the diver’s head wearing a mask and snorkel. The depth values are assigned relative to the mask, with points vertically above the mask having a positive depth, and those vertically below having a negative value, with the depth at mask being zero. Now, given a diver image, we run DATT-Net to find the 10 facial keypoints and use the 3D model of the diver’s head to estimate the head pose of the diver (i.e., translation and rotation vectors) using the PnP technique. From the translation and rotation vectors of the nose (keypoint #7), we compute its angular deviation from the center and determine the attentiveness of the diver. Table 5.2 presents the results. We see that our framework achieves a superior accuracy of 89.41% compared to the other baselines.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (↑)</td>
<td>38.30</td>
<td>42.73</td>
<td>45.07</td>
<td>89.41</td>
</tr>
</tbody>
</table>

Table 5.2: Diver attention estimation results on the evaluation set. The accuracy values are in percentages.
Robot Controller’s Performance

We deployed our robot controller on-board the Aqua AUV using ROS and tested the performance of the controller in both closed- and open-water environments. The experimental setup involved two divers and one AUV, with one diver working as a robot wrangler while the other served as the interaction target. When the AUV detected the diver to be inattentive, it initiated a unique movement pattern to draw their attention. If unsuccessful, the AUV autonomously devised an optimal trajectory to navigate and reposition itself for effective interaction with the diver. This planning process relied on the \textit{pseudo-angle} value derived from diver attention estimation. In contrast, if the diver is found to be attentive, the AUV acknowledged with another unique movement pattern and began the interaction process. The performance of the robot controller in real-world scenarios is provided in the accompanying video, showcasing its effectiveness in facilitating AUV-diver interaction.

5.6.2 Limitations of the Proposed Framework

Our method relies on the prediction of the diver face bounding box. If the predicted bounding box does not include any faces, then all the subsequent modules fail. Although the performance of DATT-Net in detecting diver faces is quite robust, we still see a few failure cases. As seen in Fig. 5.6, the algorithm mistakenly found a diver face in the water reflection and also mistook a lane marking as a face on the swimming pool floor. Also, the current robot controller is designed to only change its yaw value during the navigation-and-reorientation phase. That is, our controller makes an assumption that the interaction will happen at a fixed height in the water column.

5.7 Summary

In this chapter, we presented the design and implementation of a diver attention estimation framework to facilitate underwater diver-AUV interaction. With on-the-bench experiments, closed- and open-water trials, we showed that our proposed system is able to determine the gaze direction of the divers (\textit{i.e.}, their attentiveness) by understanding the geometric relation among 10 facial keypoints. We also proposed a robot controller
Figure 5.6: Two examples of diver attention estimation failures where the algorithm incorrectly detects a diver face in the reflection on the water surface (left image) and in the floor of the pool (right image).

that allows the AUV to leverage the attentiveness of a diver to navigate and reorient itself before initiating an interaction. The key advantages of our method are that it works on a physical AUV, can detect the attentiveness of the divers using only a monocular camera, works at various distances and lighting conditions, does not need to know the global position of the diver to work, and is invariant to different types of scuba gear.

However, there are situations when the AUV needs to assess the diver’s actions before attempting to interrupt them, even if it is well-positioned for an interaction. One such scenario arises when the diver is engaged in a time-critical task, such as rescuing a fellow dive buddy in distress, requiring swift and focused action. In such instances, it is essential not to disrupt the diver, allowing them to concentrate entirely on their responsibilities without unnecessary disturbance. Therefore, it is important for an AUV to have the capability to recognize the current activity of a diver and its significance. To this end, in the following chapter, we present a diver activity recognition framework for AUVs to analyze and understand different diver activities.
Chapter 6

Semantically-Aware Diver Activity Recognition Framework for Effective Underwater Multi-Human-Robot Collaboration

In the previous chapters, we have introduced a number of vision-based perception algorithms for AUVs to facilitate effective mHRI for successful multi-human-robot collaboration. Specifically, in Chapter 2, we have presented a human-comprehensible AUV-AUV communication framework. In Chapters 3 and 4, we have discussed our contributions towards enabling AUVs to identify unique divers to collaborate with or take instructions from. However, to ensure successful collaboration during a mission, an AUV must have the capability to effectively interact with these unique divers. In Chapter 5, we have presented a diver attention estimation framework for AUVs to facilitate such diver-AUV interactions. It allows an AUV to position itself conveniently before or during the interaction by determining if the diver is attentive. However, there are situations when the AUV needs to assess diver’s actions before attempting to interrupt them, even if it is well-positioned for an interaction. One such scenario arises when the diver is engaged
Figure 6.1: Demonstration of the proposed diver activity recognition framework, making inference on real-world underwater scenarios involving multiple divers and robots. The proposed method is able to learn highly discriminative spatio-temporal features from underwater video clips by focusing on important elements within the scene (as shown in the inset), such as divers, robots, and objects of interest, and their interactions with each other.

