Deep Learning based Underwater Object Detection

Abstract

Autonomous Underwater Vehicles (AUVs) and Remotely Operated Vehicles (ROVs) equipped with an intelligent object detection system play a vital role in various underwater applications such as marine resource exploitation, marine environment monitoring, and marine cable protection. Deep learning based object detection methods have presented great performance advantages over traditional machine learning based methods. However, these deep learning based methods lack sufficient capabilities to handle underwater object detection (UOD) due to these challenges: (1) underwater images acquired in complicated environments suffer from severe distortion which dramatically degrades image visibility, objects in the underwater datasets and real applications are usually small whilst accompanying severe noise that greatly degrade the detection accuracy of UOD tasks. (2) well-annotated underwater data is not sufficient in terms of diversity and amount which highly influences the performance of deep learning models. (3) severely imbalanced data distribution and label noise distribtuion occur in underwater datasets, driving a deep learning model to be more biased towards the majority class.

In this thesis, we aim to address all these challenges, and develop robust deep learning systems to enhance and detect objects in complex underwater images. To achieve this goal, we firstly propose novel perceptual enhancement models to enhance the quality of underwater images. Secondly, we propose a novel Sample-WeIghted hyPEr Network (SWIPENET), and a robust training paradigm named Curriculum Multi-Class Adaboost (CMA), to address the noise and small object detection problems at the same time. Finally, to address the class imbalance problem, we propose a factor-agnostic gradient re-weighting algorithm (FAGR) that can adaptively fine tune the gradients of individual classes according to the distributions

of their detection precision. We have evaluated the proposed methods by conducting extensive experiments on public datasets. Experimental results show the effectiveness of our methods for underwater image synthsis, image enhancement and object detection.

Acknowledgements

First and foremost, I am extremely grateful to my supervisor, Prof. Huiyu Zhou for his invaluable advice, continuous support, and patience during my PhD study. His immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life. I also would like to thank Prof. Junyu Dong, my master supervisor in Ocean University of China, for his continous support and advice during my PhD study. Thanks to my second supervisors Profs Rajeev Raman and Daniel Crooks for their support and assistance.

My work on this thesis was also supported by many people, in particular, all of my colleagues in Prof Zhou's research group (Zheheng Jiang, Liping Wang, Lei Tong, Yunfeng Zhao, Honghui Du, Zhihua Liu, HaiChao Zhu, Fang Chen, Feixiang Zhou, Jialin Lv and Tengyue Li) at University of Leicester. It is due to their kind help and support that have made my study and life in the UK a wonderful experimence. I would like to thank the academic visitors in Prof Zhou's research group (Kun Zhang, Pinggai Zhang, Zhenhua Zhang, Changkui Lv, Aite Zhao and Fengyin Li) for their support, guidance and companionship.

My gratitude extends to the China Scholarship Council (CSC) and University of Leicester for their funding to support my PhD study in the UK. Many thanks to the examinaers for taking the time to read this thesis and providing very valuable and constructive feedback. Finally, I would like to express my gratitude to my parents. Without their tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my study.

Table of contents

List of Acronyms

AP Average Precision

AUC Area Under The Curve

AUVs Autonomous Underwater Vehicles

BAGS Balanced Group Softmax

CycleGAN Cycle-Consistent Adversarial Networks

CL Curriculum Learning

CNN Convolutional Neural Network

CMA Curriculum Multi-Class Adaboost

DCP Dark Channel Prior

DSSD Deconvolutional Single Shot Detector

DL Deep Learning

DNN Deep Neural Network

ENR Effective Number-based Re-balancing algorithm

FAGR Factor-agnostic Gradient Re-weighting

FP False Positive

FPS Frames Per Second

FN False Negative

FRR Frequency-based Re-balancing algorithm

GAN Generative Adversarial Network

GDCP Generalised Dark Channel Prior

GDL Gradient Difference Loss

GMM Gaussian Mixture Model

GT Groud Truth

HOG Histograms of Oriented Gradients

IMA Invert Multi-Class Adaboost

IMOS Integrated Marine Observing System

IoU Intersection over Union

LBP Local Binary Pattern

LSTM Long Short-term Memory

mAP mean Average Precision

MSE Mean Square Error

ML Machine Learning

ROVs Remotely Operated Vehicles

ReLU Rectified Linear Unit

RF Random Forest

ROC Receiver Operating Characteristic

ROI Region of Interest

RPN Region Proposal Network

SC Shape Context

SSD Single Shot MultiBox Detector

SSIM Structural Similarity

SGD Stochastic Gradient Descent

SIFT Scale-invariant Feature Transform

SOTA State-of-the-art

SWIPENET Sample-WeIghted hyPEr Network

SVM Support Vector Machine

MA Multi-Class Adaboost

MFF Multi-scale Contextual Features Fusion

MBP Multi-scale Blursampling

NECMA Noise-eliminating Stage of CMA

NLCMA Noise-Learning Stage of CMA

NR Noise Removal

PSNR Peak Signal-to-Noise Ratio

PCQI Patch-based Contrast Quality Index

UCIQE Underwater Color Image Quality Evaluation

UDCP Underwater Dark Channel Prior

UIE Underwater Image Enhancement

UIS Underwater Image Synthesis

UICM Underwater Image Colorfulness Measure

UIConM Underwater Image Contrast Measure

UIQM Underwater Image Quality Measure

UISM Underwater Image Sharpness Measure

UOD Underwater Object Detection

VGG Visual Geometry Group

List of figures

List of tables

Chapter 1

Introduction

1.1 Preliminary of Underwater Object Detection

The ocean is the beating blue heart of our planet and the largest habitat on the Earth. It covers approximately 70% of the Earth's surface and holds 97% of the Earth's water. It plays a very foundational role in supporting both the environment and the economy. For our environment, it produces more than half of the oxygen and absorb most of the carbon. It also provides the biggest habitats for the natural organisms. Sea grass meadows and coral reefs are the foundation of marine food chains, habitat provision and nutrient cycling. If marine species (such as coral reefs and sea grasses) are damaged, the global climate will be largely deteriorated and human food resources will be largely reduced. For our economy, maintaining the ocean ecosystem services is also important. Scientific ocean exploration and exploitation help us to effectively manage, conserve, regulate, and use ocean resources. However, our understanding of the ocean remains limited for a long time due to scarce tools and technologies to explore the ocean world. Fortunately, vision-based object detection technique [2, 3], as an effective way to explore the ocean, offers enormous potential to make the exploration and exploitation more intelligent and efficient.

In the past, human beings explore and exploit the ocean in an traditional way. For example, in the ocean research field, researchers try to quantify human impacts on fish biodiversity to preserve the marine ecosystems. People fish by fishnet or visit the fish

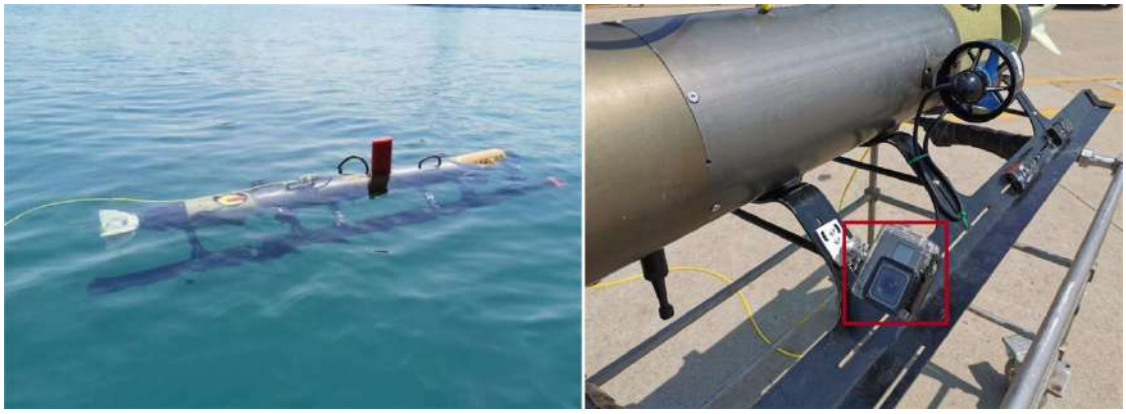

Fig. 1.1 AUVs with GoPro cameras (the images come from [1]).

community by diving into the ocean world [4]. These quantification approaches help us to conduct in situ sampling of the fish community, but cannot collect sufficient data and require considerable human physical resources. In recent years, Autonomous Underwater Vehicles (AUVs) [5, 6] and Remotely Operated Vehicles (ROVs) [7, 8] equipped with intelligent underwater object detection systems offer us opportunities to explore and protect the ocean resources. Many research institutes and scientists attached underwater cameras to AUVs and ROVs, which assist human beings to monitor the underwater environments and perform different underwater tasks, such as marine organism capturing, environment surveillance and biodiversity monitoring. An intelligent underwater object detection system is an indispensable technology to fulfill these tasks. As shown in Fig. 1.1, researchers attached the cameras to an AUV to collect underwater images and videos. The intelligent underwater object detection algorithm embedded in the camera system is able to recognise, detect and count the marine organisms. The cameras can record the underwater world for a long time that provides abundant data for oceanography and fishery science research. Underwater vision-based applications are increasingly developed in marine ecology studies and marine management. Unfortunately, complicated underwater environments and lighting conditions introduce considerable noise into the captured images and videos, which has posed massive challenges to intelligent vision-based object detection systems [9, 10]. Therefore, it

is crucial to develop robust underwater object detection techniques which effectively handle the challenges in the underwater scenes for AUVs and ROVs.

1.2 Challenges of Underwater Object Detection

The underwater environment is one of the most challenging conditions for object detection. Underwater object detection faces several challenges. Firstly, underwater images or videos collected in the underwater scenes are of very low quality and contain considerable small-size objects degrades the accuracy of the detection frameworks [9, 11]. In the underwater scenes, the light received by any camera suffers from wavelength-dependent light absorption and scattering caused by the particles in the water [12]. Light absorption leads to the loss of image information and serious color distortion while the light scattering produces haze-effects, reduces image contrast, and suppresses image details. These negative effects pose great challenges on an underwater detection framework. Moreover, the detection performance has also been affected by shadow noise, non-uniform illumination, camera shaking and complex background interference. Hence, underwater image enhancement, as the preprocessing procedure, is commonly used to assist underwater vision tasks.

Secondly, well-annotated underwater data is not sufficient in terms of diversity and amount [13]. The performance of machine learning models is highly influenced by the size of the available datasets. For example, deep learning is a part of machine learning, and the established deep learning models trained on few training examples usually present poor performance, because they are likely to suffer from the over-fitting problem (a model fits exactly on the training data but cannot perform accurately on new data). For the underwater detection problem, previous detection frameworks have been designed and evaluated on underwater datasets collected under constrained conditions. The scale of the datasets is very small because the annotation of the underwater objects is extremely labor- and time-consuming. As reported in [14], most of the constructed underwater data has little human annotations, for example, the Integrated Marine Observing System (IMOS) collected millions of images of coral reefs around Australia, but less than 5% of the data accompanies marine

expert analysis. For the National Oceanic and Atmospheric Administration, the rate of expert analysis is even lower, only 1–2%. The object detection frameworks, especially deep learning based object detection frameworks, trained on these constrained datasets, cannot achieve satisfying performance in real-world applications.

Thirdly, current state-of-the-art detection frameworks have limited generalisation and robustness capabilities. They cannot learn effective feature representation of datasets with imbalanced data distributions and imbalanced label noise distributions. Large-scale datasets with balanced and correct human annotations bring large performance advantages to machine learning models, however, machine learning models may fail when the annotation quality of the training data cannot be guaranteed.

1.3 Previous Works of Underwater Object Detection

Machine learning algorithms have been employed in underwater recognition and detection tasks for a long history. The traditional machine learning approaches designed hand-crafted features for underwater object detection. Some of them selected shape, color, or texture features. For example, Beijbom et al. [14] extracted texture and color features and exploited Support Vector Machine (SVM) as the classifier to detect the underwater corals of multiple scales. Kim et al. [15] proposed an underwater object detection method based on multi-template object selection and color-based image segmentation. Chuang et al. [16] extracted texture features using phase Fourier transform for detecting fishes. Several machine learning algorithms employed more complex features such as scale-invariant feature transform (SIFT) [17], Histogram of Oriented Gradients (HOG) [18], or shape context (SC) [19].

For some specific underwater objects or datasets, these carefully selected hand-crafted features have led to good performances, however, they may fail if we use a new data set. This is because the hand-crafted feature-based methods have several disadvantages: (1) These methods are task-specific and limited in the generalization capability. The features designed for weak illumination scenes may not be suitable to the underwater scenes with sufficient illuminations. Also, once the objects to be detected change significantly, the corresponding

features may not be suitable to the detection task; (2) The hand-crafted feature extraction and classification has been established independently, therefore, the extracted features may not be suitable to the developed classifiers, leading to poor classification performance. For example, Villon et al. [18] first extracted Histogram of Oriented Gradients (HOG) features from the underwater images, then they employed Support Vector Machine (SVM) as the classifier for fish classification. The performance of this HOG+SVM framework lags far behind the end-to-end deep learning framework in [18]. Moreover, it takes considerable experiences to propose and valid an effective hand-crafted features. Instead, the supervised deep learning algorithms can automatically extract features from big data.

Deep learning is a specialized subset of machine learning. A deep learning model uses a layered structure to analyze data, and its layered structure is inspired by the biological network of neurons in the human brain, which can learn discriminate knowledge from big data[20]. A good deep learning model requires lots of training data, from which it extracts useful and discriminate features. The biggest difference between traditional machine learning and deep learning algorithms is that deep learning requires less human interventions. The traditional machine learning models are trained to simulate specific functions or carry out specific tasks. When the tasks have changed, they need some human intervention to some degree, human experts have to step in and adjust the used features or classifiers. Differently, deep learning architectures can effectively learn features from the input data with less human intervention.

Contributing to large amounts of training data, deep learning networks have shown promising performance in various computer vision and image understanding tasks. For example, convolutional neural networks pre-trained on the large scale data set ImageNet [21] have achieved unprecedented successes in image classification, image segmentation, object detection, tracking and so on. Moreover, deep learning has also been widely deployed in underwater object detection. Choi [22] applied convolutional neural network (CNN) to classifying fish species, while Villon et al. [18] employed a deep learning model to detect coral reef fishes. Li et al. [23] directly exploited the commonly used general object detection framework Fast-RCNN to detect fish species, lately, they applied the faster detection

framework Faster-RCNN [24] to accelerate fish detection. To meet the real-time detection requirements, Yang et al. [25] applied the real-time detection framework YOLOv3 [26] for underwater object detection. Even though deep learning detection models show large advantages over the traditional machine learning detection models, they still cannot well-handle noisy data and the class imbalance problem. Deep learning models cannot effectively detect the small objects in some cases, leading to high false positives and false negatives. Hence, efforts are still needed to handle challenging problems in deep learning based underwater object detection.

1.4 Contributions of This Thesis

We here developed a robust frameworks for underwater image enhancement (UIE), underwater image synthesis (UIS), and underwater object detection (UOD). The final objective is to achieve robust detection performance in noisy and challenging underwater scenes with little human intervention. We make the following contributions to achieve the objective:

1. To improve the quality of the underwater images, we propose a novel deep detection-perceptual underwater image enhancement model to generate detection-favouring images to improve the detection accuracy. To our knowledge, this is the first practice for underwater image enhancement, aiming to generate detection-favouring rather than visually pleasing images. Our proposed perceptual enhancement model outperforms several state-of-the art UIE algorithms using image quality evaluation metrics and task-related detection accuracy metrics on both synthetic and real-world underwater datasets.

2. To generate sufficient training data for training our proposed underwater image enhancement model, we propose a novel hybrid underwater image synthesis model which can synthesise underwater images from high-quality in-air images. Different from previous underwater image synthesis models, our proposed hybrid synthesis model incorporates both physical priors and data-driven cues to generate more realistic underwater images in terms of color distortion, haze-effects and diversity, enabling our underwater image enhancement model to be generalised to handle real-world underwater scenes.

3. To address the noisy data problem and improve the small-size object detection accuracy in underwater object detection, we propose a novel deep detection framework, which consists of a powerful backbone network SWIPENET and a noise-immune training paradigm Curriculum Multi-class Adaboost (CMA). The SWIPENET+CMA framework trains a robust deep ensemble detector for the object detection task in the underwater scenes with heterogeneous noisy data and small objects. SWIPENET fully takes advantage of both high resolution and semantic-rich Hyper Feature Maps that significantly enhance small object detection, and CMA controls the influence of the noisy samples on SWIPENET according to their weights to address the noisy data problem. To achieve the balance between detection accuracy and computational costs, we propose a selective ensemble algorithm to choose the best detector trained with large data diversity.

4. To address the imbalance detection problem in underwater object detection, we propose a novel factor-agnostic gradient re-weighting (FAGR) algorithm, which produces the precision balanced gradients of all the classes and re-balance the precision distributions. The proposed class imbalance algorithm FAGR, by re-balancing precision distributions, performs much better than those classic imbalance algorithms which re-balance the data distributions. Moreover, we provide theoretical analysis and empirical evidence to the problem that imbalance data distributions are not the only cause of imbalance detection. This is the first work to report that imbalanced label noise distributions also lead to the imbalance detection problem.

1.5 Thesis Outline

Chapter 2 gives an overview of the related literature. We will discuss the state-of-the-art algorithms on underwater image enhancement, underwater image synthesis and underwater object detection.

Chapter 3 presents a novel detection-perceptual enhancement model and a novel hybrid underwater image synthesis model. The detection-perceptual enhancement model aims to generate images that can boost the detection accuracy of the following detection framework,

while the hybrid synthesis model aims to synthesise more realistic underwater images for training the enhancement model.

Chapter 4 presents a novel detection framework SWIPENET+CMA. We first introduce our proposed backbone network SWIPENET which is designed for improving small object detection. Then, we move on to describe the noise-immune training paradigm Curriculum Multi-class Adaboost (CMA), which aims to address the noisy data problem.

Chapter 5 first analyses the factors that lead to the class imbalance problem in the underwater object detection, and discover that the imbalance data distributions are not the only cause of imbalance detection. Then, we present a novel factor-agnostic gradient re-weighting (FAGR) algorithm to address the class imbalance problem in the underwater object detection.

Chapter 6 summarises our contributions and discusses future directions.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, we first introduce some background knowledges and review the related works for underwater object detection (UOD), underwater image enhancement (UIE) and underwater image synthesis (UIS) in the literature. For better readability, our literature review is illustrated as a tree structure shown in Fig. 2.1. Underwater object detection algorithms perform poorly with blurry and noisy underwater images. Underwater image enhancement is commonly used to improve the visual quality of underwater images so that we can improve the performance of the underwater object detection frameworks. Deep learning based underwater object detection and underwater image enhancement frameworks

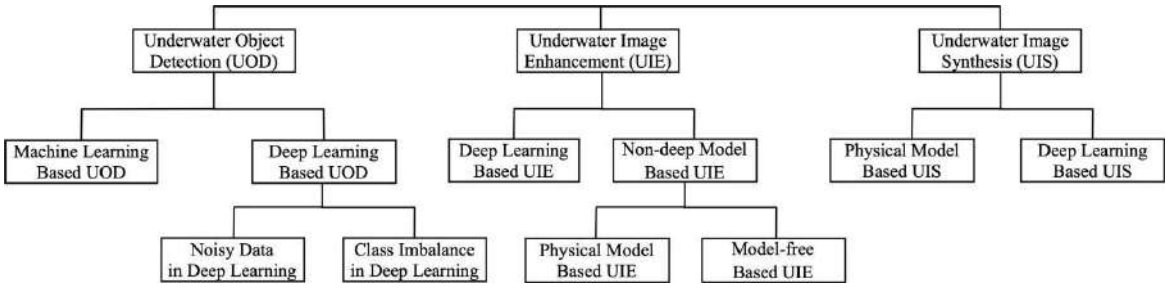

Fig. 2.1 The tree structure of our literature review.

highly rely on the amounts of training data, and underwater image synthesis is an effective way to synthesize more data for training deep learning frameworks.

2.2 Underwater Object Detection (UOD)

In this section, we first review the related work of traditional machine learning and deep learning algorithms for underwater object detection. Table 2.1 summarises the well-known traditional machine learning and deep learning approaches for underwater object detection, including the datasets, the features and the classifiers used in each of the approaches. Then, we move on to introduce two main challenges in underwater object detection, i.e. the noisy data and the class imbalance problems.

2.2.1 Traditional Machine Learning based Underwater Object Detection

Traditional machine learning algorithms have been employed in underwater object detection for many years. They can be diveded into sonar- and camera-based underwater object detection. These two approaches have their advantages and disadvantages.

Underwater scenes often present large decrease in visibility because the signals received by sensors has been absorbed and distorted by water bodies. Sonar sensors have been widely applied to the collected underwater data because they can provide relatively reliable data regardless of scene visibility. Sonar sensors are sensitive to geometrical structure information and can provide structural information of underwater scenes even in low-visibility environments. In general, two sorts of sonars are widely used in sonar-based underwater object detection, including side-scan sonar (SSS) and multi-beam forward-looking sonar (FLS). SSS provides long range, high resolution data, and allows for performing detection in vast survey areas (i.e. hundreds of meters long survey tracks), while FLS allows for a closer, more in-detail inspection of possible underwater object locations. Hayes et al [36] showed that the high-resolution imagery provided by SSS is suitable for detecting possible locations of objects laying on the seafloor of vast surveyed areas. Nonetheless, after this

Table 2.1 Summary of traditional machine learning and deep learning approaches for underwater object detection.

Ref	Year	Dataset	Features	Classifier
[27]	2008	Videos of coral reef transects from the Great Barrier Reef	Local Binary Pattern (LBP) and Normalized Chromaticity Coordinates histogram	Linear discriminant analysis followed by a three layer neural network.
[28]	2014	Moorea Labeled Corals and Heriot-Watt University Atlantic Deep Sea Digital dataset	Color, shape and texture features	Convolutional Neural Networks (CNNs)
[23]	2015	RBG images and videos from LifeCLIEF Fish Task of ImageCLIEF	RGB color space	Fast R-CNN
[18]	2016	Marine Biodiversity Exploitation and Conservation Dataset	Motion from previous sliding window	Convolutional Neural Networks (CNNs)
[29]	2016	National Data Science Bowl	Multi-scale deep learning features	Convolutional Neural Networks (CNNs) inspired by GoogleNet
[30]	2016	Woods Hole Oceanographic Institution (WHOI-Plankton) dataset	Transfer learning to address class imbalance	Convolutional Neural Networks (CNNs).
[31]	2016	ZooScane System Dataset	Data augmentation to reduce the overfitting	ZooPlanktoNet
[32]	2016	Moorea Labelled Coral (MLC) dataset	Color, texture and Convolutional Neural Networks (CNNs) features	VGGNet
[33]	2020	URPC2017 and URPC2018 datasets from the Underwater Robot Picking Contest	Invert Multi-class Adaboost to address the noisy data problem	SWIPENET
[7]	2020	URPC2017 and URPC2018 datasets from the Underwater Robot Picking Contest	Curriculum Multi-class Adaboost to address the noisy data problem	SWIPENET
[34]	2020	URPC2018 dataset from the Underwater Robot Picking Contest	Data augmentation method RoIMix to boost performance	Improved Faster RCNN
[35]	2020	Underwater detection dataset UWD	Multi-scale contextual features	FERNet

large-scale detection one might need to reacquire the detected targets for a closer and more in-detail inspection task, for which FLS is a more suitable option. Galceran et al. [37] proposed a real-time underwater object detection algorithm for detecting manmade objects in images collected by multibeam forward-looking sonars. This work applied integral-image representation to extracting features without requiring training data, and largely reduced the computational overload by working on smaller portions of underwater images. Sonar images extend the perceptual range, but they are less intuitive and cannot be easily understood by human because of missing visual features. Moreover, sonar images contain a considerable level of noise, so it is difficult to ensure the reliability of sonar image recognition and analysis.

In contrast to sonars, cameras can produce a large number of visual images at high spatial and temporal resolutions. Vision based underwater object detection has advanced due to various traditional machine learning approaches which applied hand-crafted features to recognising underwater objects. Strachan et al. [38] used color and shape descriptors to recognise fish transported on a conveyor belt, monitored by a digital camera. Lee et al. [39] used contour matching to recognise fish in the fish tanks. Spampinato et al. [40] presented a vision system for detecting, tracking and counting fish in video, which consist of video texture analysis, object detection and tracking processes. Larsen et al. [41] used shape and texture as features, and LDA as classifier to recognise fish species. Huang et al. [42] presented a Balance-Guaranteed Optimised Tree (BGOT) algorithm to control error accumulation in hierarchical classification. They carried out an experiment on a data set containing 3179 fish images of 10 species collected from underwater videos and got the accuracy of 95%. Lately, they further used GMM to improve the reject rate in hierarchical classification. Using the proposed BGOTþGMM, they achieved the average precision of 65% on a large dataset.

The traditional machine learning algorithms also have diadvantages that they cannot handle complex real-world underwater images or applications, because they are restricted by hand-crafted features and limited experimental datasets: (1) most of them are tested on simple underwater datasets captured under constrained conditions such as human-made tanks; (2) the experimental datasets are relatively small, and these classical algorithms work well

on small datasets but fail to work on large datasets; (3) all of them use hand-crafted features, therefore, they only work for some special tasks or on some datasets. These classical methods are task-specific and limited in their generalisation capabilities. As we have known, our goal is to design an accurate and robust underwater object detection system working on large and unconstrained datasets.

2.2.2 Deep Learning based Underwater Object Detection

Traditional machine learning approaches heavily rely on hand-crafted features, which have a limited representation ability. To improve the detection accuracy, several researchers introduced deep learning frameworks to produce deep features for underwater object detection. Villon et al. [18] compared a deep learning method against the Histogram of Oriented Gradients (HOG)+Support Vector Machine (SVM) method in detecting coral reef fish, and the experimental results show the superiority of the deep learning method in underwater object detection. Li et al. [23, 43] exploited Fast RCNN and Faster RCNN to detect and recognise fish species. However, these established methods used the features from the last convolution layer of the neural network, which is coarse and cannot effectively detect small objects.

To improve feature representation, Fan et al. [35] proposed a stronger backbone named FERNet to exploit multi-scale contextual features, and an anchor refinement module was also introduced to solve the problem of sample imbalance. To achieve high efficiency, Wang et al. [13] proposed a lightweight network named UnderwaterNet, which incorporates a Multi-scale Contextual Features Fusion (MFF) block and a Multi-scale Blursampling (MBP) module to reduce the network parameters. Since the underwater images suffer from severe distortion that greatly degrades detection precision, Liu et al. [9] first applied an underwater image enhancement algorithm to improve the quality of underwater images, then they trained and tested their deep detection networks using the enhanced underwater images to improve the detection precision. Chen et al. [44] proposed a perceptual enhancement model to generate detection favourable images rather than the images of high visual quality. In addition, underwater object detection datasets are so scarce that hinders the development

of underwater object detection techniques. Jian et al. [45, 46] proposed the OUC underwater dataset for underwater saliency detection with object-level annotations that can be used to evaluate the existing systems. Lin et al. [34] proposed a data augmentation method RoIMix that focuses on interactions between images and mixes the region proposals from multiple images. This proposal-level data augmentation strategy greatly improves the performance of their underwater object detector.

Deep learning based underwater object detection also faces two main challenges, including the problems of noisy and class-imbalanced data. Since these two challenges are the focuses of our thesis, we review the related works in the following subsections.

2.2.2.1 Noisy Data in Deep Learning

Deep learning based underwater object detection frameworks largely rely on well-annotated large-scale datasets. However, incorrect labeling may be accidentally neglected in complex data annotations [47–49]. Image annotations fully depend on subjective human judgments which are not always precise [50, 51]. In underwater object annotation, it is quite often to generate false labels due to low visibility and poor contrast of underwater images. Deep neural networks trained over these noisy underwater datasets produce much less satisfactory results [33, 7].

Sample re-weighting has been widely used to address the noisy data problem [52] or hard sample mining [53]. It usually assigns a weight to each sample and then optimises the sample-weighted training loss function. These established sample re-weighting methods can be divided into training loss and testing results based methods. For the training loss based sample re-weighting approaches, we may have two research directions. For example, focal loss [54] and hard example mining [53] emphasise on hard samples with high training losses while self-placed learning [55, 56] encourages learning easy samples with low losses. These two possible solutions take different assumptions over the training data. The first solution assumes that hard samples are informative and should be learned more, whilst the second one assumes that hard samples are prone to be noise. In underwater object detection tasks, hard samples are probably not useful because they confuse the detector rather than

helping it. Different from the training loss based sample re-weighting methods, Multi-Class Adaboost [57] re-weights the samples according to the testing classification results. This method focuses on learning misclassified samples by increasing their weights during the iteration. However, both training loss and testing results based sample re-weighting methods cannot well-handle the noisy data problem because it is impossible to perfectly distinguish the noisy data from the clean data according to the training loss or the testing results.

Sample selection is another intuitive way to address the noisy data problem. This approach is able to filter out the mislabeled data and to leave the clean data in the training data for the latter training process. Sample selection is straightforward but arguably sub-optimal, because the filtered data may contain clean data which are meaningful for the model generalisation. For example, the Invert Multi-class Adaboost (IMA) algorithm [33] regarded the undetected objects as the noisy data and ignored learning them in the later training exercises. It is able to alleviate the influence of the noisy data and reduce the detection errors in underwater object detection, however, it inevitably excludes several hard but clean data in addition to the noisy data. The undetected objects contain considerable hard but clean data. If we directly discard them, considerable hard training samples will be lost and the detectors cannot well-detect these hard data which limit their generalisation.

To make full use of all the available data, Chen et al. [7] introduced an easy-to-hard learning paradigm, Curriculum Multi-Class Adaboost (CMA), which learned easy data first and then moved on to learn hard data. CMA is motivated by the curriculum leaning paradigm, which is designed to imitate the human learning system. In the human education system, it may confuse the learner if s/he directly learns the hard knowledge in the beginning. Instead, the beginner starts from learning easy knowledge while skipping disturbing hard knowledge. In such way, the learning exercise is efficient and effective [58, 59]. This idea has been widely used in many machine leaning algorithms. For example, curriculum learning [60] and self-pace learning [55, 56] are two representatives inspired by the idea of learning easier aspects of the task before moving into a difficult level. Both approaches have been reported to provide better generalisation for the used model in [60, 55, 56]. However, curriculum learning requires to rank the samples in the datasets in the order of incremental difficulty

levels, but preparing such datasets is not trivial at all in practice. Self-pace learning addresses the sample order issue by training the used model and ranking the samples according to the samples' loss values using the learned model. It assumes the samples with low loss values are easy samples. One major drawback of self-pace learning is that it does not incorporate prior knowledge into the learning and hence loose the generalisation ability. In addition, both methods only train a single model without considering its capacity to learn diverse data. The developed models may be over-fit on some samples and under-fit on other samples. CMA combined the learning tricks from curriculum learning and Multi-Class Adaboost into a novel noise-immune training paradigm, which dynamically trains multiple detectors on the samples with a large diversity and combines them into a unified noise-immune deep ensemble detector.

2.2.2.2 Class Imbalance in Deep Learning

Imbalance data distributions commonly exist in the real-world datasets [61–63]. The class imbalance problem has been widely investigated in general classification and detection tasks but has not attracted much interest in the underwater object detection task. The class imbalance problem has influenced the generalisation performance of deep classification networks [64–66]. Standard solutions usually re-balance data classes according to their data distributions via re-sampling [67] or re-weighting [68]. Re-sampling [67, 64, 69] aims to produce roughly uniform distributions of classes, including over-sampling (repeating data for minority classes) and under-sampling (removing data for majority classes). Re-sampling generates a relative balanced data distribution, however, it may either introduce large amounts of duplicated samples, which slows down the training and makes the model susceptible to overfitting due to over-sampling, or discard valuable examples that are important for feature learning due to under-sampling.

Re-weighing [54, 52] is another widely used re-balancing strategy in classification, which usually highlights classes by augmenting their sample weights. During the training on imbalanced datasets, minority classes tend to have higher losses than majority classes as the features learned from minority classes are usually less discriminative. To address

the imbalanced classification problem, several class frequency-based re-weighing methods [67, 64] have been proposed. These methods usually assign sample weights inversely proportionally to the class frequency. The training samples of minority classes have been assigned large weights while the training samples of majority classes have been assigned small weights in loss functions. These simple heuristic methods have been widely adopted. However, recent works [69, 70] show these methods cannot effectively handle large-scale real-world scenarios and tend to meet optimisation difficulty. Instead, Mahajan et al [69] proposed a balanced version of weights that are empirically set to be inversely proportional to the square root of class frequency. Cui et al. [68] proposed to adopt the effective number of samples instead of proportional frequency as a weight to balance the data.

More recent approaches employed metric learning and hard negative mining strategies to address the class imbalance problem in classification. A variety of novel learning losses have been proposed to implement metric learning and hard negative mining strategies, such as range loss [71] and lifted structure loss [54]. The range loss is designed for deep face recognition with imbalanced training data, which can reduce overall intrapersonal variations while enlarging interpersonal differences simultaneously. Liu et.al [72] combined the tricks from metric and meta learning, and proposed a dynamic meta learning to address the class imbalance problem. Data distribution based re-balancing is able to improve the accuracy of minority classes in a classification task, however, directly adopting these methods to detection frameworks cannot achieve satisfactory performance due to the intrinsic difference between detection and classification [73].

To address the class imbalance problem in the object detection tasks, Lin et. al [54] reported a focal loss to down-weight the loss of well-classified easy samples which may drive the optimiser to pay more attention to the hard samples. Li et. al [73] presented a balanced group softmax (BAGS) module for optimising the classifiers within the detection framework through group-wise training. It implicitly modulates the training process for the majority and minority classes and ensures they are both sufficiently trained, leading to state-of-the-art (SOTA) detection performance on the LVIS [74] dataset. The class imbalance problem has been extensively studied in the general classification and detection tasks, unfortunately,

previous works on underwater object detection rarely discuss the influences of class imbalance in noisy underwater scenes.

2.3 Underwater Image Enhancement (UIE)

Underwater image enhancement is an indispensable step to improve the visual quality of underwater images and can be categorised into the following three groups: model-free, physical model based, and deep-learning based methods.

2.3.1 Model-free Image Enhancement Methods

Model-free UIE methods [75–77] aim to adjust image pixel values to improve the visual quality without referring to any physical imaging model. Iqbal et al. [78] proposed an unsupervised color correction method to balance the color channels and correct the contrast in both RGB and HSI color spaces. To further improve the approach in [78], Chani et al. [79] proposed the Rayleigh-stretched contrast-limited adaptive histogram method to enhance underwater images, which effectively reduced the number of over/under-enhanced regions. Ancuti et al. [75] proposed a fusion-based underwater image enhancement method by fusing a contrast enhanced underwater image and a color corrected image in a multi-scale fusion strategy. Later on, Ancuti et al. [80] fused two images derived from a white-balanced version of the underwater image with corresponding weighted maps in a multi-scale way, and important faded features and edges are recovered in the enhanced images. Fu et al. [77] presented a two-step approach for underwater image enhancement, which includes a color correction algorithm based on piece-wise linear transformation and a contrast enhancement algorithm.

Another research line of underwater image enhancement is based on the Retinex theory [81] of fish visual perception. The Retinex theory is supported by previous studies [82, 83] which discovered that sea species have evolved to own highly adapted visual perception systems with the specific mechanisms of spatial resolution, contrast sensitivity and color discrimination. The retinal structure of fishes mainly consists of photoreceptors, horizontal

cells, bipolar cells, and ganglion cells, which have different architectures and functions [84, 85]. By mimicking the architectures and functions of these cells, different Retinex-based underwater image enhancement algorithms are able to restore background illumination and lightness. Fu et al. [76] proposed a variational Retinex-based method for underwater image enhancement, which consists of color correction, layer decomposition and enhancement. Zhang et al. [86] extended the standard Retinex-based method by utilising bilateral and trilateral filters on the three channels of the image in a CIELAB color space. Li et al. [85] mimicked the adaptive mechanisms of the teleost fish retina to construct the enhancement model, which can effectively eliminate the haze and the nonuniform color bias. The model-free methods can improve visual quality to some extent, but may accentuate noise, produce artifacts, and introduce color distortion.

2.3.2 Physical Model based Image Enhancement Methods

Physical model based methods [87, 88] treat underwater image enhancement as an inverse problem of image degradation. These methods usually establish a physical underwater image degradation model, and then estimate the unknown model parameters using various prior assumptions. Finally, high quality images can be retained by inverting this degradation process. For physical model based methods, estimating the accurate unknown model parameters (e.g. the medium transmission) is a key point to performance enhancement. Many research works focus on exploring effective image priors to achieve the satisfactory medium transmission. Among them, the popular dark channel prior (DCP) [89] has been widely studied and used to enhance the underwater images.

Drews et al. [87] proposed an underwater dark channel prior (UDCP) in order to adapt the dark channel prior into underwater scenes and enhance the quality of the underwater images. UDCP helps to estimate more accurate medium transmission than DCP; however, the prior does not always hold when underwater images contain white objects or artificial light. Peng et al. [88] proposed a generalised dark channel prior (GDCP) to estimate medium transmission, which incorporates adaptive color correction into an image formation model. Based on the observation that colors associated to different wavelengths have different attenuation rates in

the water, Galdran et al. [90] proposed a variant of the dark channel prior algorithm [89], namely Red Channel, which recovers the lost contrast of an underwater image by restoring the colors associated with short wavelengths.

Instead of using the DCP [89] prior, Zhou et al. [91] exploited the color-line prior to recover the color line of image patches and well-handled the scattering and absorption problems. Inspired by the fact that the dark channel of an underwater image tends to be a zero map, Liu et al. [92] proposed a cost function to improve image contrast, then achieved optimal transmission maps which maximise image contrast by minimising the cost function. Wang et al. [93] proposed a novel underwater image restoration method based on adaptive attenuation-curve priors. They also set up the saturation constraint to adjust medium transmission to prevent over saturation and reduce noise. Li et al. [94] employed a random forest regression model to estimate medium transmission of underwater scenes. Peng et al. [95] estimated the scene depth via image blurriness and light absorption, and employed the estimated depth to enhance underwater images. Moreover, some works try to combine multiple enhancement algorithms to achieve better performance. Chiang [12] combined an image de-hazing algorithm with an wavelength dependent compensation algorithm to remove the color distortion of underwater images and the negative effects caused by artificial light. Similarly, Li et al. [96] combined the contrast enhancement algorithm with the image de-hazing algorithm to generate images with vivid color and high image contrast.

These priors work in some underwater scenes, however, they do not always succeed due to complex and changing underwater environment. For example, when large white objects or artificial light exist in underwater images, the underwater dark channel prior will fail, and the estimated medium transmission is not accurate.

2.3.3 Deep Learning based Image Enhancement Methods

Over the last decade, deep learning based image enhancement methods have made remarkable progress due to its powerful feature learning ability. Deep convolutional neural networks based models provide state-of-the-art performances for image enhancement tasks, such as image de-blurring and image de-hazing, in both terrestrial and underwater domains. Deep

learning based image enhancement methods [97–100] usually construct deep neural networks and train them using pairs of degraded underwater images and high-quality counterparts. The deep models learn high-level non-linear filters from the paired training data, which provide significantly better performances compared to physical models. Some researchers tried to construct feed-forward network structures for underwater image enhancement. Li et al. [98] first synthesised underwater images from RGB-D in-air images, and then trained a two-stage network for underwater image restoration with the synthetic training data. Fabbri et al. [101] proposed a gradient difference loss (GDL) for their designed feed-forward network to remove the haze effects in the enhanced underwater images. Li et al. [99] employed a gated fusion network architecture to learn three confidence maps used to weight the three input images. A major limitation of the above-mentioned feed-forward networks is that they require paired training data, which may not be available or can be difficult to acquire for many practical applications. Moreover, a simple feed-forward network with a random initialisation is unable to estimate the desired output properly.

More recently, Generative Adversarial Networks (GANs) based models have reported impressive results for image-to-image translation tasks without paired training data. The GAN-based models attempt to improve generalisation performance by employing a two-player min-max game, where a generator tries to fool the discriminator by generating fake images and the discriminator tries to discard the fake images. This forces the generator to learn realistic enhancement while evolving with the discriminator. Among the available GAN models, the two-way GANs such as CycleGAN and DualGAN, can translate an image from one domain to another domain without using paired training data. CycleGAN is the most used and effective two-way GAN for image-to-image translation. The most important component in CycleGAn is the cycle consistency loss which enables the GANs to learn the mutual mappings between two domains from unpaired data. To address the lack of paired training data, Fabbri et al. proposed an underwater color transfer model [101] based on CycleGAN [102] without needing paired training data. Li et al. [103] developed a weak underwater image color correction model based on the cycle-consistent adversarial network (CycleGAN), while a multi-term loss function is also used to preserve detailed content and

structural information. Ye et al. [10] proposed an unsupervised adaptation network to jointly estimate scene depths and correct color from monocular underwater images.