In a time-critical task, such as rescuing a fellow dive buddy in distress, requiring swift and focused action. In such instances, it is essential not to disrupt the diver, allowing them to concentrate entirely on their responsibilities without unnecessary disturbance. Therefore, it is important for an AUV to have the capability to recognize the current activity of a diver and its significance.

Although recent terrestrial activity recognition methods (e.g., [157], [158]) have achieved a high degree of accuracy, there is still a lack of research addressing the problem of recognizing underwater diver activity. One of the main reasons behind this is the difficulty associated with underwater data collection. Diver data collection is extremely challenging due to several unique constraints in the underwater domain, such as high
pressure, low visibility, unpredictable currents, extreme temperatures, and reduced situational awareness [21]. Additionally, divers must remain fully focused to ensure the proper functioning of life support equipment. These challenges have contributed to a near-complete absence of underwater diver activity datasets involving multiple divers and underwater robots collaborating with each other, which is necessary to design a deep-learned recognition framework.

In this work, we present a novel recognition framework to analyze and understand diver activities underwater. Additionally, this will enable the AUV to take proactive actions based on the activities of its dive partners and the semantics of the surroundings. For example, when an AUV detects underwater trash and observes a team of divers and robots investigating the area, it can proactively join them to remove the trash, thus increasing the overall efficiency of the collaboration (see Fig. 6.1). The main element of the proposed framework is a transformer-based architecture named DAR-Net (Diver Activity Recognition Network), which can learn highly discriminative spatio-temporal features from underwater human-robot collaborative scenes. Recent research (e.g., [132, 159]) indicates that significant performance improvements can be attained by constraining the spatial area of interest that models learn from. This approach involves focusing the model’s attention on specific regions of the input image, which can lead to more effective learning and better performance outcomes. To this end, we propose supervising the training of DAR-Net with scene semantics to enforce the network to focus on important elements in the scenes that are significant for analyzing the activity of the divers. To enable this, we formulate the first-ever Underwater Diver Activity (UDA) dataset, consisting over 2400 semantically segmented images depicting various activities of divers in a multi-human-robot collaborative setting, with pixel annotations for divers, robots, and objects of interest. We conduct comprehensive experimental evaluations of the proposed framework, showcasing its effectiveness in comparison to various existing State-Of-The-Art (SOTA) activity recognition models. We make the following contributions in this work:

1. We propose an end-to-end transformer-based deep network called DAR-Net to analyze and classify different diver activities in underwater multi-human-robot collaborative scenes. The training pipeline includes a multi-loss objective function that prioritizes important regions to learn from, instead of learning from the whole
image.

2. Additionally, we present UDA, the first-ever Underwater Diver Activity dataset that includes 2460 annotated underwater multi-human-robot collaborative scenes divided into 6 diver activity categories. The data were collected from several closed-water robot trials and contain pixel-level annotations for divers, robots, and objects of interest.

3. Furthermore, we conduct both quantitative and qualitative experiments, demonstrating that the proposed framework derives significant benefits from incorporating additional supervision from semantic labels. This enables the model to learn highly discriminative spatio-temporal features essential for recognizing different diver activities.

6.1 Related Work

Human Activity Recognition (HAR) is an active research area within computer vision and robotics, with research efforts extending over several decades [160, 161, 162]. Human activity can be modeled as a sequence of stochastic actions, characterized by feature vectors encompassing both local motion descriptors and global contextual information. These feature vectors can be encoded in various ways, and novel methods have been proposed to distinguish similar features and group them into the same clusters, thereby giving rise to different HAR frameworks. Sensor-based HAR has been a cornerstone in activity recognition research, owing to its ubiquity and ease of data collection [163]. Early approaches, such as those proposed by Wang et al. [164], utilized wearable sensors to capture motion and orientation data for activity recognition. These methods often employed techniques like Hidden Markov Models (HMMs) and Support Vector Machines (SVMs) to model temporal dependencies and classify activities based on sensor readings. Advancements in deep learning have led to the development of more sophisticated models for sensor-based HAR. For instance, Jiang et al. [165] have used Convolutional Neural Networks (CNNs) for learning discriminative features directly from raw sensor data, achieving superior performance compared to traditional feature engineering approaches.
Vision-based HAR, on the other hand, has gained significant traction due to the proliferation of cameras in various environments, enabling non-intrusive monitoring of human activities [166]. Early work in this domain focused on extracting handcrafted features from images, such as Histograms of Oriented Gradients (HOG) [167] and Local Binary Patterns (LBP) [168], for activity recognition. In recent years, deep learning techniques have revolutionized image-based HAR [169]. Researchers have explored architectures like CNNs to automatically learn hierarchical representations of human activities from image sequences. For example, Simonyan and Zisserman [53] proposed a two-stream CNN architecture that processes both spatial and temporal information from video frames, achieving SOTA performance in action recognition tasks. In general, video-based HAR offers rich temporal information, enabling the modeling of complex human behaviors over time [54]. Early approaches typically involved handcrafted feature extraction from video sequences [170, 171], followed by classification using methods, such as SVMs or decision trees. With the advent of deep learning, however, researchers have developed end-to-end architectures for video-based HAR. For instance, the works in [172] and [55] have introduced 3D convolutional networks, which extend traditional CNNs to capture spatio-temporal features directly from video clips. This approach has been successful in recognizing human actions from videos with varying viewpoints and environmental conditions. Hara et al. [173] have also proposed to use 3D kernels in CNNs to extract spatio-temporal features from videos for activity recognition. Although 3D CNNs are typically susceptible to overfitting due to their large number of parameters, the use of large-scale activity datasets (e.g., UCF-101 [174], Sports-1M [51], ActivityNet [175], Kinetics [176]) for training has mitigated the issue. In contrast, Wu et al. [177] have showed that augmenting 3D CNNs with a long-term feature bank can yield SOTA results in activity recognition task. Furthermore, recent advancements in Large Language Models (LLMs) have facilitated significant progress in the development of robust HAR methods [59, 60].