However, GANs also have some disadvantages: (1) the adversarial training can be unstable, in many cases, the discriminator and the generator both cannot eventually reach equilibrium because the discriminator becomes too good; (2) mode collapse frequently occurs when the generator learns to produce samples that only fit a part of the real distribution. For example, we train a GAN to synthesise 100 images, but the generator may translate any input into only one desired image rather than 100 different images. Since the training of GANs is unstable, careful hyper-parameter selection and intuitive loss function adaptation are required to ensure the convergence. Some researchers made improvements on the loss functions. For example, Wasserstein GAN employed the earth-mover distance to measure the distance between the model distribution and the real data distribution, while Energy-based GANs modeled the discriminator as an energy function to improve the training stability.

Previous underwater image enhancement algorithms aim to generate visual pleasing images, however, in practice they usually serve as a preprocessing step for mid- and high-level vision tasks, i.e. improving the performance of the later vision tasks. Hence, it is more attractive to propose image enhancement networks that can generate detection favorable images rather than visually pleasing images.

2.4 Underwater Image Synthesis (UIS)

Due to the lack of training image pairs for training UIE frameworks, some UIS methods have been proposed to synthesise distorted underwater images from high quality in-air RGB or RGB-D images, and construct underwater-in-air image pairs as training data. These methods can be broadly classified into two categories: physical model based and deep learning based methods.

2.4.1 Physical Model based Image Synthesis Methods

Physical model based UIS methods usually use an underwater image formation model to synthesise underwater images. The Jaffe-McGlamery model [104] is the most commonly used underwater image formation model. It assumes that an underwater image is formed with three optical processes: light absorption, light back-scattering and light forward-scattering. First, light absorption indicates the light energy has been absorbed by the water and other suspended particles when the light travels in the water. Light absorption causes the color attenuation. Color channels have different wavelengths that result in different attenuation coefficients [12]. The red light possesses a longer wavelength and it attenuates faster than the light of other colors in the water. Thus, underwater images and videos present bluish or greenish tones. Second, a photon of light travels through the water, subject to light scattering which actively builds the characteristics of haze-effects. Light scattering can be divided into forward- and backward-scattering. Light forward-scattering indicates the process that the light reflected from the object is scattered on its way back to the camera. It results in an effect similar to Gaussian blurring. Light back-scattering occurs when the ambient light is scattered on its way to the camera.

According to the Jaffe-McGlamery model [104], Luczynski et al. [105] simulated the light absorption and light back-scattering processes. They also simulated the forward-scattering process using the Gaussian blurring. In the end, they fused the output of three processes and generated underwater-like images. In [12], Chiang et al. pointed out that light forward-scattering is much less prominent and can be neglected, thereafter, they proposed a simple underwater image formation model only containing the light absorption and back-scattering processes. Following Chiang's underwater image formation model, several works such as [11, 95] synthesised underwater images from RGB-D in-air images. Takumi et al. [106] took the water type as a consideration factor when synthesising underwater images, because the water types result in different attenuation coefficients, i.e., different synthesised images. They synthesised underwater image datasets using 10 Jerlov water types [107]. Li et al. [11] synthesised underwater image degradation datasets which cover a diverse set of degradation levels and water types. To synthesise more realistic underwater images, Peng

et al. [95] improved the image formation model by incorporating a point spread function, which can simulate image blur.

However, the physical model based UIS methods have a major limitation that they require the depth data of the in-air images, which is a necessary parameter in the underwater image formation model. But, the depth data may be not available for many datasets.

2.4.2 Deep Learning based Image Synthesis Methods

Recently, Generative Adversarial Networks (GANs) [98, 101, 108, 105] have been investigated in the underwater image synthesis field due to its successes in image-to-image translation tasks. Li et al. [98] treated the UIS task as an image-to-image translation task and exploited a single GAN to synthesise underwater images from in-air RGB-D images. Their generator model can be broken down into three stages: (1) light absorption, which simulates the light absorption process by referring to optical priors; (2) light scattering, which simulates the light scattering process using a convolutional neural network without referring to optical priors; (3) Vignetting, which produces a shading effect on image corners, is caused by certain camera lenses. However, their weakly-supervised synthesis model have to be trained on unpaired images, making it difficult to simulate image details such as colors and textures. To alleviate the needs for training image pairs, Fabbri et al. [101] applied a two-way CycleGAN [102], which allows the learning of mutual translation between in-air and underwater images. The CycleGAN includes two generators, where one generator translates in-air images to underwater ones and can be regarded as the synthesis model, whilst the other one translates underwater images into in-air images which can be regarded as the enhancement model. Gupta et al [108] proposed a single-stage network with two generators to simultaneously synthesise underwater images and estimate underwater depth maps.

However, both physical model and deep learning based UIS methods cannot accurately model the degradation progress of underwater imaging, and often result in unsatisfactory synthetic images [9, 109]. Current physical model based UIS approaches [11, 95] only synthesise the scenes of 10 Jerlov water types and considers only two factors in the degradation progress, leading to significant errors in the generated images. Moreover, deep learning

based methods [110, 111] may encounter the model collapse problem that generates images with monotonous colors and frequent artifacts. Their capability to modelling haze-effects is also limited. Different from the previous works, in this thesis, we first improve the physical image formation model [12], and leverage both physical priors and data-driven cues to a hybrid synthesis model to create more realistic underwater images.

2.5 Summary

In this chapter, we focus on reviewing machine learning approaches for underwater object detection (UOD), underwater image enhancement (UIE) and underwater image synthesis (UIS) considering their powerful feature learning abilities. UIS approaches generated additional data for training machine learning based UIE frameworks (in particular deep learning methods), while UIE approaches assisted UOD to achieve higher detection precision by improving the quality of noisy underwater images. For each of three tasks, we classified the related works into several categories, and discussed the advantages and disadvantages of each category. Moreover, to achieve robust detection in the underwater scenes, several challenges such as the noisy data and class imbalance problems should be addressed. We have identified these challenges that hinder the development of underwater object detection, which will be the major discussion aspects in this thesis.

Chapter 3

Underwater Image Enhancement with Deep Learning and Physical Priors

3.1 Introduction

Underwater object detection (UOD) is of great importance for underwater applications such as ocean exploring and monitoring and autonomous underwater vehicles [9]. However, underwater images acquired in complicated environments suffer from severe distortion which dramatically degrades image visibility and affects the detection accuracy of UOD tasks.

In recent years, underwater image enhancement (UIE) technologies [9, 97], especially deep learning based approaches, work as a pre-precessing operation to boost the detection accuracy of UOD tasks by improving the visual quality of underwater images. However, most of the existing strategies consider UIE and UOD tasks as two separate pipelines, whereas the UIE task is evaluated on the visual quality of images while the UOD task is evaluated on the detection accuracy. Separate optimisation of the two tasks results in inconsistency in the pursuit of image quality and detection accuracy: These two tasks have different optimisation objectives, leading to different optimal solutions. Moreover, current top-performing deep learning based UIE methods, e.g. [98, 101], are normally trained on synthetic images due to the lack of large training data (i.e., pairs of degraded underwater images and high-quality counterparts). The enhancement models trained on synthetic images

cannot always be generalised to underwater scenes because the quality of synthetic images cannot be guaranteed by the existing image synthesis (UIS) methods.

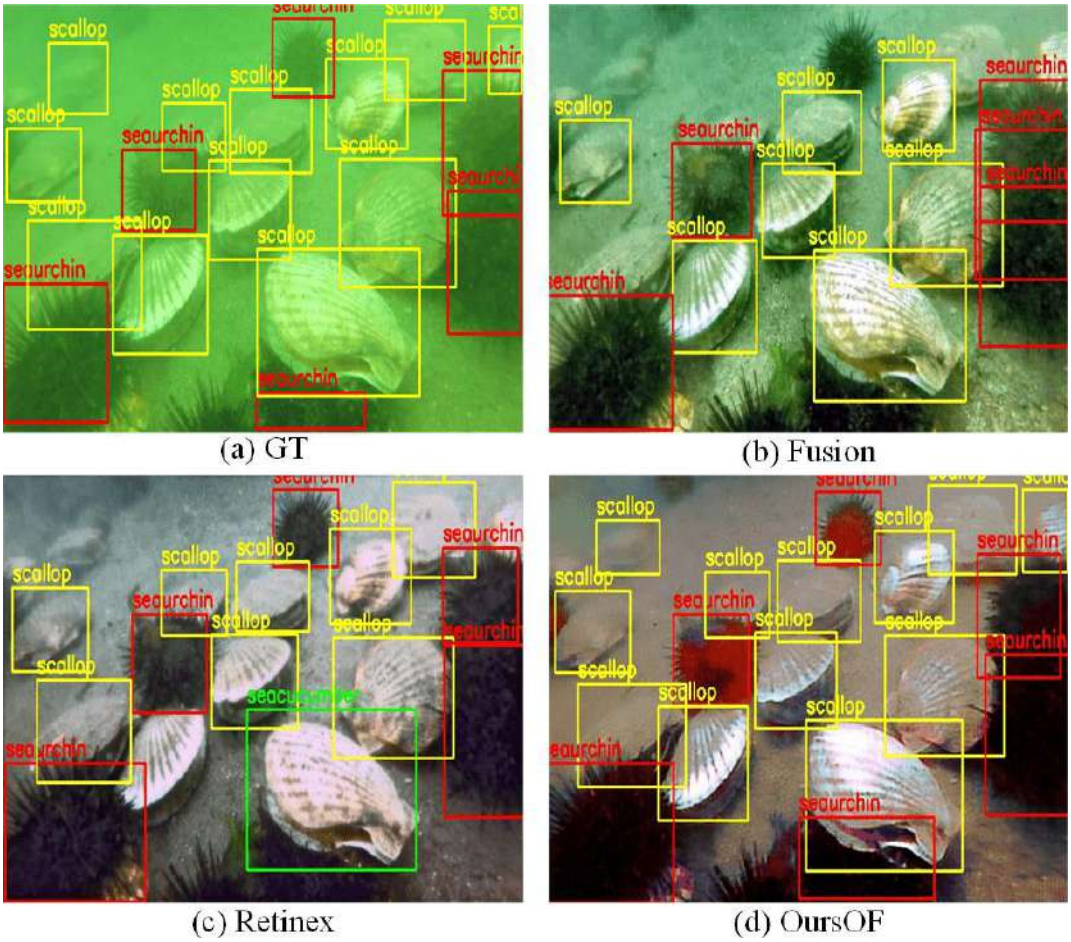

Fig. 3.1 Object detection results of the Single Shot MultiBox Detector [112] after we have applied different UIE algorithms, including (b) Fusion [75], (c) Retinex [76], and (d) Our proposed object-focused perceptual enhancement model. (a) Raw underwater image with ground-truth annotations.

To address these two concerns, we firstly propose a hybrid underwater image synthesis model to synthesize realistic training data using data-driven cues and physical priors. This enables our enhancement model to be properly generalised on real-world underwater scenes. Secondly, we propose two detection-perceptual enhancement models, each of which consists of an enhancement model and a detection perceptor. The detection perceptor is well-trained on high-quality in-air images and encodes fine object details and potential detection favouring information of high-quality in-air images. Two perceptual losses are designed to transfer

the knowledge encoded in the detection perceptors to the enhancement model in the form of gradients (as inference of updating directions). One of the detection perceptors is named patch detection perceptor with a patch perceptual loss that guides the enhancement model to generate patch level visually pleasing images. The other one is named object-focused detection perceptor with an object-focused perceptual loss which guides the enhancement model to generate detection-favouring images. Fig. 3.1 shows the object detection results of the same Single Shot MultiBox Detector (SSD) [112] trained on the enhanced results of different UIE algorithms. The deep detectors trained on the enhanced results of two representative UIE algorithms, i.e., Fusion [75] and Retinex [76], often miss detecting "noisy" objects or predict incorrect object categories, while our object-focused perceptual enhancement model (denotes as OursOF) can largely improve the detection accuracy of the standard deep detector. The gradients from the object-focused perceptual loss highlight the region containing the object and produce clear reddish color around the object in Fig. 3.1(d). The proposed synthesis and the perceptual enhancement models are integrated into a unified framework named HybridDetectionGAN that can work in an end-to-end style.

3.2 Proposed Underwater Image Enhancement Model

In this section, we introduce our proposed perceptual enhancement framework named Hybrid-DetectionGAN. We have designed two novel components in HybridDetectionGAN, including a novel hybrid synthesis model and a novel perceptual enhancement model. We first present the overview of HybridDetectionGAN. Then, we describe the novel hybrid synthesis and perceptual enhancement models. Finally, we introduce the training of the framework.

3.2.1 The Overview of the Proposed Framework

As shown in Fig. 3.2, our HybridDetectionGAN framework exploits two cycle-consistency paths, which can learn the transformation between underwater and in-air domains from the unpaired images. Particularly, the forward cycle-consistency path starts with real in-air RGB-D images and finishes with reconstructed in-air images (i.e., enhanced underwater images

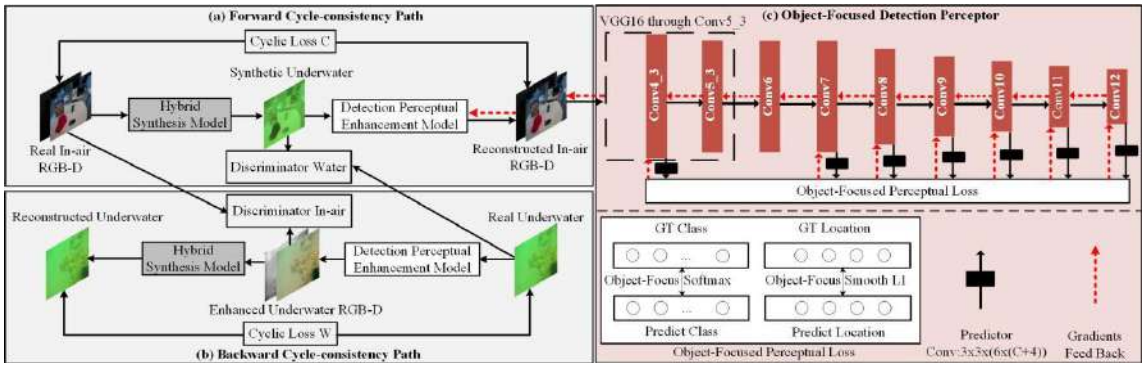

Fig. 3.2 The overview of our HybridDetectionGAN. It consists of two cycle-consistency paths (a) and (b) to learn the transformation between underwater and in-air domains from unpaired images. An object-focused detection perceptor (c) guides the enhancement model to generate detection favourable images.

and depth images). The hybrid synthesis model transforms in-air RGB-D images into their underwater counterparts, and the enhancement model transforms underwater images into in-air images. The cyclic/cycle-consistency loss regularises both the synthesis and enhancement models to generate better structural content in the images. Following the enhancement model, a novel object-focused detection perceptor is activated, which is trained with the detection loss on the real in-air images. The enhancement model and the detection perceptor construct the complete detection-perceptual enhancement model. During the training of the enhancement model, detection-favouring information (red dashed arrows in Fig. 3.2) is given to the enhancement model in the form of gradients via the object-focused perceptual loss, which is consisted of an object-focused Softmax loss and an object-focused SmoothL1 loss. The backward cycle-consistency path starts with real underwater images and finishes with the reconstructed underwater ones. The adversarial process between the enhancement model and the discriminator helps produce realistic in-air images.

3.2.2 Hybrid Underwater Image Synthesis Model

The generalisation of the enhancement model highly relies on the quality of the synthetic training data. To develop a robust synthesis model, we incorporate an improved physical model and a data-driven CNN model into a hybrid synthesis model. The improved physical

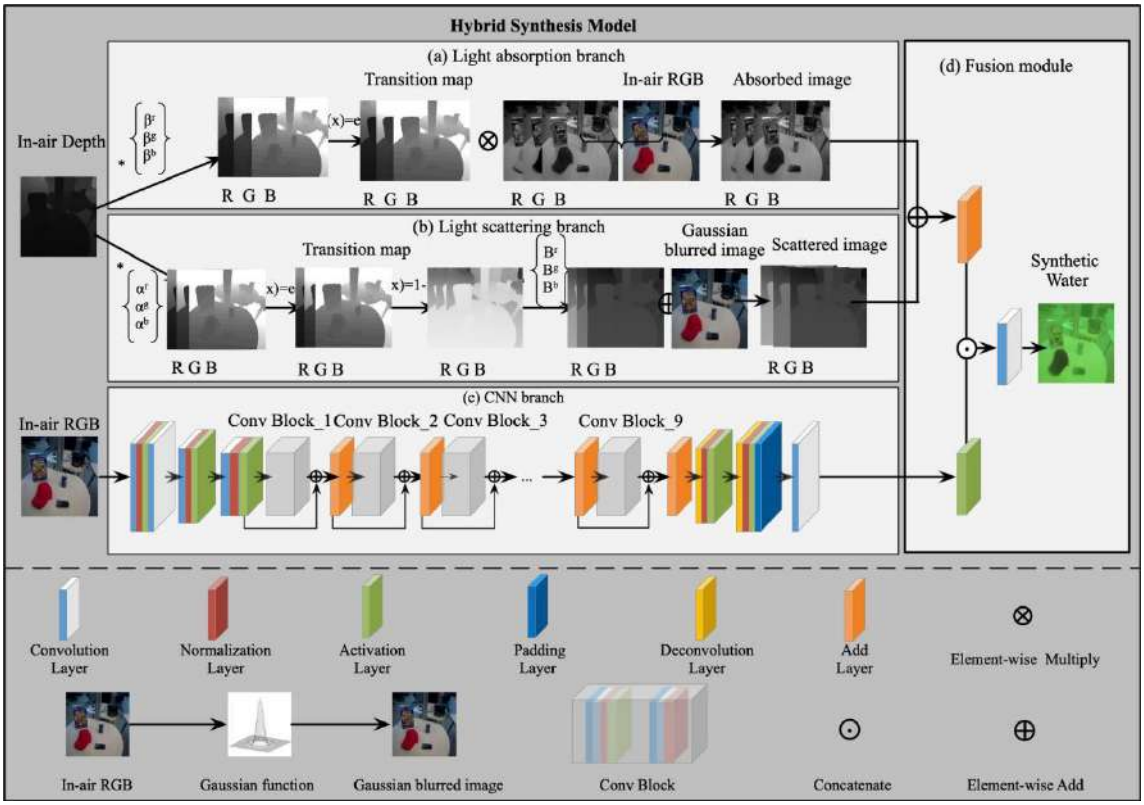

Fig. 3.3 The overview of the proposed hybrid synthesis model, which consists of (a) light absorption branch, (b) light scattering branch and (c) CNN branch. The outputs of three branches are combined into the final synthetic underwater image shown in (d) fusion module.

model is able to simulate evident haze-effects and maintains coarse views of underwater images by applying the priors of light absorption and scattering. The CNN model works as a supplement to generate finer details of underwater images by modelling other factors. Fig. 3.3 shows the overall structure of the hybrid synthesis model. It consists of three branches, i.e., light absorption, light scattering, and CNN branches. The light absorption and scattering branches are built on optical priors, and construct the complete physical model.

3.2.2.1 Improved Underwater Image Formation Model

Our underwater image formation model is based on the physical model proposed in [12], and formulated as Eq. (3.1).

$$I_{sw}^{\lambda}(x,y) = \sum_{w=0}^{W}\sum_{h=0}^{H}\sum_{m=0}^{M} I_{con}(x+w, y+h, m)\vartheta_f^{\lambda}(w,h,m), \quad \lambda \in \{r,g,b\} \tag{3.1}$$

where

$$I_{con} = I_{ab} + I_{sc} \odot I_{cnn} \tag{3.2}$$

$I_{sw}(x,y)$ denotes the pixel value of the synthetic underwater image at each point (x,y). λ denotes different channels of an image, including red, green, and blue channels. ϑ_f^{λ} is a $W \times H \times M$ convolutional filter, responsible for converting the outputs of the three branches into the λ-channel of the synthetic underwater image. I_{ab}, I_{sc} and I_{cnn} are the output of the light absorption branch, the light scattering branch and the CNN branch, respectively. I_{con} is the fused output of three branches and formulated as Eq. (3.2), where $+$ and $\odot$ denote the element-wise add operation and channel-wise concatenation operation respectively. The final synthetic underwater image I_{sw} is achieved through the convolution operation between ϑ_f and I_{con} in Eq. (3.1).