On the contrary, HAR techniques for underwater domain have received considerably less attention. While there have been a few research endeavors addressing issues, such as diver motion prediction in the context of Underwater Human-Robot Interaction (UHRI) [178, 179, 180] and non-human motion prediction [181, 19], none of these specifically tackle the problem of diver activity recognition. This is primarily due to the
lack of large-scale human activity recognition datasets tailored specifically for underwater environments. Diver data collection is extremely challenging because of several constraints unique to the underwater domain, such as high pressure, low visibility, unpredictable currents, extreme temperatures, and reduced situational awareness [21]. As a result, data collection in different water bodies may not always be feasible. This challenge intensifies in scenarios involving multiple divers and underwater robots within the scene. There are a few diver dataset available in the literature [182, 138], however, they do not include data of divers actively engaged in different activities or tasks, specifically collaborating with AUVs. To this end, we propose to formulate the first-ever Underwater Diver Activity (UDA) dataset involving multiple divers and AUVs, so that it can be used to supervise the learning and validation of diver activity recognition frameworks.

6.2 UDA Dataset

We prepare the UDA dataset by capturing underwater scenes involving multiple divers and AUVs from several robot trials in closed-water environments. This dataset represents the first-ever compilation of diverse diver activities, including scenes where robots and humans are working together. We collect the data as video clips having resolutions of $1920 \times 1080$ pixels, using GoPros [183]. Based on our engagement with professionals who work in the aquatic domain, and our extensive UHRI research endeavors, we have categorized the data into six activities commonly seen in collaborative underwater scenarios involving multiple robots and divers. We chose these six categories as representative activities to demonstrate the efficacy of our proposed framework for the classification task. Increasing the number of categories would require significant effort for data collection and annotation, which is a challenging and also hazardous undertaking. A brief description of the categories are as follows:

1. *Diver Assigning Task to Robot:* This category shows instances where a diver is giving instructions to an observing robot (to fetch an object of interest, for instance). These scenes do not include any robots.

2. *Diver Available:* This category includes images where a diver is present but not actively engaged in a task. Therefore, the observing robot has the opportunity to approach, interact and/or collaborate with the diver.
Figure 6.2: A few sample images and their semantic labels from the proposed UDA dataset. In our dataset, divers, robots, and objects of interest are annotated with purple, red, and green colors, respectively.
3. *Diver Busy*: This category captures images where a diver is occupied with a specific activity (e.g., inspecting the robot for leaks) and not available for interaction or collaboration. The robots found in the scene are not collaborating with the divers.

4. *Diver-Diver Interaction*: This category includes moments where two divers are communicating.

5. *Diver-Robot Collaboration*: This category depicts scenarios in which divers and robots collaborate on a task. Depending on the task’s scope, the observing robot can actively engage and offer assistance.


Since we want to leverage the scene semantics during the training of DAR-Net, we extract sequential frames from the captured video clips and perform pixel-level annotations. We used an open-source platform, called Computer Vision Annotation Tool (CVAT) [184], to perform the annotations. We hosted the platform on our local server and enabled remote access for annotators. Generally, pixel-level semantic segmentation is a time-consuming task because each pixel in the image needs to be carefully labeled according to the corresponding object or class it belongs to. To alleviate this issue, we use the SAM [185] foundation model for image segmentation, which can easily generate segmentation masks for unknown objects. Once generated, we refine the segmented annotations manually, thus streamlining the annotation process and reducing the time and effort required. Our annotations focused on three key elements: *divers*, *robots*, and *objects of interest*, aiming to accurately represent multi-human-robot collaborative scenes. In total, we have prepared 2640 semantically segmented images, with each diver activity category consisting of 440 images. To diversify the dataset, we collected images from various viewpoints and locations. Fig. 6.2 showcases sample images from our dataset alongside their semantic labels. We have made the dataset accessible through the following link: [https://irvlab.cs.umn.edu/uda](https://irvlab.cs.umn.edu/uda)
6.3 Diver Action Recognition Framework

DAR-Net takes RGB video clips as input and extracts robust spatio-temporal features for recognizing various diver activities. It employs a multi-loss objective function and minimizes both classification and segmentation losses.