Light absorption causes the change of color tones, and the image suffering from light absorption can be described by Eq. (3.3). Different channels of a RGB image have different absorption coefficient β. The effect of light absorption becomes stronger with increasing object-camera distance d as more energy is absorbed by water. For each pixel on the image, its depth d comes from depth map I_d. To retain the absorption image, we first compute the transition map $T = e^{-I_d\beta^{\lambda}}$, indicating part of the light has been absorbed during the propagation in the water. Then, we compute each channel of the absorption image I^{λ}_{ab} using element-wise multiplication operation $\otimes$ between in-air image I_a and T.

$$I_{ab}^{\lambda} = I_a^{\lambda} \otimes e^{-I_d\beta^{\lambda}}, \lambda \in \{r,g,b\} \tag{3.3}$$

Light scattering is an optical process in underwater imaging, including forward and back scattering [44]. Forward scattering occurs when the light reflected from the object is scattered on its way to the camera, resulting in an effect very similar to Gaussian blurring. However, the commonly used physical model only considers back scattering priors while ignoring forward scattering priors. Hence, we simulate the haze-effects caused by forward scatters using a Gaussian blur function. The scattered image I_{sc} is formulated as

$$I^{\lambda}_{sc} = I^{\lambda}_{bsc} + I^{\lambda}_{fsc}, \lambda \in \{r, g, b\} \tag{3.4}$$

where

$$I^{\lambda}_{bsc} = B^{\lambda}(1 - e^{-I_d \alpha^{\lambda}}), \lambda \in \{r, g, b\} \tag{3.5}$$

$$I_{fsc} = I_a \Phi(x, y) \tag{3.6}$$

$$\Phi(x, y) = Ae^{\frac{-x^2+y^2}{q^2}} \tag{3.7}$$

where A is determined by

$$\int\int A e^{\frac{-x^2+y^2}{q^2}} dxdy = 1 \tag{3.8}$$

where I_{bsc} and I_{fsc} are the images suffering from backward and forward scattering, respectively. B and α denote the ambient light and the backscatter coefficient. $\Phi(x, y)$ denotes the Gaussian function that adds the Gaussian blurring effect onto the in-air image I_a.

Different from the previous physical model based synthesis methods, which use predefined parameters to synthesize the scenes of 10 Jerlov water types, the parameters of our proposed underwater image formation model can be learnt using gradient-based optimisation algorithms and better simulate the characteristics of the targeted underwater images. Moreover, a CNN branch works as a supplement of the physical model to simulate more degraded characteristics of the underwater images. The light absorption and scattering priors of the physical model can be used to simulate coarse color distortion and haze-effects. However, significant errors exist in the resultant images due to image quality degrading. For instance,

the existence of artificial lights leads to non-uniform illumination on the images, and the movement of the cameras bring in noise in the captured images. All of these factors have not been modelled in the physical model, and the CNN branch helps us to simulate these factors and generates finer color tones, illumination change and noise. The detailed structure of the CNN branch can be found in Fig. 3.3.

3.2.3 Detection Perceptual Enhancement Model

The perceptual loss has been widely used to improve the quality of images or generate images of interests. A number of recent papers have employed perceptual loss to improve image quality. One of the means is to use feature reconstruction-based perceptual loss [113–116], which has achieved large successes in a wide variety of image translation tasks, such as transfer learning, image super resolution and synthesis. These works first extract convolutional features from pre-trained CNNs, and then design a perceptual loss function for other image transformation networks, which minimise the discrepancy between the extracted CNN features of the generated and target images, which guides the transformation network to generate high-quality images similar to the target images. The other means is to use classification-based perceptual loss [117–120], which generates images by computing and returning the gradients of a given class with respect to the input image in a learnt classification network. For example, given a learnt classification network on the large-scale ImageNet dataset and a class of interests, Karen et al. [117] supplied a zero image to the well-trained classification network, then computed the gradients of the given class with respect to the zero input image, and used the gradients to update the model parameters, and finally a newer version of the input image is generated. Since the newer image is generated by maximising the score of the given class, some discriminative attributes of the interested object class help one to recognise the objects of the interested class. Similar optimisation techniques can also be used to generate high-confidence fake images [119, 120]. The success of perceptual loss based strategies lies in that a high-capacity neural network trained beforehand could implicitly learn to encode relevant image details and semantics. The use of perceptual loss functions allows the transfer of knowledge from the high-capacity neural networks to the

transformation networks or the input. However, all these perceptual losses aim to generate high quality images without explicitly considering the requirements of the following tasks. The high capacity networks are trained on image-level labels with a classification loss without detailed object information. In this work, we propose two detection perceptual losses, guiding the enhancement network to generate visually pleasing or detection-favouring images.

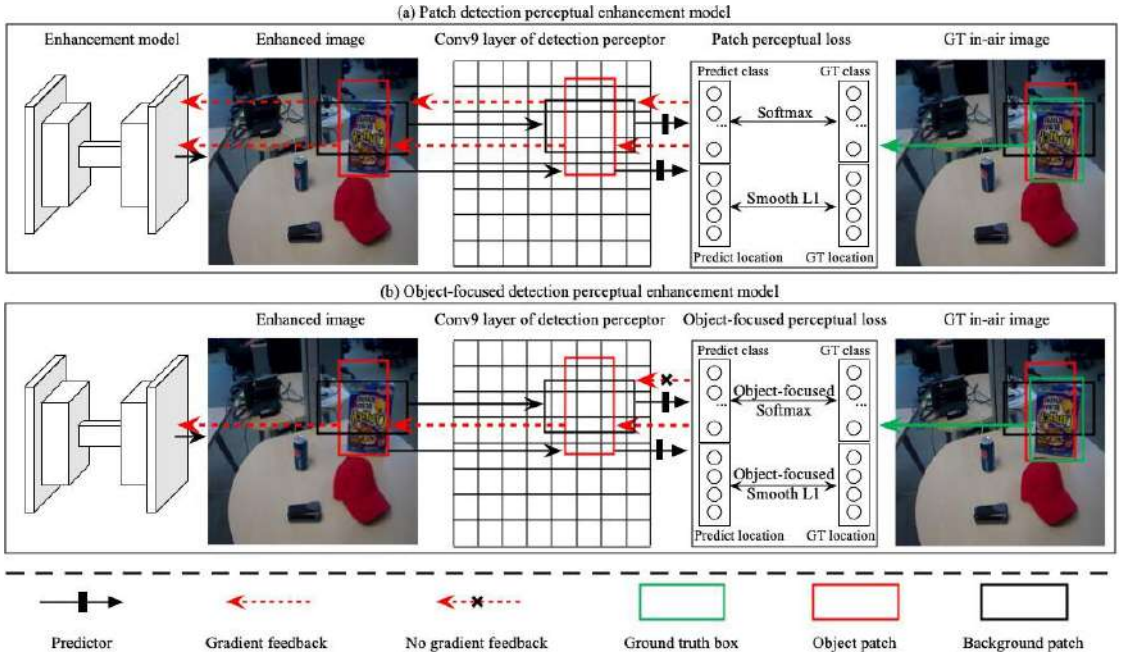

Fig. 3.4 The framework of (a) patch detection perceptual enhancement model and (b) object-focused detection perceptual enhancement model. The perceptual loss refers to the discrepancy between the patches of the enhanced image and the ground truth in-air image and we feeds this discrepancy back to update the enhancement model in the form of gradients. The patch detection perceptual enhancement model supplies the information of the background and object patches, while the object-focused detection perceptual enhancement model only provides the object information.

The success of perceptual loss in many tasks demonstrates that a high-capacity CNN properly trained has the capability of implicitly encoding fine image details and task-related semantic knowledge. Inspired by these observations, we propose the patch detection perceptual enhancement model and object-focused detection perceptual enhancement model. The framework of patch detection perceptual enhancement model and object-focused detection perceptual enhancement model can be found in Fig. 3.4. We first train a one-stage deep detector [112] on high quality in-air images, which encodes object details and potential detec-

tion favouring images. Then, we add this deep detector as a detection perceptor (its weights are kept fixed to those obtained from training) to provide detection-favouring perceptual information to the enhancement model. During the adversarial training, the enhanced image goes straight to the detection perceptor. The detection perceptor is an one-stage deep detector [112], which first associates 6 default patches of different scales and aspect ratios at each of the seven convolutional layers (we only show 2 default patches on the 9-th convolutional layer in Fig. 3.4 for simplicity). Then, the predictor works with a $C + 1$-dimension class vector and a 4-dimension location vector for each default patch using a 3×3 convolutional filter. C denotes the number of the object classes and 1 denotes the background class. Next, it assigns the ground-truth class and the location for each default patch using the following matching rules: If the Interaction over Union (IoU) between a default patch and its overlapped ground-truth object is larger than 0.5, the class and the location of the object are assigned as those of the default patch. If the default patch does not match any ground-truth object with an IoU larger than 0.5, it is labelled as the background patch and has no ground-truth location. For example, the default patch (red) shown in Fig. 3.4 matches the cereal box (green box) with the IoU of larger than 0.5, so it is an object patch and its ground-truth class and location are labelled as those of the cereal box. The black patch does not match any object, so it is labelled as the background patch and has no target location. Finally, a detection based perceptual loss indicates the discrepancy between the patches of the enhanced image and those of the high quality in-air image, and feeds back the discrepancy to the enhancement model in the form of gradients, based on the enhancement model of continuously updating its parameters. Until no gradient has any impact, when the enhanced images are the same as the in-air images in the detection perceptor space, the object details and detection favouring information of the in-air image encoded in the detection perceptor space have been properly transformed via the enhancement model.

We design two perceptual loss functions having different objectives for two detection perceptors. The first loss is named patch detection perceptual loss that aims to generate patch-level in-air images. The second one is named object-focused detection perceptual loss which aims to generate detection-favouring images that can improve the detection accuracy.

Fig. 3.4 shows the overall structure of the patch detection perceptual enhancement and the object-focused detection perceptual enhancement models, each of which consists of an enhancement model and a detection perceptor with a specially designed perceptual loss.

3.2.3.1 Patch Detection based Perceptual Loss

Patch detection perceptual loss function L_p is an one-stage detection loss [112], which is a weighted sum of classification loss L_{cls} and localization loss L_{loc}.

$$L_p = \frac{1}{N} \sum_{i \in all} L_{cls}(pcls^i, gcls^i) + \frac{1}{\bar{N}} \sum_{i \notin bg} L_{loc}(ploc^i, gloc^i) \tag{3.9}$$

where $pcls^i$ and $gcls^i$ denote the predicted and ground-truth class vectors of $C + 1$ dimensions for the i-th default patch. $ploc^i$ and $gloc^i$ denote the predicted and ground-truth location vector of 4-dimensions for the i-th default patch. The 4-dimension location vector includes the coordinates of center (cx, cy) with width w and height h. all and bg are the set of all the default patches and the patches belonging to the background samples, without any contribution to the location loss because of absent ground-truth location. N and $\bar{N}$ are the numbers of all the default and the object patches. Specially, the classification loss L_{cls} is a softmax loss.

$$L_{cls}(pcls, gcls) = -\sum_{c=1}^{C+1} pcls_c log(gcls_c) \tag{3.10}$$

where pre_cls_c and gt_cls_c indicate the c-th element of the predicted and the ground-truth class vectors, respectively. The localisation loss is a smooth L1 loss [112] between the predicted and the ground-truth locations.

$$L_{loc}(ploc, gloc) = \sum_{l=1}^{4} smooth_{L1}(ploc_l - gloc_l) \tag{3.11}$$

L_{cls} encourages the enhancement model to generate images which minimise the class discrepancy between the generated and the ground-truth patches, and L_{loc} encourages the enhancement model to generate the images of minimising the location discrepancy between

the generated and the ground-truth patches, thus the patch perceptual detection loss can guide the enhancement model towards generating more realistic patches at accurate locations.

3.2.3.2 Object-focused Detection based Perception Loss

For underwater object detection applications, many objects look very similar to the background. The complex background may degrade the detection accuracy of the detectors. To deal with this challenge, we propose the object-focused detection based perceptual loss L_{of}:

$$L_{of} = \frac{1}{\bar{N}} \sum_{i \notin bg} L_{cls}(pcls^i, gcls^i) + L_{loc}(ploc^i, gloc^i) \tag{3.12}$$

Different from the patch detection perceptual loss, the object-focused detection perceptual loss only focuses on feeding back the informations of object patches, while ignoring the background patches, as shown in Fig. 3.4 (the black cross indicates background patches have no information feedback to the enhancement model). From the optimisation perspective, L_{of} is designed to assign ground-truth classes and locations of the object patches on the enhanced image, for achieving detection accuracy of the deep detector trained on the enhanced images.

3.2.4 Training of Our Overall HybridDetectionGAN

We first train the standard one-stage deep detector on high quality in-air images. Afterwards, we add the detector after the enhancement model in the forward cycle-consistency path. We then move on to train the synthesis and enhancement models.

3.2.4.1 Training of the Hybrid Synthesis Model

Denoting $G_{\vartheta_{a2w}}$ as the hybrid synthesis model parameterised by ϑ_{a2w} and ϑ_{cnn} as the parameters of the CNN branch, then we have $\vartheta_{a2w} = \{\alpha, \beta, B, \vartheta_{cnn}, \vartheta_f\}$. Denote $G_{\vartheta_{w2a}}$ as the enhancement model parameterised by ϑ_{w2a}. We obtain ϑ_{a2w} by minimising the loss function L_{a2w}, which is a combination of an adversarial loss L_{adv_w} and a cycle-consistency loss L_{cyc_w}.

$$L_{a2w} = w_1 L_{adv\_w} + w_2 L_{cyc\_w} \tag{3.13}$$

The first term is an adversarial loss produced by the discriminator, which is denoted as $D_{\vartheta_{dw}}$ and parameterised by ϑ_{dw}. Taking the synthetic underwater image $G_{\vartheta_{a2w}}(I_a, I_d)$ as the input, the discriminator outputs the estimated probability of the synthetic underwater image treated as a "real" underwater image, denoted as $D_{\vartheta_{dw}}(G_{\vartheta_{a2w}}(I_a, I_d))$. By fooling the discriminator with the synthetic underwater image, the adversarial loss is formulated as $L_{adv\_w} = -logD_{\vartheta_{dw}}(G_{\vartheta_{a2w}}(I_a, I_d))$, which encourages the hybrid synthesis model to produce more realistic underwater images. The cycle-consistency loss L_{cyc_w} is computed as the L_1 distance between the reconstructed and ground-truth underwater image I_w, i.e., $L_{cyc\_w} = \|G_{\vartheta_{a2w}}(G_{\vartheta_{w2a}}(I_w)) - I_w\|_1$.

Different from the physical-model based synthesis model, our physical parameters α, β and B can better simulate the characteristics of the target underwater images, which are learnt from the training data via the gradient descent optimisation algorithm. The optimisation algorithm iteratively updates the parameter β^λ by

$$\beta^\lambda = \beta^\lambda - \eta \frac{\partial L_{a2w}}{\partial \beta^\lambda} \tag{3.14}$$

where η is the learning rate. In order to update β^λ, we need to compute $\frac{\partial L_{a2w}}{\partial \beta}$, which indicates the gradient of L_{a2w} with respect to β^λ. Denote I^λ_{add} as the output of the physical branch, $I^\lambda_{add} = I^\lambda_{ab} + I^\lambda_{sc}$. I^λ_{ab}, I_{con} and I_{sw} are presented in Eqs. (3.1)-(3.3). We can derive $\frac{\partial L_{a2w}}{\partial \beta^\lambda}$ using the chain rule:

$$\frac{\partial L_{a2w}}{\partial \beta^\lambda} = \frac{\partial L_{a2w}}{\partial L_{adv\_w}} \frac{\partial L_{adv\_w}}{\partial \beta^\lambda} + \frac{\partial L_{a2w}}{\partial L_{cyc\_w}} \frac{\partial L_{cyc\_w}}{\partial \beta^\lambda} = w_1 \frac{\partial L_{adv\_w}}{\partial \beta^\lambda} + w_2 \frac{\partial L_{cyc\_w}}{\partial \beta^\lambda} \tag{3.15}$$

where $\frac{\partial Ladv\_w}{\partial \beta^\lambda}$ and $\frac{\partial Lcyc\_w}{\partial \beta^\lambda}$ can be derived using the chain rule as follows:

$$\frac{\partial L_{adv\_w}}{\partial \beta^\lambda} = \frac{\partial L_{adv\_w}}{\partial I_{sw}} \frac{\partial I_{sw}}{\partial I_{con}} \frac{\partial I_{con}}{\partial I^\lambda_{add}} \frac{\partial I^\lambda_{add}}{\partial I^\lambda_{ab}} \frac{\partial I^\lambda_{ab}}{\partial \beta^\lambda} = -I_a \otimes e^{-I_d \beta^\lambda} I_d \frac{\partial L_{adv\_w}}{\partial I_{sw}} \frac{\partial I_{sw}}{\partial I_{con}} \frac{\partial I_{con}}{\partial I^\lambda_{add}} \tag{3.16}$$

$$\frac{\partial L_{cyc\_w}}{\partial \beta^\lambda} = \frac{\partial L_{cyc\_w}}{\partial I_{sw}} \frac{\partial I_{sw}}{\partial I_{con}} \frac{\partial I_{con}}{\partial I^\lambda_{add}} \frac{\partial I^\lambda_{add}}{\partial I^\lambda_{ab}} \frac{\partial I^\lambda_{ab}}{\partial \beta^\lambda} = -I_a \otimes e^{-I_d \beta^\lambda} I_d \frac{\partial L_{cyc\_w}}{\partial I_{sw}} \frac{\partial I_{sw}}{\partial I_{con}} \frac{\partial I_{con}}{\partial I^\lambda_{add}} \tag{3.17}$$

Combining Eqs. (3.15)-(3.17) with Eq. (3.14), we have:

$$\beta^{\lambda} = \beta^{\lambda} + \eta\left(w_1 \frac{\partial L_{adv\_w}}{\partial I_{sw}} + w_2 \frac{\partial L_{cyc\_w}}{\partial I_{sw}}\right)\left(I_a^{\lambda} \otimes e^{-I_d \beta^{\lambda}} I_d \frac{\partial I_{sw}}{\partial I_{con}} \frac{\partial I_{con}}{\partial I_{add}^{\lambda}}\right) \tag{3.18}$$

Similarly, we update α and B through the chain rule, and have the final parameters when the models converge.

$$\alpha^{\lambda} = \alpha^{\lambda} - \eta\left(w_1 \frac{\partial L_{adv\_w}}{\partial I_{sw}} + w_2 \frac{\partial L_{cyc\_w}}{\partial I_{sw}}\right)\left(B^{\lambda} e^{-I_d \alpha^{\lambda}} I_d \frac{\partial I_{sw}}{\partial I_{con}} \frac{\partial I_{con}}{\partial I_{add}^{\lambda}}\right) \tag{3.19}$$

$$B^{\lambda} = B^{\lambda} - \eta\left(w_1 \frac{\partial L_{adv\_w}}{\partial I_{sw}} + w_2 \frac{\partial L_{cyc\_w}}{\partial I_{sw}}\right)\left((1 - e^{-I_d \alpha^{\lambda}}) \frac{\partial I_{sw}}{\partial I_{con}} \frac{\partial I_{con}}{\partial I_{add}^{\lambda}}\right) \tag{3.20}$$

We obtain ϑ_{dw} by optimising the loss function L_{d_w}, which encourages the discriminator to identify the difference between the synthetic and real underwater images:

$$L_{d\_w} = -\log D_{\vartheta_{dw}}(I_w) - \log\left(1 - D_{\vartheta_{dw}}(G_{\vartheta_{a2w}}(I_a, I_d))\right) \tag{3.21}$$

3.2.4.2 Training of the Detection Perceptual Enhancement Model

We obtain ϑ_{w2a} by optimising the loss function L_{w2a}, which is a weighted combination of adversarial loss L_{adv_a}, cycle-consistency loss L_{cyc_a}, and perceptual loss L_{of} (or L_{patch}).

$$L_{w2a} = w_1 L_{adv\_a} + w_2 L_{cyc\_a} + w_3 L_{of} \tag{3.22}$$

Denote $D_{\vartheta_{da}}$ as the discriminator parameterised by ϑ_{da}. Taking the enhanced RGB-D underwater image $G_{\vartheta_{w2a}}(I_w)$ as input, the discriminator outputs the estimated probability of the enhanced image as a real in-air image, denotes as $D_{\vartheta_{da}}(G_{\vartheta_{w2a}}(I_w))$. To fool the discriminator with the enhanced underwater image, the adversarial loss L_{adv_a} is formulated as $L_{adv\_a} = -\log D_{\vartheta_{da}}(G_{\vartheta_{w2a}}(I_w))$. L_{cyc_a} is computed as the L_1 distance between the reconstructed and ground-truth in-air images, $L_{cyc\_a} = \|G_{\vartheta_{w2a}}(G_{\vartheta_{a2w}}(I_a)) - I_a\|_1$. L_{of} is the object-focused perceptual loss that encourages the enhancement model to generate detection-favouring outcomes.

We obtain ϑ_{da} by optimising L_{d_a}, which encourages the discriminator to address the difference between the enhanced underwater images and the real in-air images.

$$L_{d\_a} = -logD_{\vartheta_{da}}(I_a) - log(1 - D_{\vartheta_{da}}(G_{\vartheta_{w2a}}(I_w))) \tag{3.23}$$

3.2.4.3 How the Detection Perceptor Influence the Enhancement Model in the Form of Gradients

During the training of the enhancement model, the optimisation algorithm iteratively updates the enhancement model's parameter ϑ_{w2a} by

$$\vartheta_{w2a} = \vartheta_{w2a} - \eta \frac{\partial L_{w2a}}{\partial \vartheta_{w2a}} = \vartheta_{w2a} - \eta\left(w_1 \frac{\partial L_{adv\_a}}{\partial \vartheta_{w2a}} + w_2 \frac{\partial L_{cyc\_a}}{\partial \vartheta_{w2a}} + w_3 \frac{\partial L_{of}}{\partial \vartheta_{w2a}}\right) \tag{3.24}$$

$$\frac{\partial L_{of}}{\partial \vartheta_{w2a}} = \frac{1}{\bar{N}} \sum_{\substack{i \neq b \\ g}} \frac{\partial L_{cls}(pcls^i, gcls^i)}{\partial \vartheta_{w2a}} + \frac{\partial L_{loc}(ploc^i, gloc^i)}{\partial \vartheta_{w2a}} \tag{3.25}$$

In each iteration, the detection perceptor feeds the gradients $\eta w_3 \frac{\partial Lof}{\partial \vartheta_{w2a}}$ back to the enhancement model. From Eqs. (3.24) and (3.25), we can see that the enhancement model continuously updates its parameter ϑ_{w2a} to minimise $L_{cls}(pcls^i, gcls^i)$ and $L_{loc}(ploc^i, gloc^i)$, equivalently maximising the class and location prediction accuracy of the object patches. Thus the gradients $\eta w_3 \frac{\partial Lof}{\partial \vartheta_{w2a}}$ help the enhancement model to generate the images with accurate object detection in the following process.

3.3 Experimental Setup

To demonstrate the effectiveness of the proposed method, we conduct comprehensive evaluations on both the unpaired ChinaMM-MultiView and paired OUC datasets. In this section, we first introduce the experimental datasets and the evaluation metrics. Then, we describe the implementation details.

3.3.1 Datasets

The unpaired ChinaMM-MultiView dataset is constructed by collecting images from an underwater image dataset ChinaMM [9] and an in-air image dataset MultiView [121]. ChinaMM is a public competition dataset for evaluating UIE algorithms. The owner of the dataset has publicly released the train set of 2,071 images and the validation set of 676 images. This dataset provides bounding box annotations and contains three object categories: seacucumber, seaurchin and scollap. The resolution of each image is 720x405 pixels. The in-air dataset Multiview consists of 14,179 training images and 1,206 testing images which are captured in the in-door scenes with high quality. This dataset provides RGB images (640 ×480 pixels), depth images and bounding box annotations. It contains five object categories: bowl, cap, coffee mug, cereal box and soda can. To construct the unpaired ChinaMM-MultiView dataset, we randomly choose 2,071 images as the training set and 676 images as the testing set of MultiView.

The paired OUC dataset [122] provides underwater images, high quality reference images and bounding box annotations. The training set contains 2,500 image pairs where the testing set contains 1,198 image pairs. The dataset does not provide depth images which are needed by our hybrid synthesis model, so we apply the technology reported in [108] to obtaining depth maps for all the reference images.

The unpaired Berman-MultiView dataset is constructed by collecting images from the well-known underwater dataset of Berman et al [123] and the in-air image dataset MultiView [121]. The Berman dataset [123] provides 114 high resolution TIF images (5,474x3,653 pixels), raw images, camera calibration files, and the reconstructed scene distance maps that can be downloaded from the project's webpage: http://csms.haifa.ac.il/profiles/tTreibitz/datasets/ambient_forwardlooking/index.html.

3.3.2 Evaluation Metrics

We conduct extensive experiments to quantitatively and qualitatively evaluate the proposed hybrid synthesis and detection perceptual models. For the qualitative evaluations, we directly

present the resultant images. For the quantitative evaluations, we apply several commonly used full-reference image quality evaluation metrics, where ground-truth or references are available. The full-reference metrics include the widely used Mean Square Error (MSE), Structural Similarity (SSIM) and Peak Signal-to-Noise Ratio (PSNR), and Patch-based Contrast Quality Index (PCQI) [124]. Among these four full-reference metrics, MSE measures the image similiarity by calculating the mean square error between each pixels for the two images. Hence, smaller MSE scores indicate higher image quality while higher SSIM, PSNR and PCQI scores indicate higher image. However, both MSE and PSNR have the same limitations that they are poorly correlate with human perception of visual system. Differently, SSIM and PCQI can effectively measure the perceptual similarity between two images.

For the experiments with no reference image, we apply two non-reference image quality evaluation metrics, including underwater image quality measure (UIQM) [125] and underwater color image quality evaluation (UCIQE) [126]. UIQM is a linear combination of three components, i.e., Underwater Image Colorfulness Measure (UICM), Underwater Image Sharpness Measure (UISM), and Underwater Image Contrast Measure (UIConM). Speically, higher UCIQE and UIQM scores indicate higher underwater image enhancement performance. In the experiments, we also reveal the values of these three components for detailed discussion. In addition to the image quality evaluation metrics, we train the deep detectors using the enhanced images by different UIE algorithms and use the mean Average Precision (mAP) as a detection task-specific evaluation metric to evaluate different UIE algorithms.

3.3.3 Implementation Details

All the experiments are conducted on a server with an Intel Xeon CPU @ 2.40GHz and 2 parallel Nvidia Tesla P100 GPUs. We implement the proposed HybridDetectionGAN framework using the Keras framework. We train the detection perceptor using the Adam optimiser [127] with 120 epochs and an initial learning rate of 1e-3. The learning rate is decreased by a factor of 10 after 60 epochs. We train HybridDetectionGAN for 200 epochs. The initial learning rate of the hybrid synthesis model, the perceptual enhancement

model and two discriminators are 2e-4, and after 100 epochs, we apply a linear decay of the learning rate for all four components. The source code will be available at:https://github.com/LongChenCV/HybridDetectionGAN.

3.4 Experimental Results and Discussion

In this section, we present and discuss the experimental results and findings. We first conduct ablation experiments to investigate the influence of different components of our proposed HybridDetectionGAN framework. Then, we compare our method against several state-of-the-art methods on the two datasets. Finally, we investigate how these UIE algorithms influence the deep detectors in the following process.

3.4.1 Ablation Studies

The proposed HybridDetectionGAN framework integrates a hybrid underwater image synthesis model and a detection perceptual underwater enhancement model. We conduct ablation experiments in order to evaluate them on both unpaired ChinaMM-MultiView and paired OUC datasets.

3.4.1.1 Ablation Studies of the Hybrid Synthesis Model

We conduct ablation experiments to investigate how the CNN, scattering and absorption branches and the complete physical branch (consisting of the scattering and absorption branches) influence the synthetic image results.

Fig. 3.5 presents the qualitative comparison of the synthesis models with different component settings on MultiView dataset. We observe that our complete hybrid synthesis model produces the visual appearance most similar to the real underwater images of ChinaMM. The synthesis models without the scattering branch have a limited capability of modelling the haze-effects. For example, the resultant images of CNN-Only and CNN+Absorption synthesis models are relatively clear in spite of severe color distortion. After having incorporated the scattering prior in these two models, we clearly witness the haze-effects on the images.

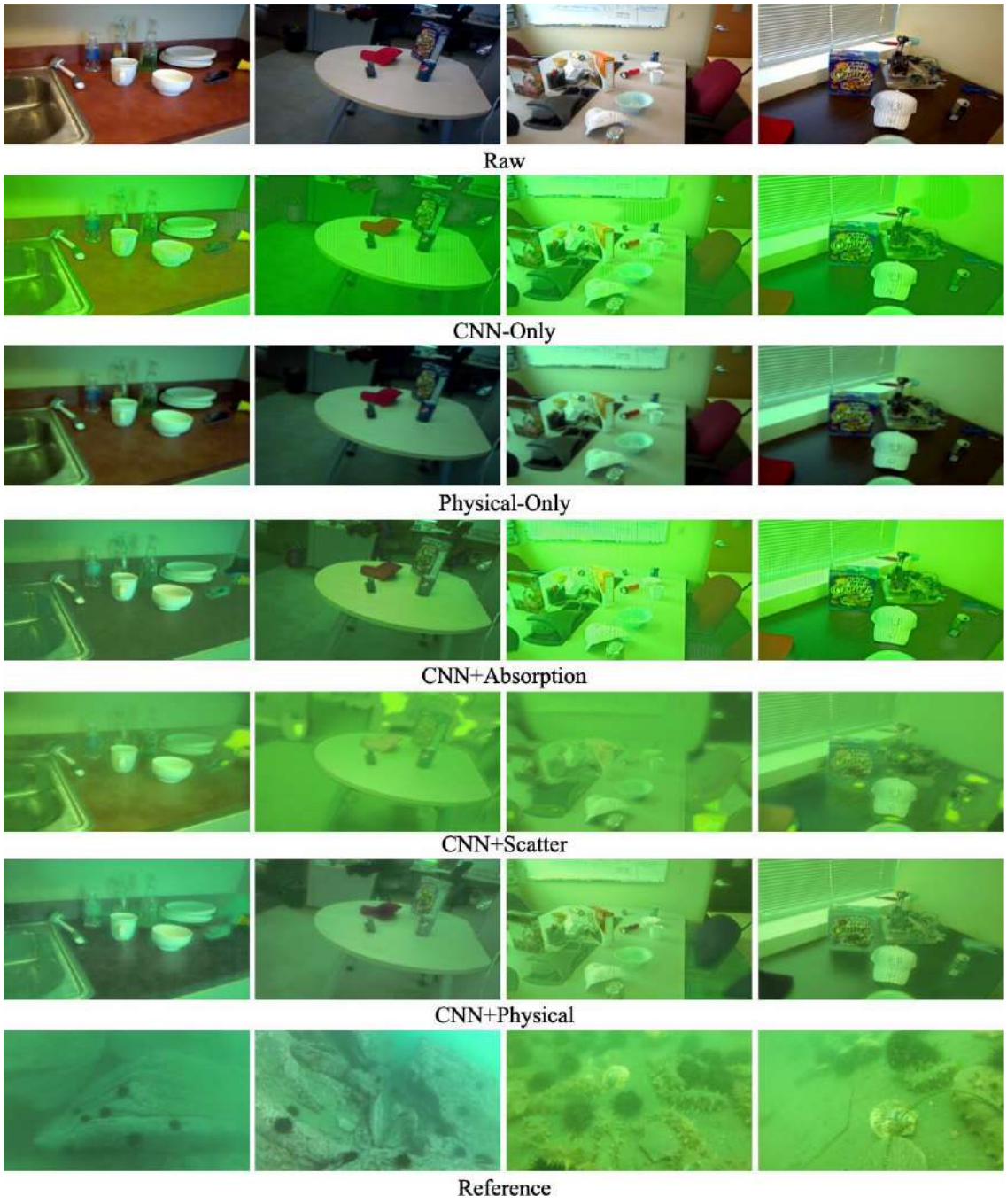

Fig. 3.5 Qualitative comparison of synthesis models with different component settings on MultiView. From left to right are raw in-air images, the results of the synthesis models with only the CNN branch, only the physical branch, CNN and absorption branches, and CNN and scattering branches, complete hybrid synthesis model, and reference underwater images from ChinaMM. Best viewed in the digital form.

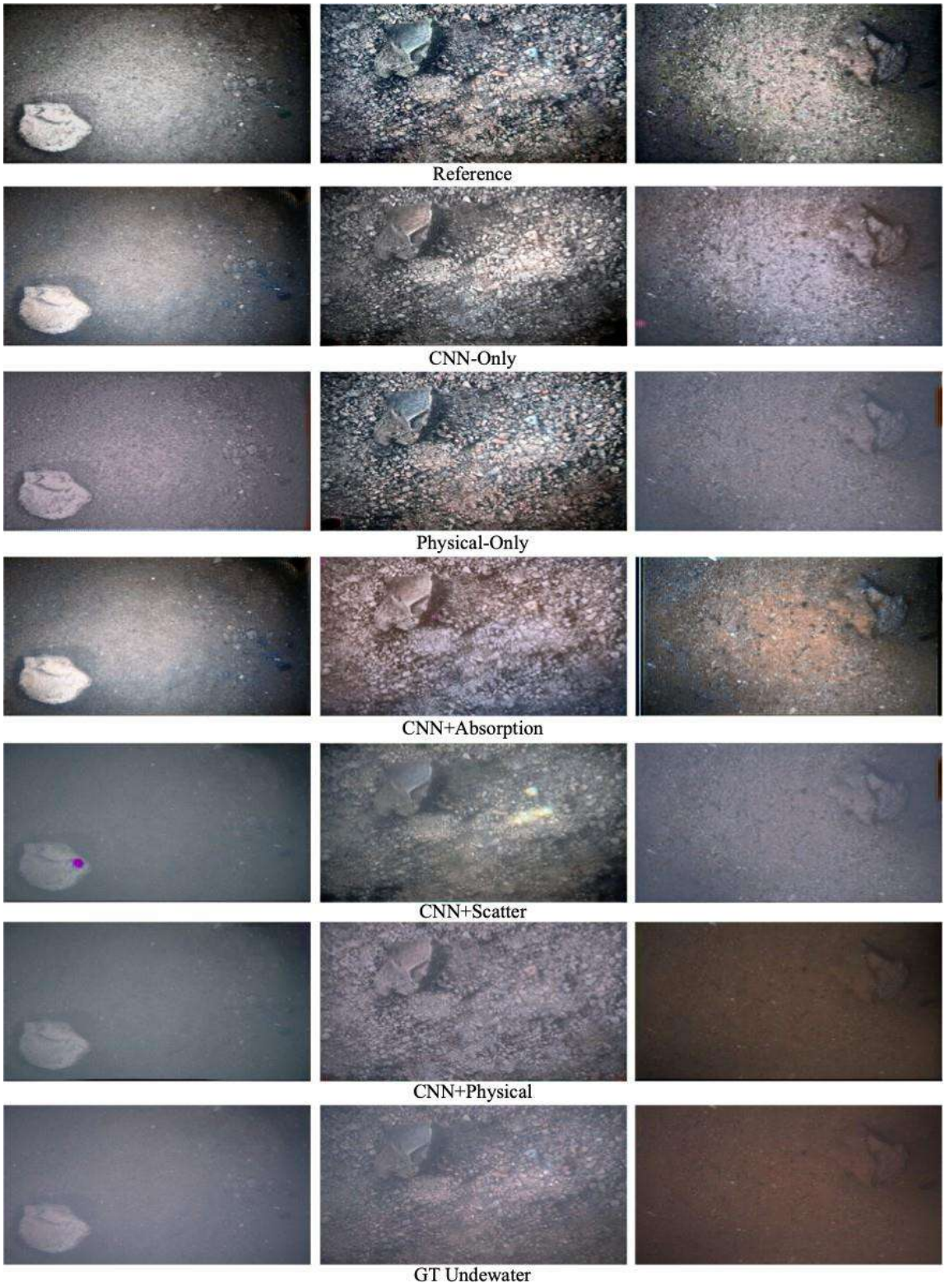

Fig. 3.6 Qualitative comparison of the synthesis models with different component settings on OUC. From left to right are high quality reference images, the results of the synthesis models with only CNN branch, only physical branch, CNN and absorption branches, and CNN and scattering branches, complete hybrid synthesis model, and the ground-truth underwater images.

Table 3.1 Quantitative comparison of the synthesis models with different components on the OUC dataset.

Model	CNN	Absorption	Scatter	MSE	PSNR	SSIM	PCQI
Synthesis Models	✓			0.2777	23.7386	0.7664	0.6627
	✓	✓		0.1360	26.8716	0.8963	0.9398
	✓		✓	0.1289	27.0851	0.9046	0.9364
		✓	✓	0.2882	23.5787	0.7659	0.6600
	✓	✓	✓	**0.0978**	**28.2895**	**0.9131**	**0.9518**

Both CNN-Only and Physical-Only synthesis models do not generate diverse results. This is because the former one usually runs into the classical mode collapse problem that only produces outcomes of a single mode, e.g., all the synthetic images are of the same color tone. The latter also generates underwater images in a monotonous style. Once the physical model has been trained, only one fixed set of parameters have been optimised, leading to optimal results within a specific environment. When we integrate the CNN and absorption branches, diverse results are obtained. The absorption prior helps CNN to generate images with different color tones whilst avoiding the artifact problem. For example, artifacts frequently occur in the resultant images of the synthesis models without the absorption branch. Fig. 3.6 presents the qualitative comparison of the synthesis models with different component settings on the OUC dataset.

In addition to the qualitative comparison, we also use four full-reference image quality evaluation metrics to evaluate the synthesis models supported by the reference images in the OUC dataset. From Table 3.1, we observe the superiority of the complete hybrid synthesis model over the other models as to four metrics. This indicates the synthetic underwater images of the hybrid model are the closest ones to the reference images. After having removed the absorption branch, the values of the four metrics decrease due to the existence of frequent artifacts. Removing the scattering branch also decreases the values of the four metrics due to the haze-effects. The synthesis model with the physical model only generates the images with color distortion and the worst quantitative scores.

3.4.1.2 Ablation Studies of the Enhancement Model

We first investigate the influence of the quality of the training data, i.e., the synthetic underwater images, on the enhancement model. Then, we analyse how the detection perceptor affects the enhancement model.

The influence of the quality of the synthetic underwater images on the enhancement model. We divide the synthetic underwater images into four categories: (A) Synthetic underwater images with incorrect color tones generated by the Physical-Only synthesis model; (B) Synthetic underwater images without evident haze-effects generated by the CNN+absorption synthesis model; (C) Synthetic underwater images with artifacts generated by the CNN+scattering synthesis model; (D) Synthetic images with pleasing appearance generated by the hybrid synthesis model.

For the unpaired ChinaMM-MultiView dataset, we examine different enhancement models on two sub-datasets, i.e., the synthetic underwater dataset, MultiviewUnderwater, which is generated by our hybrid synthesis model with the RGB-D in-air images of MultiView, and the real-world underwater dataset, ChinaMM. Fig. 3.7 shows the qualitative comparison of the enhancement models trained on different synthetic underwater images on ChinaMM and MultiviewUnderwater. We observe that the enhancement model trained on (A) (synthetic images with incorrect color tones) cannot correct the color casts due to the lack of learning on color transformation between the underwater and in-air images. The enhancement model trained on (B) (haze-free synthetic underwater images) leaves evident haze-effects on its results while the enhancement model trained on (C) (synthetic underwater images with artifacts) further aggravates the artifacts problem in the resultant images. By contrast, the enhancement model trained on (D) (visually pleasing synthetic images) performs much better in removing haze-effects and artifacts. However, the results of the enhancement model trained on (D) still suffer from minor artifacts and haze-effects. This is because the deep enhancement model is constructed using fully convolutional layers, which have a limited ability to remove artifacts and haze-effects.