6.3.1 Feature Extraction

Motivated by the successful learning of maximally discriminative spatio-temporal features for an underwater robotic activity prediction task demonstrated in [19], we choose to use ResNeXt-101 [63] as the feature extractor for our model. ResNeXt is known to be a highly modularized network architecture for image classification, which can extract rich features from the input. It employs a repeating building block prepared with a \textit{split-transform-aggregation} strategy, which aggregates transformations with identical topology. This results in a homogeneous, multi-branch structure with few hyperparameters to set. The method introduces a new term called “\textit{cardinality} of the network”, which represents the size of the set of transformations. It has been shown that increasing the cardinality of the network significantly improves the model’s performance compared to making it deeper or wider. Following [59, 60], we incorporate positional and classification encodings into the feature representation. Both encodings are initialized using the normal distribution $\mathcal{N}(0, 0.02^2)$. This enables each feature location to possess positional information, thereby enhancing the classification process. Subsequently, this feature representation is fed into two separate branches for further downstream processing (see Fig. 6.3).

6.3.2 Objective Function Formulation

The main objective of the work is to robustly analyze and determine different diver activities. We accomplish this by formulating our objective function as a multi-loss optimization problem. Here, the computed spatio-temporal features are processed separately through two different branches. First, a transformer [59] architecture-based \textit{Classification Branch} is utilized to enable the model to focus on important temporal regions of the features, thus grasping the global context effectively. Second, an encoder-decoder architecture-based \textit{Segmentation Branch} is employed to consider spatial scene
Figure 6.3: An overview of the network architecture of DAR-Net. It takes an underwater diver activity video clip as input and extracts highly discriminative spatio-temporal features by incorporating additional supervision from scene semantics. Intermediary skip connections are used to avoid overfitting.

Semantics, thereby capturing the local context. These branches are supervised by the class and semantic labels, respectively. This hybrid learning strategy facilitates robust feature learning for the classification task at hand. Given $x_{n,y_n}$ as the unnormalized logit for the diver activity category $y_n$, where $n$ refers to the $n$-th sample from the mini-batch, we can define the classification cross-entropy loss and the segmentation binary cross-entropy loss functions as follows.

$$L_{class} = -\sum_{n=1}^{N} \log \frac{\exp(x_{n,y_n})}{\sum_{i=1}^{6} \exp(x_{n,i})} y_n$$

$$L_{seg} = -\sum_{n=1}^{N} [y_n \log x_{n,y_n} + (1 - y_n) \log(1 - x_{n,y_n})]$$

where $N$ is the batch size. In total, we minimize the following multi-loss objective function while training our model using the UDA dataset.

$$L = \alpha L_{class} + \beta L_{seg}$$ (6.1)

where the weights $\alpha$ and $\beta$ are set as trainable parameters.
Figure 6.4: The training performance of DAR-Net. Note the convergence in both the classification and segmentation validation loss graphs.

6.3.3 Implementation Details

We use PyTorch libraries [65] to implement DAR-Net. The input data has been resized to have a spatial resolution of $320 \times 256$ and was processed as chunks of 64 sequential frames, representing approximately 3 seconds of video duration. We employ several data augmentation techniques, such as random cropping, image distortion, flipping, normalization, etc., following the recommendations in [66]. For training DAR-Net, we choose a batch size of 4, learning rate of $10^{-5}$, and ADAMW as the optimizer [67] with a momentum of 0.9. The $\alpha$ and $\beta$ values in the multi-loss objective function (Eq. 6.1) were set as trainable parameters. We adopted a similar transformer architecture to that used in [19] and made modifications in the final fully-connected layers. For the segmentation branch, we employ a similar encoder-decoder architecture to that used in [132] to learn from the scene semantics and feed the encoded information into the classification branch.
We used skip connections in the segmentation branch to avoid the vanishing gradient problem. The individual elements of each encoder and decoder block is also shown in Fig. 6.3. We trained our model on an Nvidia RTX6000 Ada Generation GPU for 200 epochs, utilizing cross-entropy loss for the classification task and binary cross-entropy loss for the segmentation task, where we observed convergence in the validation losses, as shown in Fig. 6.4. The performance of the network is evaluated on an AMD Ryzen 9 7950X 16-Core Processor.

6.4 Experimental Evaluations

6.4.1 Evaluation Process and Metrics

To evaluate the performance of the proposed framework, we prepared a test set consisting of 30 video clips depicting underwater multi-human-robot collaborative scenes, distinct from both the training or the validation sets. Each activity category within this set is represented by five video clips. To maintain consistency in our evaluation, the SOTA activity recognition models are also trained on the UDA dataset using the recommended parameters mentioned in their respective papers. Once trained, we compute the prediction performance of DAR-Net and the SOTA models on the test set as follows.