Tables 3.2 reports the quantitative scores of the enhancement models trained on different synthetic underwater images on the testing sets of MultiviewUnderwater and ChinaMM. We

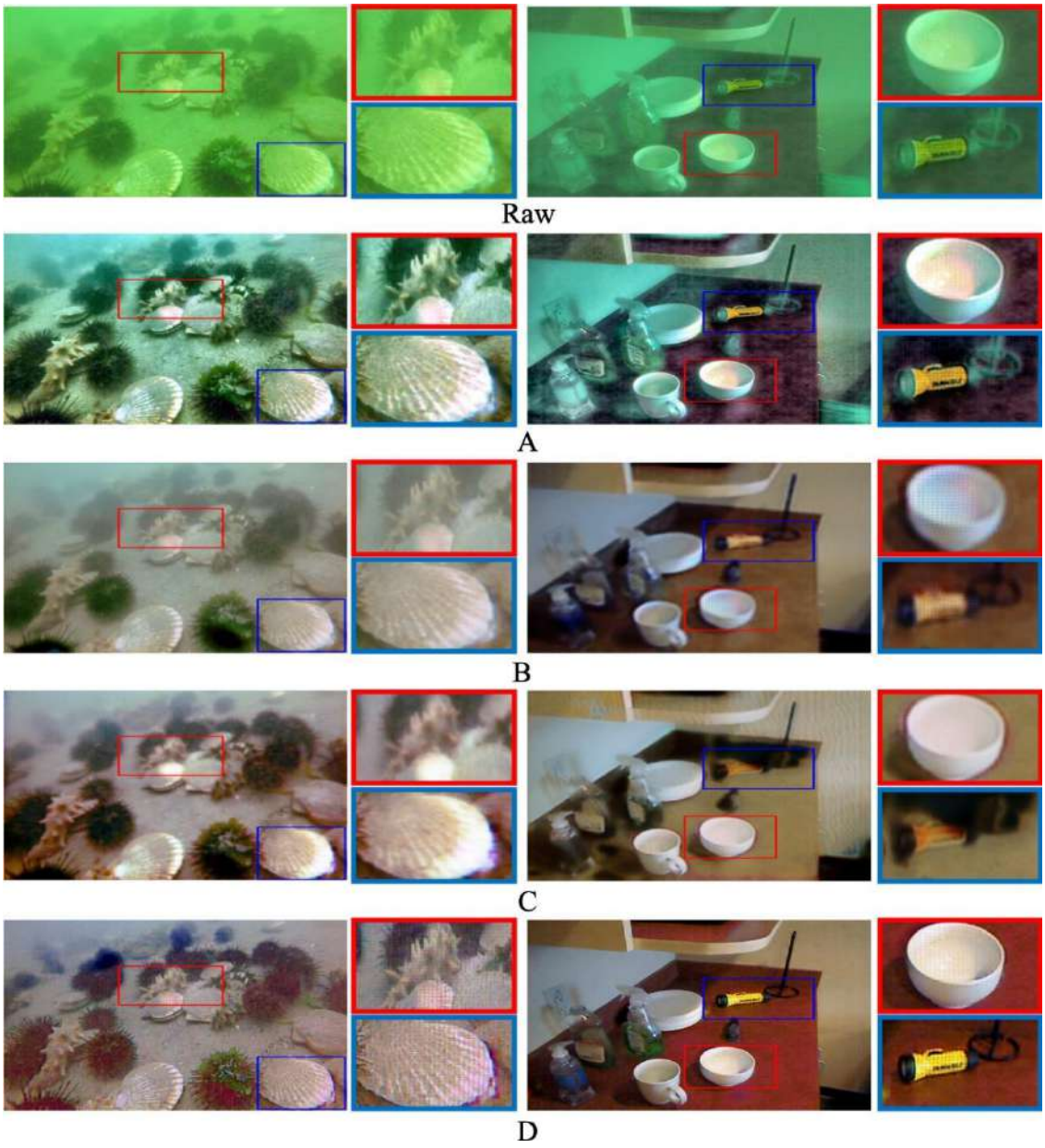

Fig. 3.7 Qualitative comparison of the enhancement models trained on different synthetic underwater images on ChinaMM (top row) and MultiviewUnderwater (bottom row). From left to right are the raw underwater images, the results of the enhancement model trained on (A) synthetic images with incorrect color tones, (B) synthetic images without evident haze-effects, (C) synthetic images with artifacts, and (D) synthetic images with pleasing appearance.

Table 3.2 Quantitative comparison of the enhancement models trained with different synthetic underwater images on the MultiviewUnderwater and ChinaMM datasets.

	MultiviewUnderwater					ChinaMM		
Enhancement models	MSE	PSNR	SSIM	PCQI	mAP	UCIQE	UIQM	mAP
Model trained on A	1.3061	15.5623	0.1906	0.5294	75.9	23.2144	2.3864	72.2
Model trained on B	1.0058	17.0883	0.2971	0.5549	76.3	25.2220	2.7138	74.0
Model trained on C	0.9249	18.4655	0.3831	0.5425	72.1	25.6417	3.1529	71.1
Model trained on D	**0.7153**	**20.5488**	**0.5632**	**0.5677**	**77.5**	**27.3206**	**3.8196**	**76.5**
GT in-air image	0.0000	Inf	1.0000	1.0000	79.9	-	-	-
GT underwater image	-	-	-	-	-	21.3083	1.4410	68.6

also list the full-reference scores of the ground-truth in-air images on MultiviewUnderwater and the non-reference scores of the ground-truth underwater images of ChinaMM as references. In terms of image quality evaluation metrics, the enhancement model trained on (D) visually pleasing images performs best while the one trained on (A) images with color distortion achieves the worst scores. The color cast in the results of the latter enhancement model leads to the decreasing scores of the image quality metrics. In terms of mAP, the model trained on (C) images with artifacts achieves the lowest score even though it has relatively higher quantitative scores for the image quality evaluation metrics. The artifacts smear the details of the images or objects, deteriorating the detection accuracy more than the incorrect color casts and haze-effects. The enhancement models trained on the visually pleasing images obtain the best scores of image quality evaluation metrics and mAP on the two datasets. In summary, the enhancement model trained on more realistic synthetic underwater images can learn more accurate mappings between the images in underwater and in-air domains and be better generalised on the real-world underwater dataset.

The influence of the detection perceptor on the enhancement model. We compare three enhancement models with different detection perceptor settings, i.e., enhancement model without detection perceptor (denoted as OursWDP), enhancement model with a patch detection perceptor (denoted as OursPatch), and enhancement model with an object-focused detection perceptor (denoted as OursOF).

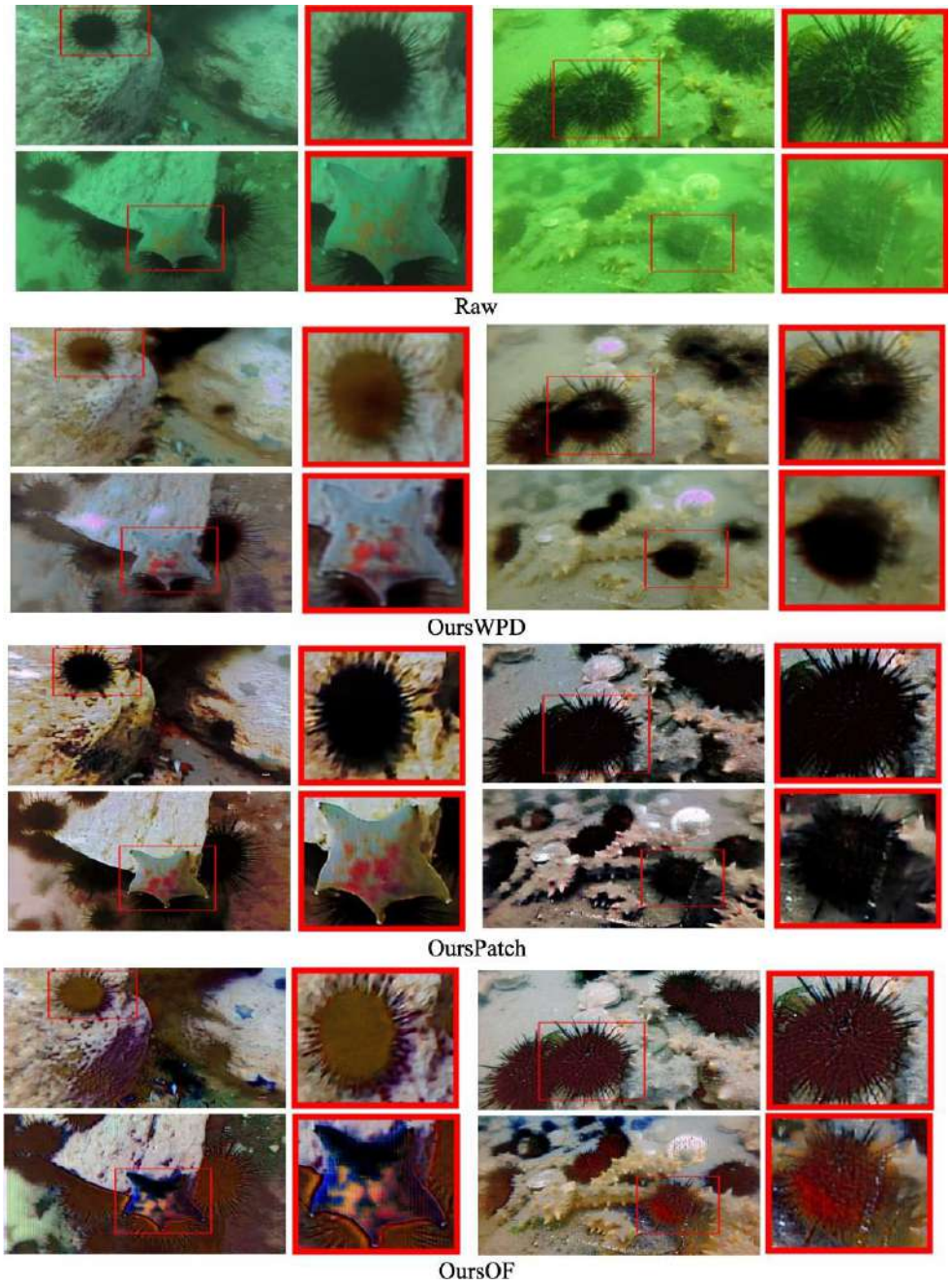

Fig. 3.8 Qualitative comparison of the enhancement models with different detection perceptor settings on ChinaMM. From left to right are the raw underwater images, the results of OursWDP, OursPatch, and OursOF, respectively.

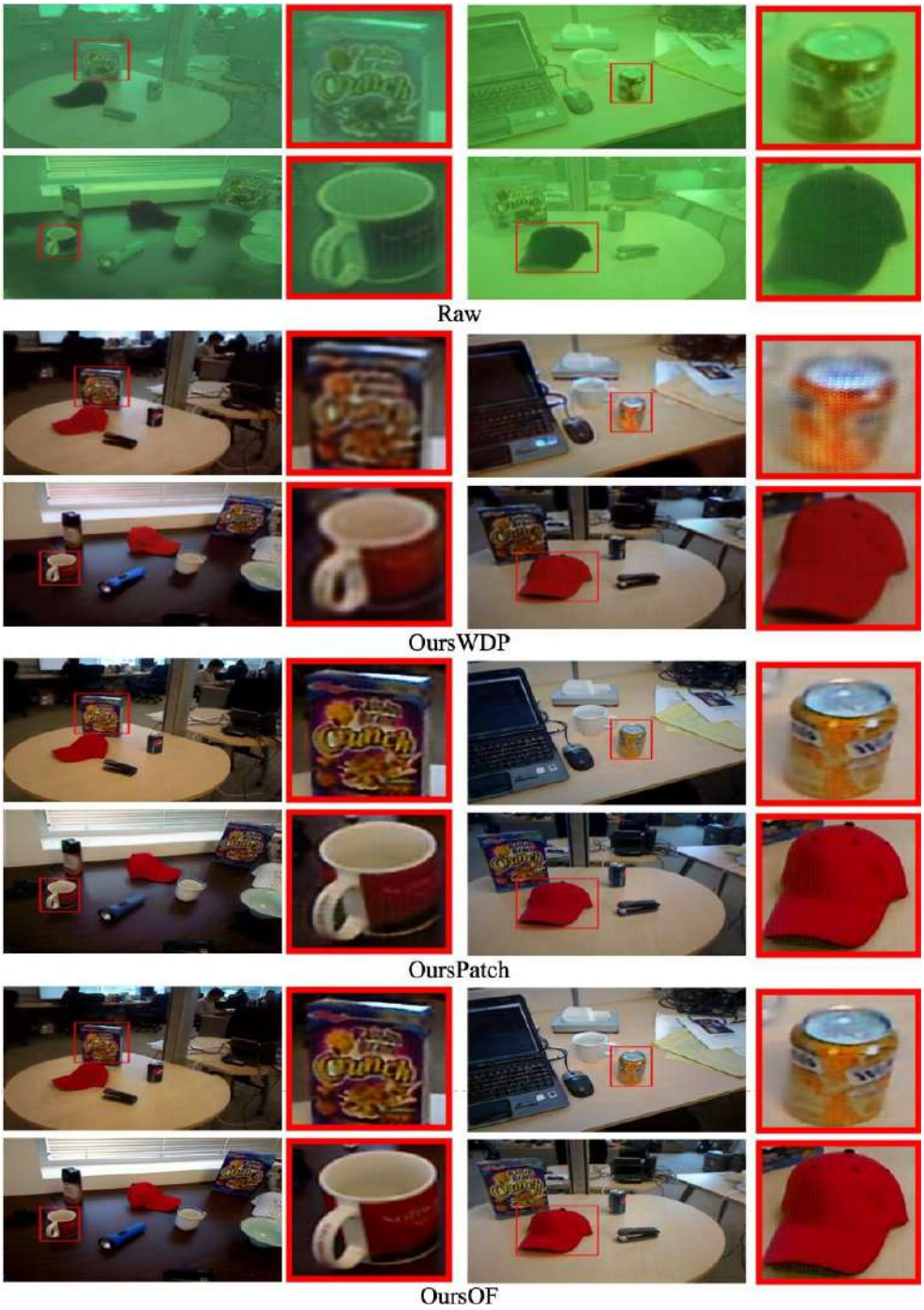

Fig. 3.9 Qualitative comparison of the enhancement models with different detection perceptor settings on MultiviewUnderwater. From left to right are raw underwater images, results of OursWDP, OursPatch, and OursOF, respectively.

Fig. 3.10 Visualization of object detection results after having applied enhancement models with different perceptor settings on ChinaMM. GT donetes the ground truth annotations. Raw denotes the detection results without applying any enhancement algorithms.

Fig. 3.8 and Fig. 3.9 present the qualitative comparison of the enhancement models with different detection perceptor settings on ChinaMM and MultiviewUnderwater, respectively. We observe that without a detection perceptor, the enhanced results of OursWDP on the two datasets still contain artifacts and haze-effects. OursPatch removes artifacts and restores the details of the image patches such as color tones, visibility, and saturation on the two datasets. This is because the patch detection perceptor trained on the high quality in-air images can properly learn potential attributes of high visual quality. These potential attributes, in the form of gradients, help restore the details of the image patches. We notice that the detection perceptor is only trained on the in-air images of MultiView, and the object categories of MultiView are different from those of ChinaMM, however, the detection perceptor helps the enhancement model to improve the quality of the images of ChinaMM. This indicates that a well-trained detection perceptor not only encodes category-dependent attributes but also category-agnostic attributes such as high quality edges, textures and colors. Compared to OursPatch, OursOF seems to generate sharp objects with high contrast between the objects and the background. We also reveal the detection results after having applied different enhancements models on ChinaMM in Fig. 3.10, from which we can see the detector trained on the enhanced results of OursOF achieves the best detection results.

Table 3.3 reports the quantitative results of the enhancement models with different detection perceptor settings on MultiViewUnderwater and ChinaMM. On MultiviewUnderwater, OursPatch achieves the best full-reference scores, and the corresponding deep detector obtains almost the same mAP as the one trained on the ground-truth in-air images of MulitiView. This quantitative performance attributes to its enhanced results similar to the ground-truth

Table 3.3 Quantitative comparison of the enhancement models with different detection perceptor settings on the MultiviewUnderwater and ChinaMM datasets.

	MultiviewUnderwater					ChinaMM		
Models	MSE	PSNR	SSIM	PCQI	mAP	UCIQE	UIQM	mAP
OursWDP	0.7153	20.5488	0.5632	0.5677	77.5	27.3206	3.8196	76.5
OursPatch	**0.2453**	**33.3680**	**0.9374**	**0.8441**	79.9	**32.5976**	**4.8720**	79.3
OursOF	0.4683	26.1421	0.6364	0.6741	**86.7**	28.3269	4.2194	**83.9**

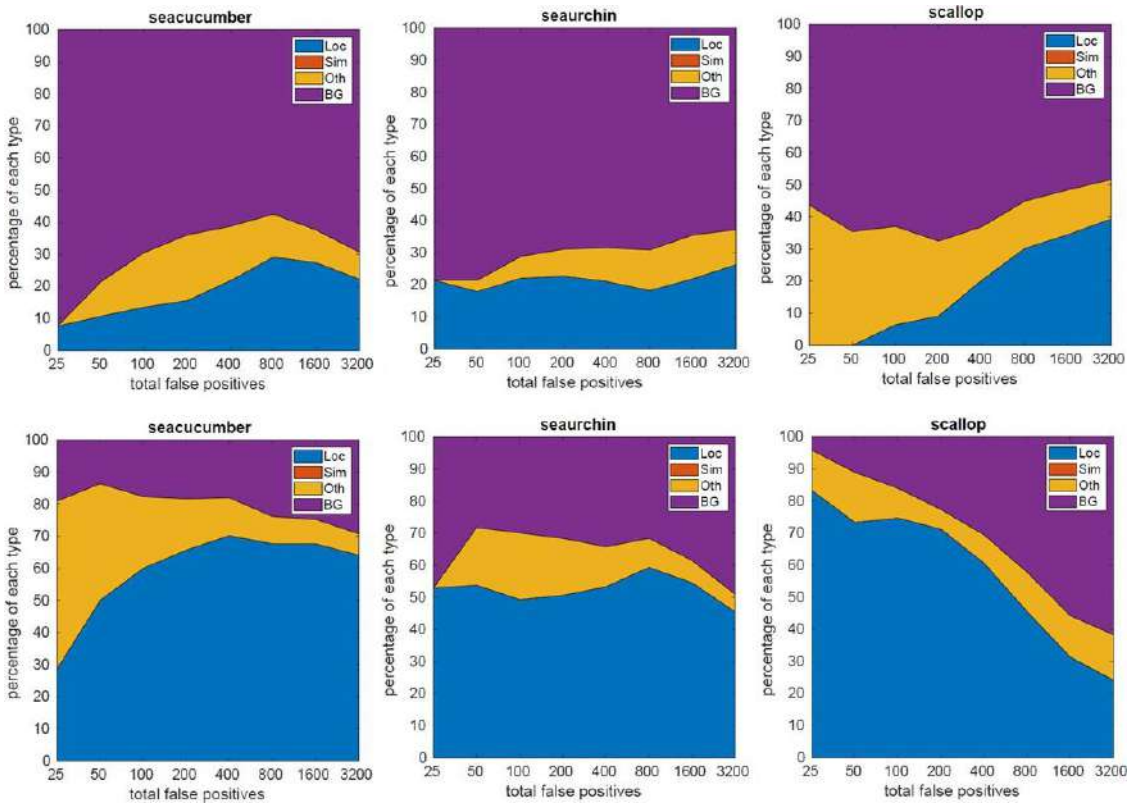

Fig. 3.11 The distribution of the top-ranked false positive measures for images of ChinaMM. The false positive measures include localisation errors (Loc), confusion with similar categories (Sim), with others (Oth), or with background (BG). The top row shows the results of OursPatch and the bottom row shows the results of OursOF.

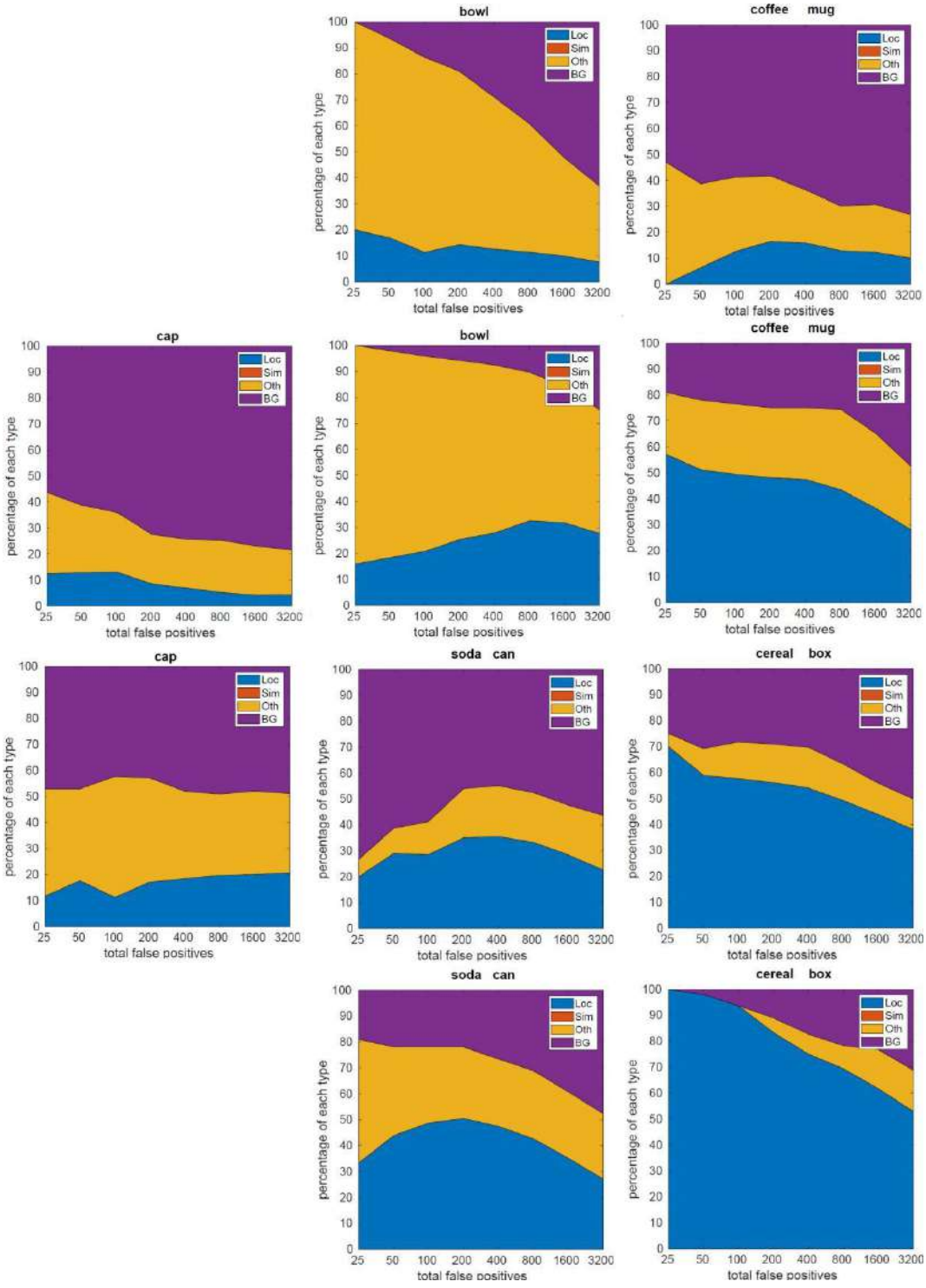

Fig. 3.12 The distribution of the top-ranked false positive measures for images of MultiviewUnderwater. The false positive measures include localisation errors (Loc), confusion with similar categories (Sim), with others (Oth), or with background (BG). The top row shows the results of OursPatch and the bottom row shows the results of OursOF.