Each test clip is processed as a chunk of 64 frames, representing approximately 3 seconds of video duration. We make the predictions on four-dimensional RGB tensors \((64, 3, h_{im}, w_{im})\) where \(h_{im}\) and \(w_{im}\) are height and width of the image, respectively. First, each test clip is fed into the trained model to compute the classification scores, \(x_{pred} = [x_{pred}^1, \ldots, x_{pred}^6]^T\). Second, the classification scores are converted to probability scores using a softmax function as shown below,

\[ P(x_{pred}^i) = \frac{\exp(x_{pred}^i)}{\sum_{j=1}^6 \exp(x_{pred}^j)} \]

where \(i\) refers to the \(i\)-th category index and can have values in the range \([1, 6]\). Finally, the predicted category is found by selecting the index with the maximum probability score.

We consider the following four metrics for our evaluation:
Table 6.1: Diver activity recognition performance on the test set. Precision, Recall, and F1-Score are computed as weighted averages. The values are in percentages.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>I3D [55]</td>
<td>56.67</td>
<td>60.00</td>
<td>56.67</td>
<td>58.28</td>
</tr>
<tr>
<td>LFB [177]</td>
<td>61.00</td>
<td>60.78</td>
<td>61.00</td>
<td>60.88</td>
</tr>
<tr>
<td>3DResNet [141]</td>
<td>63.33</td>
<td>54.29</td>
<td>63.33</td>
<td>58.46</td>
</tr>
<tr>
<td>SlowFast [54]</td>
<td>56.67</td>
<td>65.00</td>
<td>56.67</td>
<td>57.67</td>
</tr>
<tr>
<td>LateTemporal [60]</td>
<td>66.67</td>
<td>71.25</td>
<td>66.67</td>
<td>65.60</td>
</tr>
<tr>
<td>RRCommNet [19]</td>
<td>60.00</td>
<td>67.64</td>
<td>60.00</td>
<td>61.05</td>
</tr>
<tr>
<td>Ours</td>
<td>73.33</td>
<td>76.90</td>
<td>73.33</td>
<td>72.17</td>
</tr>
</tbody>
</table>

1. **Accuracy**: It measures the overall correctness of the model and is calculated as $\frac{TP + TN}{TP + TN + FP + FN}$. Here, $TP = \text{True Positives}$, $TN = \text{True Negatives}$, $FP = \text{False Positives}$, and $FN = \text{False Negatives}$.

2. **Precision**: It measures how precise the model is in making positive predictions and is calculated as $\frac{TP}{TP + FP}$. Precision is particularly useful when the cost of false positives is high.

3. **Recall**: It measures the sensitivity of the model by measuring its ability to correctly identify positive samples and is calculated as $\frac{TP}{TP + FN}$. Recall is particularly useful when the cost of false negatives is high.

4. **F1-Score**: It combines precision and recall into a single metric, giving equal weight to both measures and is calculated as $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$. This metric can help assess the model’s overall effectiveness in the classification task.

6.4.2 Results

We compare the performance of DAR-Net against six SOTA activity recognition models in terms of average classification accuracy, precision, recall, and F1-score. Table 6.1 presents the quantitative results of our experiments. From the table, we see that the
(a) Without the supervision from semantic labels. (b) With the supervision from semantic labels.

Figure 6.5: The effect of using semantic labels during the training of DAR-Net. The inclusion of scene semantics directs the model’s focus towards relevant regions in the image for feature learning. In contrast, traditional activity recognition models frequently prioritize irrelevant areas, such as pool lane markings during training. Lighter colors indicate higher attention values.

The proposed framework outperforms the rest of the models and achieve a promising classification accuracy of 73.33%. Activity recognition from video (e.g., temporal data) is significantly complicated than image classification from spatial data. This is evident as the SOTA models seem to suffer in accurately classifying the diver activities. Among the SOTA models, only the LateTemporal [60] model achieves a comparable classification accuracy of 66.67%. One potential reason for this could be its use of transformer blocks to leverage the self-attention mechanism [58], enabling it to learn maximally discriminative spatio-temporal features from the diver activity video clips. Additionally, we compute the average precision of all the models and observe superior performance from
DAR-Net, achieving an average precision of 76.90%. To further analyze the authenticity of this precision value, we also calculate the average recall and F1-score of the models. We observe superior performance from DAR-Net in terms of the average recall value and F1-score. This validates that our proposed framework accurately classifies relevant diver activities with low $FP$ and $FN$ values. Note that the accuracy and the weighted average recall values are the same because our dataset is balanced.