Table 3.4 Quantitative comparison of the enhancement models with different detection perceptor settings on the OUC dataset.

Models	MSE	PSNR	SSIM	PCQI	mAP
OursWDP	0.1168	28.3034	0.85821	0.7147	83.5
OursPatch	**0.0224**	**35.4039**	**0.9724**	**0.9389**	86.5
OursOF	0.0616	30.8662	0.9351	0.9030	**90.1**

in-air images. The deep detector trained on the enhanced results of OursOF achieves the best mAP. We have similar experimental results on ChinaMM, where OursPatch achieves the best UCIQE score (32.5976) and UIQM score (4.8720), whilst OursOF achieves the best mAP (83.9). We believe that the best detection accuracy is due to the reduction of the disturbing background. To verify this assumption, we use the detection tool of [128] to analyse the false positives of the two detectors trained on the results of OursPatch and OursOF. Fig. 3.11 and Fig. 3.12 show the distribution of the top-ranked false positive measures for each category on the testing sets of ChinaMM and MultiviewUnderwater. The former detector cannot well distinguish the objects with complex background while the latter one largely reduces the background errors. The qualitative and quantitative comparisons of the enhancement models with different detection perceptor settings on the paired OUC dataset can be found in Fig. 3.13 and Table 3.4.

3.4.2 Comparison with State-of-the-art Methods

In this subsection, we first compare our hybrid synthesis model with three state-of-the-art UIS algorithms. Then, we compare our two detection based perceptual enhancement models with eleven state-of-the-art UIE algorithms.

Comparison with state-of-the-art UIS methods. We compare our hybrid synthesis model (denoted as OursHybrid) with three state-of-the-art UIS algorithms, including physical model based UIS method (denoted as Physical) [11], CycleGAN [102] and WaterGAN [98]. Physical [11] applied the physical underwater image formation model and 10 groups of pre-defined parameters to synthesise 10 Jerlov type underwater images from the RGB-D in-air images, and the synthetic dataset contains 10 types of underwater images with various

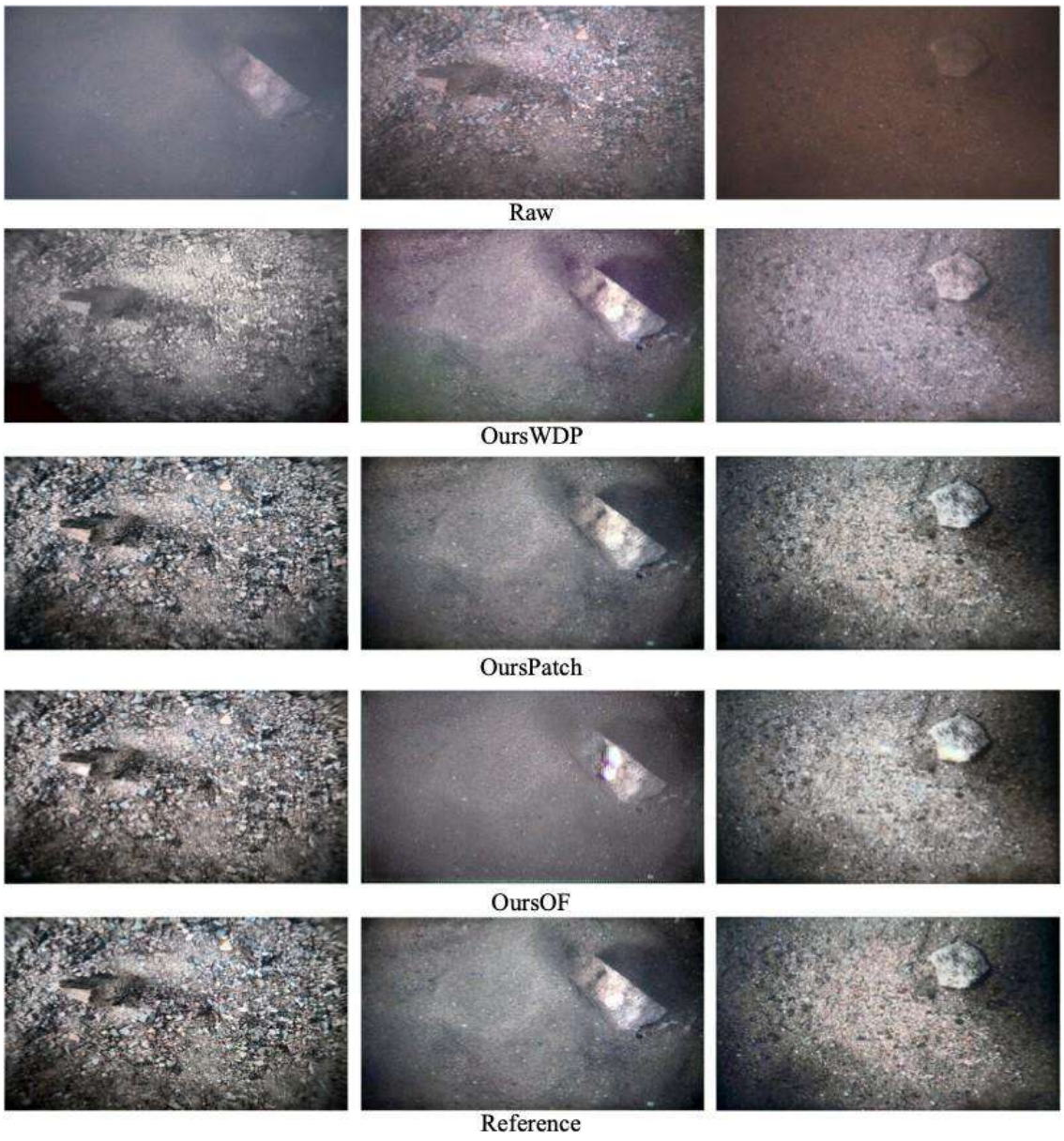

Fig. 3.13 Qualitative comparison of the synthesis models with different perceptor settings on the OUC dataset.

color distortions and haze-effects. We choose the images most similar to the ChinaMM-style underwater images in the qualitative comparison. WaterGAN and CycleGAN are deep learning based UIS methods, we tune their parameter settings to generate satisfactory results.

Fig. 3.14 shows the qualitative comparison of different UIS algorithms on MultiView. It is evident that the resultant images of our proposed hybrid synthesis model are very similar to the reference images of ChinaMM in terms of diversity, color casts and haze-effects. In contrast, the results of WaterGAN suffer from insufficient haze-effects due to the lack of light scattering prior, even though they apply a shallow convolutional network to simulating the light scattering process. However, in practice, without referring to the optical property of light scattering, the ability of CNN to simulate haze-effects is limited. In addition, the results of WaterGAN suffer from unrealistic color distortion even though it has used the light absorption prior. This is mainly because the weakly supervised WaterGAN is trained with only an adversarial loss on the unpaired images, which cannot provide sufficient supervision information to simulate color distortion. Apart from the adversarial loss, CycleGAN adds a pixel-wise cycle-consistency loss as an additional constraint to supervising the training, and the cycle-consistency loss to minimise the discrepancy between the generated and the ground-truth images in the pixel level that leads to more realistic color distortion. Nevertheless, evident artifacts without haze-effects appear in the results of CycleGAN because the CNN structure has the limited capability to simulate haze-effects and remove artifacts. It is also worth noting that WaterGAN and CycleGAN generate underwater images in a monotonous style due to the model collapse problem, restricting their generalisation capability to yield diverse real-world underwater images. By integrating the physical prior into CNN, OursHybrid can generate images with diverse properties. Physical is able to generate artifact-free results but cannot simulate realistic color distortions as it uses the fixed parameters defined by the Jerlov water type. The qualitative and quantitative comparison of different UIS methods on OUC can be found in Fig. 3.15.

Comparison with state-of-the-art UIE methods. We compare our two detection perceptual enhancement models with eleven state-of-the-art UIE algorithms, including six physical model based methods (i.e., UDCP [87], GDCP [88], Blurriness [95], Regression [94], Red-

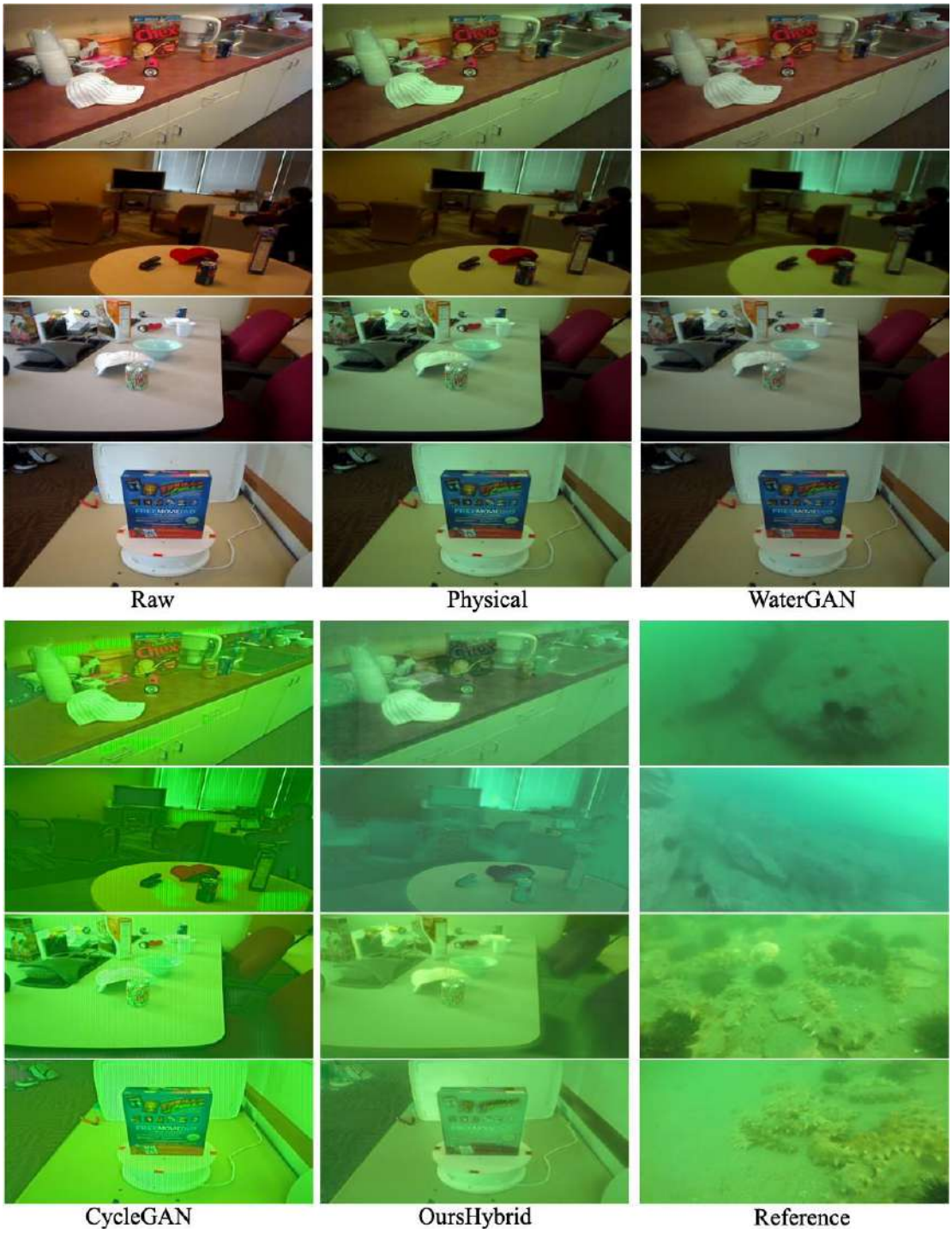

Fig. 3.14 Qualitative comparison of different UIS algorithms on the MultiView dataset. From left to right are raw in-air images of MultiView, results of Physical [11], WaterGAN [98], CycleGAN [102], OursHybrid, and real underwater images from ChinaMM as the references. Best viewed in the digital form.

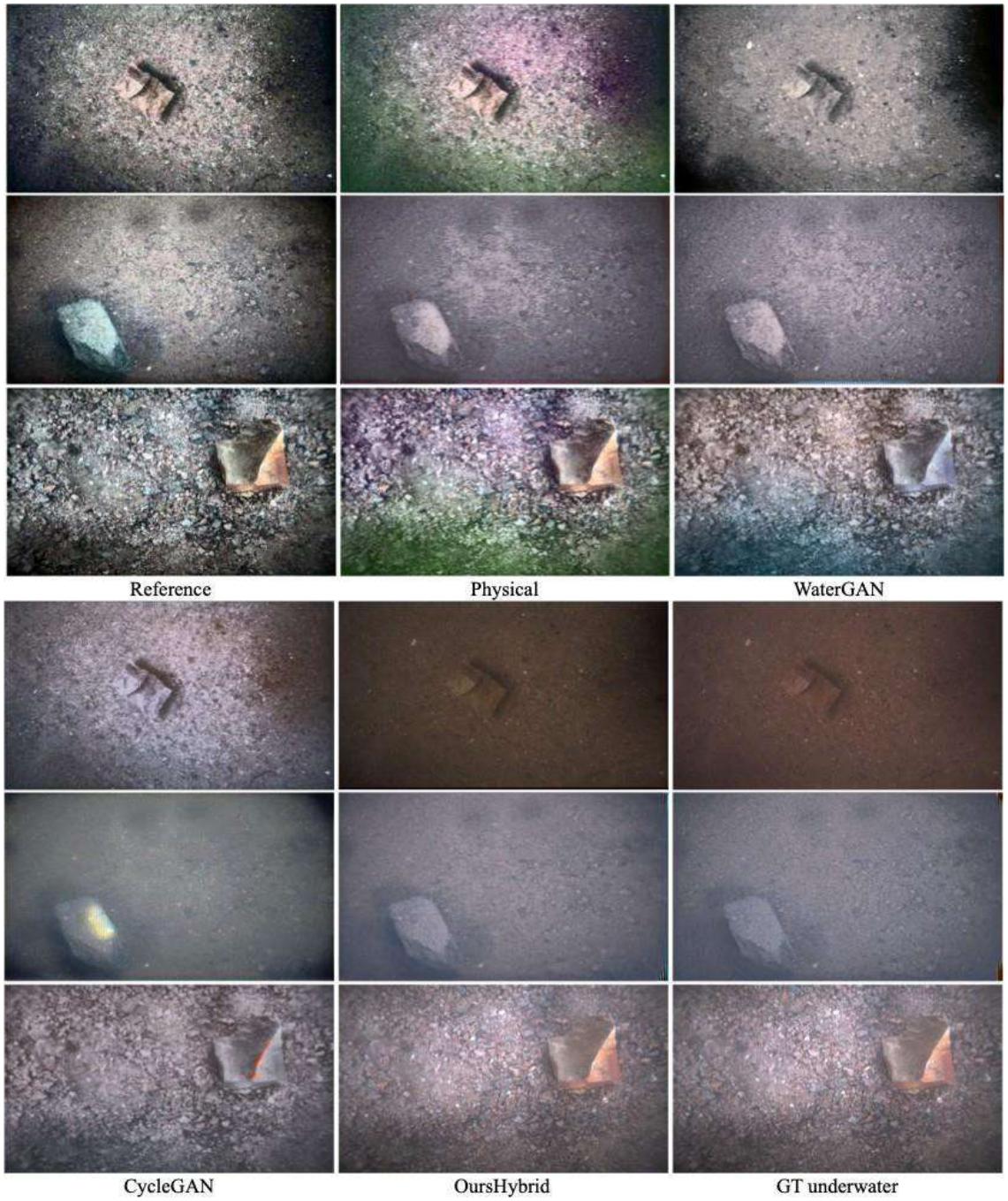

Fig. 3.15 Qualitative comparison of different UIS algorithms on the OUC dataset. From left to right are high-quality reference images, results of Physical, WaterGAN, CycleGAN, OursHybrid, and ground-truth underwater images.

Channel [90] and Histogram [96]), three model-free methods (i.e., Fusion [75], Two-step [77], and Retinex [76]), and two deep learning based methods (i.e., DUIENet [99] and CycleGAN [102]) on three datasets (i.e., synthetic MultiviewUnderwater, real-world ChinaMM and OUC). To investigate the performance of deep learning based UIE algorithms on the real-world datasets, we add another two deep learning based UIE algorithms (i.e., FUnIEGAN [97] and AIOGAN [129]) as comparison methods on real-world dataset ChinaMM.

Fig. 3.16 and Fig. 3.17 show the qualitative comparison of different UIE algorithms on ChinaMM and MultiviewUnderwater, respectively. For the two datasets, most of the physical model based UIE algorithms cannot deal with severe color distortions. Among them, Histogram performs relative better for color distortions which benefits from the histogram prior that it uses. But it generates over-saturation and excessive contrast in some image regions. UDCP and Blurriness even aggravate the color distortion due to the limitations of the priors used in these two methods. Regression tends to introduce bluish color casts on account of its inaccurate color correction algorithm, and GDCP over enhances the brightness that results in the loss of image details. RedChannel improves little on color distortion. Among the model-free algorithms, Retinex can effectively remove color distortion and produce more natural scenes while Fusion improves little on color distortion. Twostep over-enhances the contrast of the underwater images and generates unnatural results. The deep learning based methods FUnIEGAN, AIOGAN, DUIENet and CycleGAN can deal with color casts, however, both of them still leave evident haze on the resultant images. In terms of haze removal, most of the physical model based methods are able to remove haze-effects to some extent, benefiting from the use of light scattering priors. Among the model-free methods, Retinex and Fusion effectively remove the haze-effects on the underwater images while Twostep contributes little towards haze removal. Among all the UIE methods, OursPatch achieves the best qualitative performance in terms of color tones, visibility, saturation and contrast. The qualitative comparison of different UIE algorithms on OUC can be found in Fig. 3.18 .

In addition to qualitative evaluations, we also report the quantitative results of different UIE algorithms. The best values are marked in bold. The quantitative results on the testing

Fig. 3.16 Qualitative comparison of different UIE algorithms on the ChinaMM dataset.