Furthermore, we aim to analyze the effect of incorporating additional supervision from semantic labels during the training of DAR-Net. In Fig. 6.5, we visualize attention maps of the activity recognition model, showing the regions in the image it prioritizes during training. Fig. 6.5 illustrates the attention maps from the model trained without supervision from semantic labels. From the figure, it is evident that the model focuses unnecessarily on image regions irrelevant to determining the underlying diver activity category, such as lane markings on the pool bottom and sides. This could contribute to the lower accuracy recorded when scene semantics were not considered during training. In contrast, upon integrating scene semantics into the training process of our proposed framework, the model directs attention solely to crucial image regions, such as divers, robots, and objects of interest, as shown in Fig. 6.5. With DAR-Net's training supervised by semantic labels, the intermediary attention maps naturally prioritize the segmented regions, leading to superior accuracy in identifying diver activity categories. The attention maps in Fig. 6.5b (segmentation masks, in this case) are generated by performing binary thresholding on the feature vectors.

We also show a confusion matrix for the diver activity recognition task (see Fig. 6.6) by recording the predictions made by DAR-Net on our test set consisting of 30 video clips of underwater multi-human-robot collaborative scenarios. As illustrated in the figure, DAR-Net accurately classifies the majority of different diver activities. However, it encounters challenges in accurately identifying certain activity categories, such as Diver Busy and Robot-Diver Interaction. Upon closer examination of video clips from these categories, we observe that both involve a single diver working on or interacting with a robot. To delve deeper into this issue, we calculate the per-category precision, recall, and F1-score for our model and visualize the results in Fig. 6.7. Here, we notice a similar trend where the model’s performance is notably lower in predicting the aforementioned two categories compared to others. For the Diver Busy category, it is evident that the
Figure 6.6: Confusion matrix for diver activity recognition, computed on the 30 video clips from our test set. It highlights the robustness of the proposed framework in accurately identifying the majority of diver activity categories. The matrix entries are normalized.
Figure 6.7: Per-category Precision, Recall, and F1-Score for the proposed diver activity recognition framework. It indicates a similar trend where the model’s performance is notably lower in predicting the Diver Busy and Robot-Diver Interaction categories.

model struggles to detect positive samples, resulting in a low recall value, while also detecting many false positives, indicating a low precision value. On the other hand, the Robot-Diver Interaction category is not consistently detected, but when it is, the model shows high reliability, with high precision and low recall values. In contrast, the model demonstrates superior performance in predicting the other diver activity categories. This discrepancy suggests the need for further investigation to analyze and potentially decouple these two activity categories to mitigate mispredictions.

6.5 Summary

In this work, we have presented an end-to-end framework for AUVs to allow them to analyze and recognize different diver activities. The framework’s ability to extract highly discriminative spatio-temporal features from underwater multi-human-robot collaborative scenes ensures robust recognition performance. This capability allows the AUV to
make informed decisions during collaborative tasks, improving overall task efficiency. Furthermore, we have presented a dataset containing 6 different diver activities spanning over 2400 semantically segmented underwater images for training the proposed framework. Our multi-loss objective function formulation during the training of the framework incorporates additional supervision from underwater scene semantics, prioritizing important regions of the input image for learning, which significantly improves the overall classification accuracy. Through extensive experimental evaluations, we demonstrate the efficacy of the proposed framework, which outperforms several SOTA models in the diver activity recognition task.
Chapter 7

Conclusion

In this dissertation, we have presented several vision-based perception algorithms for Autonomous Underwater Vehicles (AUVs), all aimed at achieving a singular goal: enabling effective underwater multi-Human-Robot Interaction (mHRI) for successful collaboration. We have developed these systems by addressing the challenges specific to the underwater domain and have implemented them on real robotic platforms for validation. Our end-to-end solutions aim to enhance AUVs’ capabilities as dive partners in mHRI missions. Below, we will summarize the technologies we have developed, discuss potential future directions, and conclude with our final remarks.

7.1 Summary of the Developed Technologies

In Chapter 2, a human-comprehensible gestural language detection framework was introduced, representing significant initial progress towards enabling communication during missions involving multiple AUVs and divers. The developed framework can leverage simulated data for initial learning, followed by fine-tuning with real-world data. Through this interaction framework, AUVs can collaborate not only with fellow AUVs but also with human divers, fostering enhanced teamwork in underwater environments. Chapters 3 and 4 presented diver identification methods to ensure that AUVs collaborate with the intended individuals in multi-human-robot teams. Specifically, Chapter 3 introduced a framework that can correctly identify divers in close-proximity interactions using facial cues only. Although diver faces are heavily obscured by their masks.
and regulators, making them look similar to one another, our developed system takes advantage of generative models to extract highly discriminative features from the obscured faces. This framework is the first of its kind which can identify divers using monocular cameras only. Furthermore, Chapter 4 presented an improved and more robust diver identification method using divers’ body joint ratios. This technique is less susceptible to the fact that all divers look similar with dive gear. The systems developed in these two chapters ensure that AUVs collaborate with the correct individuals in multi-human-robot teams. Accurate identification eliminates the possibility of confusion and misinterpretations, paving the way for efficient, secure, and successful collaboration. Chapter 5 introduced a vision-based perception framework for AUVs to determine the attentiveness of divers. In addition, the system includes an intelligent robot controller that can plan a trajectory for the AUVs to position themselves conveniently for interaction if the diver is found to be inattentive. The ability to respond to diver attention levels enhances mutual understanding and cooperation, resulting in superior collaborative efforts. Moreover, this framework ensures that divers have fewer physical and cognitive burdens during mHRI missions as traditionally, it is primarily the diver who swims towards the AUV and positions themselves conveniently to initiate interaction. In Chapter 6 we presented a diver activity recognition framework that allows AUVs to understand and recognize diver activities during missions to gain the insight needed to make informed decisions about when and how to engage with divers at specific moments. We also introduced the first publicly available dataset containing various diver activities and provided pixel-level annotations for them, which will surely benefit the robotics community and researchers from different fields. This framework enables AUVs to take proactive actions based on the activity of their dive partners and the surrounding context, which will ensure increased efficiency and safety in collaborative underwater missions. Lastly, we provided an appendix (Appendix A) where we presented three additional technologies: two end-to-end solutions that improve the visual perception capability of AUVs, and the design and implementation of a low-cost AUV.
7.2 Future Directions

There are several research directions worth exploring to broaden the scope of the research presented in the dissertation.

7.2.1 Separability Analysis for Robust Visual Learning

Given the nature of tasks in the underwater domain, particularly in the context of mHRI, the variability in temporal aspects of input data is limited. For instance, while designing robotic motion-based gestures (Chapter 2), we discovered instances where segments of one gesture’s motion overlapped with those of another, leading to significant resemblance between the two gestures in short motion sequences. A similar pattern emerged in our work on diver activity recognition (Chapter 6), where certain diver activities shared common objects and inter-object relationships in short temporal sequence. This phenomenon undermines the performance of the learning algorithms, particularly in recognizing certain classes. To address this issue, one viable solution would be to conduct a input data separability analysis by devising a metric to quantify class similarity. Each class and its confusion with other classes need to be analyzed against the metric. By conducting such pre-deployment analysis of spatial similarities between classes, better design choice can be adopted while curating or annotating the input data in the first place. This will optimize robot learning and minimize overlaps in spatio-temporal fragments, thus enhancing performance.

7.2.2 Diver Identification in Collaborative Scenes

Humans can identify persons quite accurately under severe occlusion. For instance, a human can differentiate between two divers who are wearing similar-looking dive gear and are even swimming completely horizontally better than an AUV. Although we have enabled diver identification capability in AUVs (Chapters 3 and 4), they require the diver to be in a specific posture for the algorithm to work. Additionally, the framework requires a single diver to be in the robot’s view at a given moment. However, in a multi-human-robot collaborative scenario, there would be multiple divers present in the robot’s view and possibly in various postures. One viable solution would be to consider implementing a multi-pose estimation framework that can determine poses of multiple
people in the scene in 3D. The pose estimation in 3D would relax the requirement of the diver needing to be in a specific posture for the algorithm to work. To minimize the cost of estimation, region proposals [186] can be utilized to run pose detectors in specific regions in the image.

7.2.3 Risk Assessment in mHRI Missions

AUVs are increasingly used in collaborative settings involving humans and other AUVs due to improved on-board computing, availability of affordable AUVs (Appendix A.3), and improved mHRI capabilities (Chapters 2 and 3). With the capability of analyzing and understanding a multi-human-robot collaborative scene (Chapter 6), AUVs can be considered as able dive partners who can make informed decisions by determining the activity of divers and the semantics of the surroundings. However, the AUVs currently lack the capability of understanding task significance in such situations. Employing risk assessment-based task understanding can address this gap by breaking down tasks and associating them with their significance. These tasks’ significance need to be learned by the framework. This will empower AUVs to proactively handle emergency situations.

7.3 Final Remarks

Our developed methodologies, presented in this dissertation, will enable AUVs to be used as intelligent and proactive dive partners in mHRI missions. By recognizing the limitations of traditional approaches, which often rely on surface-controlled Remotely Operated Vehicles (ROVs), we have designed systems for AUVs to revolutionize underwater operations through direct engagement with human divers during missions. Through the development and implementation of vision-based computational methods tailored to AUV perception, we have demonstrated the feasibility of achieving robust mHRI capabilities. Our methodologies enable AUVs to interact comprehensively with multiple AUVs and divers, identify divers for secure collaboration, adaptively position themselves for interaction, and recognize diver activities for informed decision-making. We anticipate significant strides in multi-human-robot collaborative missions through the use of our research.
Appendix A

Additional Contributions

In this Appendix, we present some of our additional research and development contributions towards underwater autonomous robotics as collaborators.

A.1 Underwater Image Super-Resolution

![Underwater Image Super-Resolution Diagram]

Figure A.1: Demonstration of underwater image super-resolution technique where high-resolution image can be generated given a lower-resolution image.

This work [27] presents a generative model for underwater image super-resolution for visually guided AUVs to be able to zoom-in on objects of interest without getting too close and sharpen degraded or blurry images for better scene understanding. Fig. A.1 demonstrates the technique. Since the proposed model requires a supervised method....
(a) A few instances sampled from the HR set; the HR images are of size 640 × 480.

(b) A particular HR ground truth image and its corresponding LR images are shown.

Figure A.2: USR-248 dataset includes one HR set and three corresponding LR sets of images; hence, there are three possible combinations (i.e., 2×, 4× and 8×) for supervised training of underwater image super-resolution models.

to learn the objectives of image super-resolution, my primary responsibility was to formulate a dataset, named USR-248, to facilitate this.

The USR-248 dataset contains a large collection of High-Resolution (HR) underwater images and their respective Low-Resolution (LR) pairs. There are three sets of LR images of size 80 × 60, 160 × 120, and 320 × 240; whereas, the HR images are of size 640 × 480. Each set has 1060 RGB images for training and validation; another 248 test images are provided for benchmark evaluation. A few sample images from the dataset are provided in Fig. A.2. The HR underwater images were collected: 1. using multiple GoPros [187], Aqua AUV’s uEye cameras [6], low-light USB cameras [188], and Trident ROV’s HD camera [189] during various field experiments, and 2. from publicly available Flickr™ images and YouTube™ videos. We avoided multiple instances of similar scenes and made sure they contain different objects of interest (e.g., coral reefs, fish, divers, wrecks/ruins, etc.) in a variety of backgrounds. Fig. A.2c shows the modality in the data in terms of object categories. Once the HR images are selected and resized
to $640 \times 480$, three sets of LR images are generated by compressing and then gradually downsizing the images to $320 \times 240$, $160 \times 120$, and $80 \times 60$. Overall, USR-248 provides large-scale paired data for training $2\times$, $4\times$, and $8\times$ underwater image super-resolution models. It also includes the respective validation and test sets.

### A.2 Underwater Image Segmentation Dataset

This work [132] presents the first large-scale dataset for semantic Segmentation of Underwater IMagery (SUIM). This dataset can play a major role in understanding underwater scenes to benefit applications, such as visual servoing and saliency prediction. I contributed to the compilation of the dataset by manually annotating (pixel-level annotation) a few hundred images.

The dataset contains over 1600 training, validation, and test images with pixel annotations for eight object categories: fish or vertebrates (FV), reefs or invertebrates (RI), aquatic plants or flora (PF), wrecks or ruins (WR), human divers (HD), robots
Figure A.4: (a) LoCO AUV during deployment in the Caribbean Sea off the coast of Barbados, (b) LoCO’s OLED display showing a sample menu.

(RO), sea-floor or rocks (SR), and background waterbody (BW). A few samples from the dataset are shown in Fig. A.3a. The images are of various spatial resolutions, e.g., 1906 × 1080, 1280 × 720, 640 × 480, and 256 × 256. These images are carefully chosen from a large pool of samples collected during oceanic explorations and human-robot cooperative experiments in several locations of various water types. It also utilize a few images from large-scale datasets for underwater research, named EUVP [39], USR-248 [27], and UFO-120 [132]. Fig. A.3b shows the modality in the dataset in terms of object categories.

A.3 Display Interface for LoCO AUV

This work [30] presents the LoCO AUV, a Low-Cost, Open Autonomous Underwater Vehicle. We designed and built LoCO to enhance accessibility to underwater robotics research for research groups with limited resources. LoCO is a general-purpose, single-person-deployable, vision-guided AUV, having a display interface for human-robot interaction. Fig. A.4a shows LoCO during its deployment in the Caribbean Sea. My primary contribution to this work was to integrate an Organic Light-Emitting Diode (OLED) display interface and create an Arduino-based menu system that can communicate via messages with a Robot Operating System (ROS) [38] to control the robot on the fly using either artificial reality tags [15] or hand gestures [76].

LoCO has a monochrome 2.42”, 128 × 64 OLED display [191]. It is controlled by an
Adafruit Trinket Pro microcontroller [192], which is connected to an Nvidia Jetson TX2 via a serial cable. The OLED display is configured to support five menu items, with the provision to include individual sub-menu items. Fig. A.4b visualizes a sample menu of LoCO. The menu options can be connected to a variety of ROS endpoints, including running a launch file, launching a node, calling a service, terminating a node or setting a parameter. These ROS endpoints usually communicate through ROS messages. Hence, I designed a modular Arduino sketch on the microcontroller to receive or send ROS messages via the serial connection for controlling the content shown on the OLED display. This display interface plays a vital role when LoCO collaborates with humans. It is used primarily for AUV-to-human interaction via the menu system, but can also provide important status information to its human partners.
References