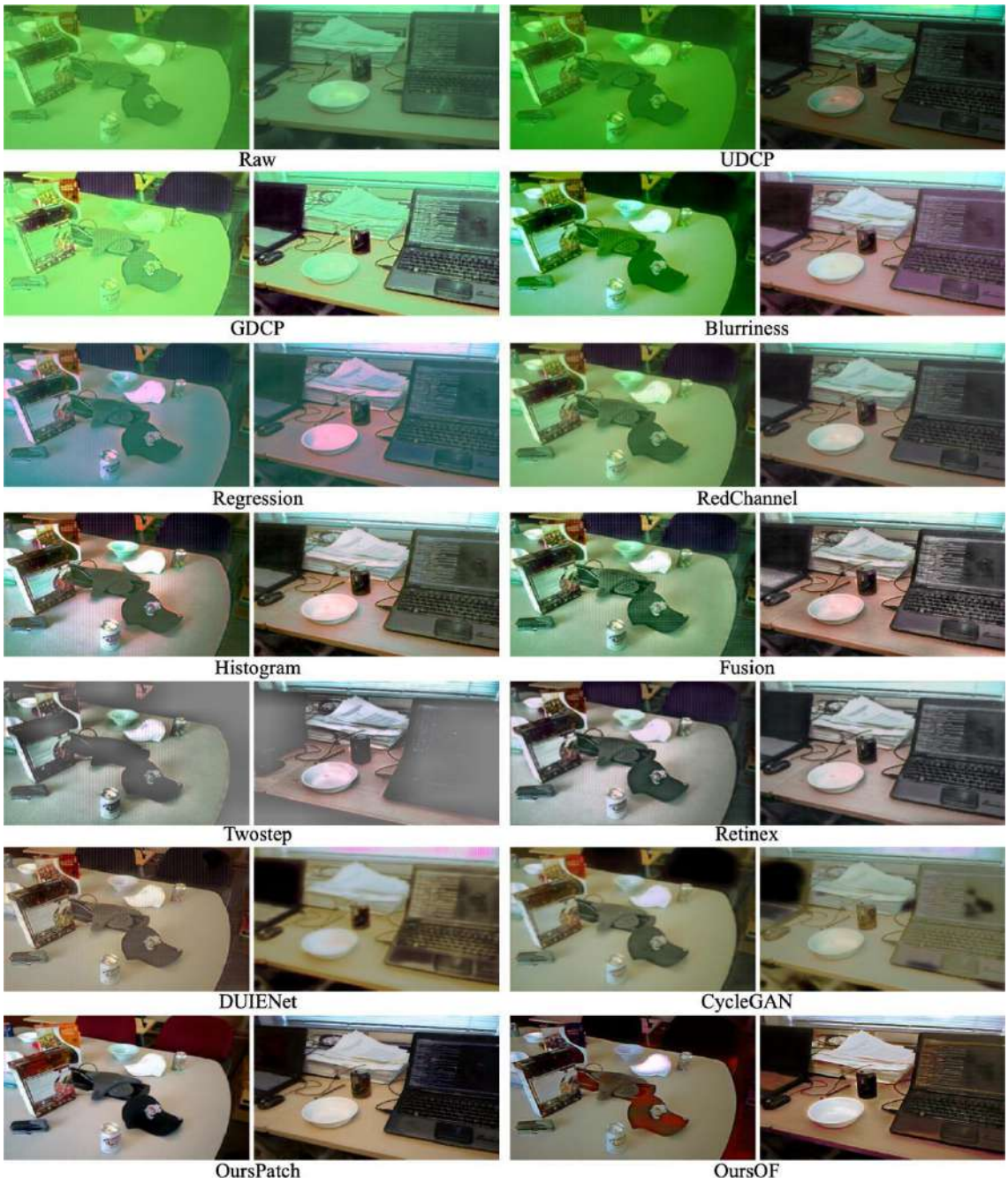

Fig. 3.17 Qualitative comparison of different UIE algorithms on the MultiviewUnderwater dataset.

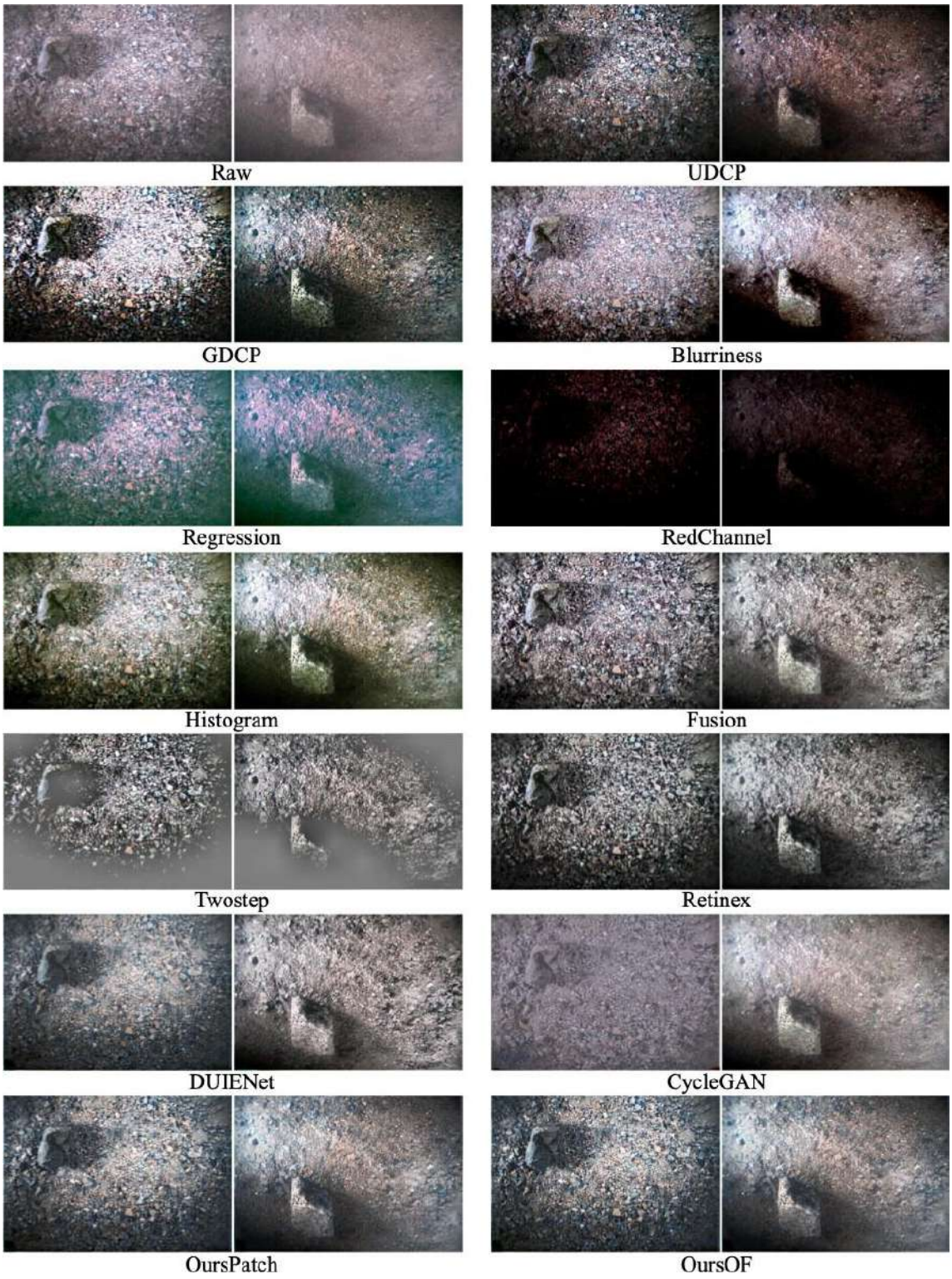

Fig. 3.18 Qualitative comparison of different UIE methods on the OUC dataset.

Table 3.5 Full-Reference image quality and detection accuracy evaluations on the synthetic MultiviewUnderwater dataset.

Methods	MSE	PSNR	SSIM	PCQI	mAP
UDCP	1.0577	18.510	0.1840	0.5463	72.1
GDCP	3.2625	13.443	0.2710	0.5374	74.1
Blurriness	0.9387	19.613	0.2518	0.5965	74.4
Regression	0.7245	19.921	0.2125	0.5997	71.9
RedChannel	0.7298	20.271	0.4035	0.5939	76.4
Histogram	0.8553	19.1972	0.5479	0.5972	78.3
Fusion	1.1339	18.232	0.4604	0.5775	77.1
Twostep	2.1455	15.010	0.3712	0.5002	66.0
Retinex	0.7351	19.858	0.5222	0.6333	78.2
DUIENet	0.2689	23.8766	0.7768	0.6840	76.3
CycleGAN	0.7778	20.7561	0.5509	0.5558	74.8
OursPatch	**0.0453**	**33.3687**	**0.9374**	**0.8441**	79.9
OursOF	0.1683	26.1421	0.6364	0.6741	**86.7**

Table 3.6 Full-Reference image quality and detection accuracy evaluations of different UIE algorithms on the OUC dataset.

Methods	MSE	PSNR	SSIM	PCQI	mAP
UDCP	3.6147	12.9696	0.4807	0.4181	87.1
GDCP	2.4115	15.7764	0.6316	0.5660	86.9
Blurriness	0.6100	20.9975	0.7239	0.6493	86.4
Regression	0.4294	22.1125	0.5343	0.6620	81.6
RedChannel	7.1857	9.7217	0.1798	0.1694	41.6
Histogram	0.5325	21.3031	0.7531	0.8089	81.5
Fusion	0.2816	28.5319	0.8794	0.8940	83.9
Twostep	1.6242	16.1558	0.6108	0.4785	74.8
Retinex	0.3469	28.0519	0.8886	0.8367	87.2
DUIENet	0.1217	27.9608	0.8412	0.7405	84.0
CycleGAN	0.1361	26.8538	0.8970	0.9355	82.0
OursPatch	**0.0224**	**35.4039**	**0.9724**	**0.9389**	86.5
OursOF	0.0616	30.8662	0.9351	0.9030	**90.1**

Table 3.7 Non-Reference image quality and detection accuracy evaluations on the ChinaMM dataset.

Methods	UCIQE	UIQM	UICM	UISM	UICONM	mAP
Raw	21.3083	1.4410	-80.1429	6.7799	0.4669	68.6
UDCP	28.6184	3.0184	-56.7266	6.7033	0.7380	71.6
GDCP	**33.6328**	2.6468	-53.7373	**6.7830**	0.6039	72.7
Blurriness	30.5865	3.6984	-58.1656	6.7155	**0.9385**	77.3
Regression	29.3877	3.7616	-21.9885	6.7060	0.6716	71.6
Histogram	33.3443	4.6728	0.4558	6.6955	0.6170	73.7
RedChannel	30.8712	3.3028	-31.2248	6.7210	0.7482	76.8
Fusion	31.7698	4.0696	-22.2500	6.6525	0.7643	75.5
Twostep	15.1238	2.6728	-4.3928	5.7978	0.3033	58.6
Retinex	28.447	4.7306	-0.4811	6.6751	0.7755	78.8
FUnIEGAN	30.4528	3.6066	-34.2738	6.7427	0.7221	73.6
AIOGAN	30.8534	3.3731	-40.8772	6.6010	0.7206	72.8
DUIENet	31.5588	2.7021	-39.4952	6.6242	0.5201	71.6
CycleGAN	30.7922	3.7036	6.9885	6.5753	0.4376	67.8
OursPatch	32.5976	**4.8720**	**13.9967**	6.7326	0.6962	79.3
OursOF	32.0967	4.4621	12.4050	6.5311	0.6107	**83.9**

set of MultiviewUnderwater and OUC are presented in Table 3.5 and Table 3.6, from which we can see that OursPatch achieves the lowest MSE score and the highest scores of PSNR, SSIM and PCQI. A higher PSNR score and a lower MSE score denote that the result is closer to the reference image in terms of image content, while a higher SSIM score denotes that the result is more similar to the reference image in terms of image structure and texture. Table 3.7 gives the quantitative scores of non-reference metrics, i.e., UIQM, UCIQE, and three components of UIQM (UICM, UISM and UIConM). The following algorithms performs best over one single metric: GDCP achieves the best UCIQE score, and Blurriness achieves the best UICONM score. However, the results of both suffer from serious color casts as shown in Fig. 3.16. There exists certain discrepancy between the qualitative images and the quantitative scores in some cases which has also been verified in [9]. For mAP, the Precision/Recall curves in Fig. 3.19 and Fig. 3.20 show that OursOF (black curve) performs best across all the categories on MultiviewUnderwater and ChainMM, suggesting that the

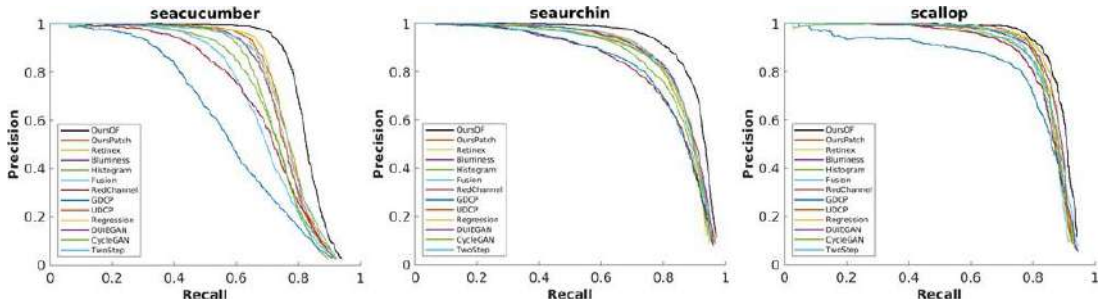

Fig. 3.19 Precision/Recall curves of deep detectors trained on the results of different UIE methods on ChinaMM.

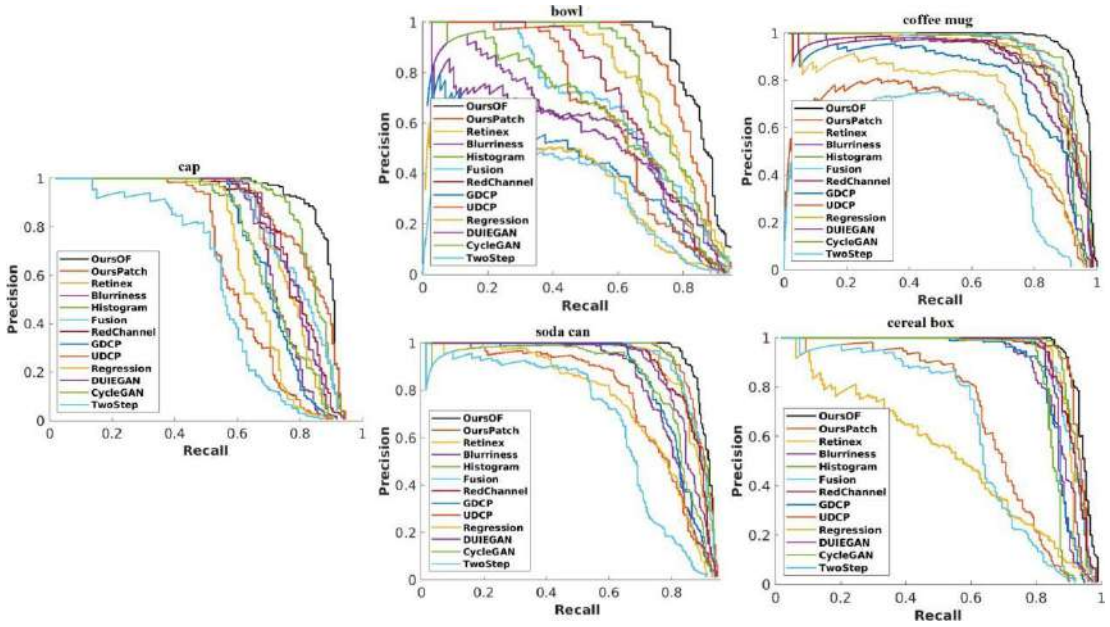

Fig. 3.20 Precision/Recall curves of different methods on the MultiviewUnderwater dataset.

interaction between the enhancement model and the detection perceptor brings significant performance improvement for the following deep detector.

To demonstrate the generalisation ability of our proposed method, we also compare our proposed method with several representative UIE algorithms (Contrast [123], Drews et al. [87], Peng et al. [95], Ancuti et al. [130], Ancuti et al. [80], Emberton et al. [131], Ancuti et al. [80], and Berman et al. [123]) provided on the webpage of the Berman dataset (stated above). The Berman dataset only contains 114 images, which are not sufficient to train OursPatch. We conduct data augmentation by randomly cropping 114 high resolution images (5,474x4,653 pixels) into 2,280 patches (512x512 pixels) and use the 2,280 patches

Fig. 3.21 Qualitative comparison of OursPatch with data augmentation (B) and without data augmentation (A).

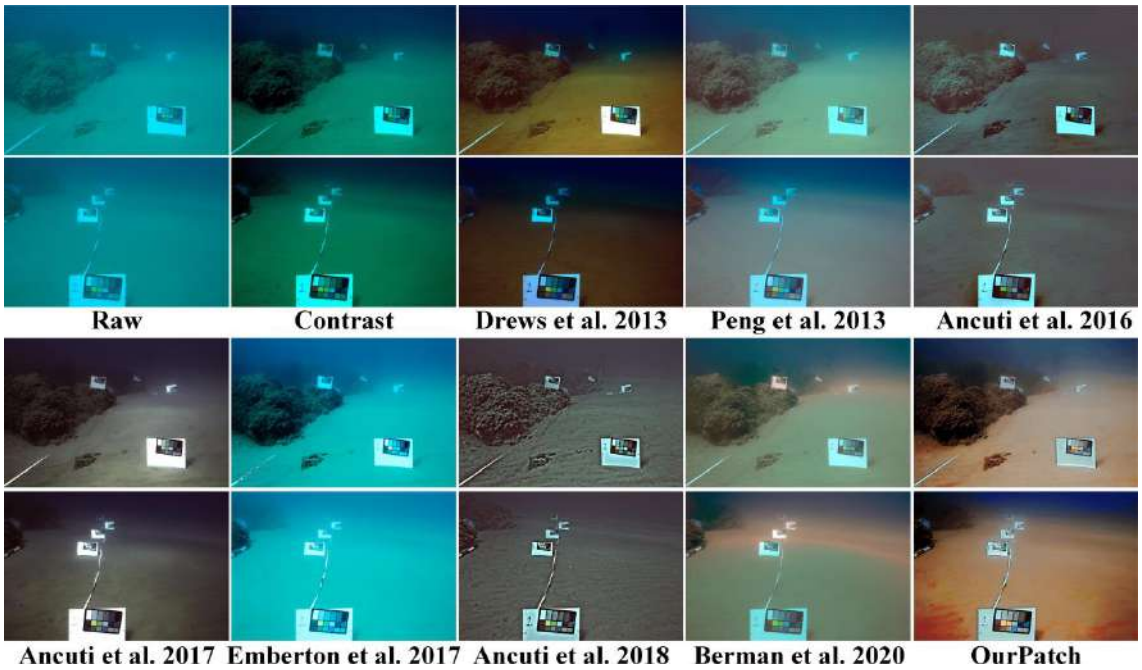

Fig. 3.22 Qualitative comparison of different UIE algorithms on the Berman dataset. From left to right are raw underwater images, results of Contrast [123], Drews et al. [87], Peng et al. [95], Ancuti et al. [130], Ancuti et al. [80], Emberton et al. [131], Ancuti et al. [80], Berman et al. [123] and OursPatch.

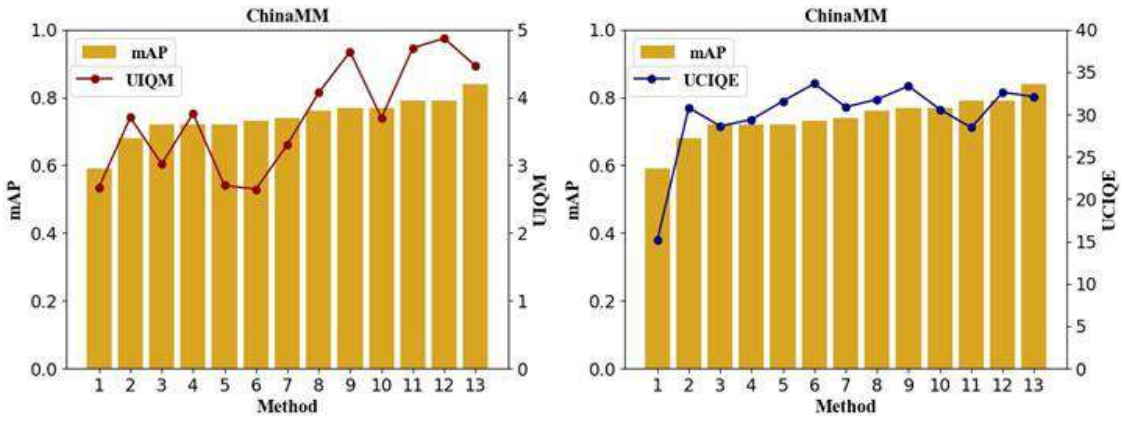

Fig. 3.23 Image quality evaluation metrics and mAP on ChinaMM. The histogram represents the mAP and the polyline represents different image quality evaluation metrics. Numbers 1 to 13 refer to thirteen UIE algorithms ordered according to increasing mAP values.

to train OursPatch. Fig. 3.21 shows that OursPatch trained on 114 images cannot generate satisfactory results, and data augmentation helps greatly improve the visual quality of the enhanced results. The Berman dataset does not provide high quality reference images and bounding box annotations, so we only present qualitative comparison of OursPatch and the contrast UIE algorithms in Fig. 3.22. For fair comparison, we directly download the enhanced results of the contrast UIE algorithms from the webpage above. We observe that OursPatch performs much better in restoring the colors of the color charts and sands.

3.4.3 The influences of UIE algorithms on the detection task.

Previous works seem to suggest that UIE algorithms will bring improvements of image quality, which further boosts the performance of the following high-level detection tasks. We conduct the following analysis to investigate whether or not UIE brings improvement of detection accuracy.

We compare the quantitative scores of the raw underwater and the enhanced underwater images by different UIE algorithms on the three underwater datasets. It is observed that not all the UIE algorithms increase the quantitative scores of the raw underwater images. Deep learning based methods can stably improve image quality on both full-reference and non-

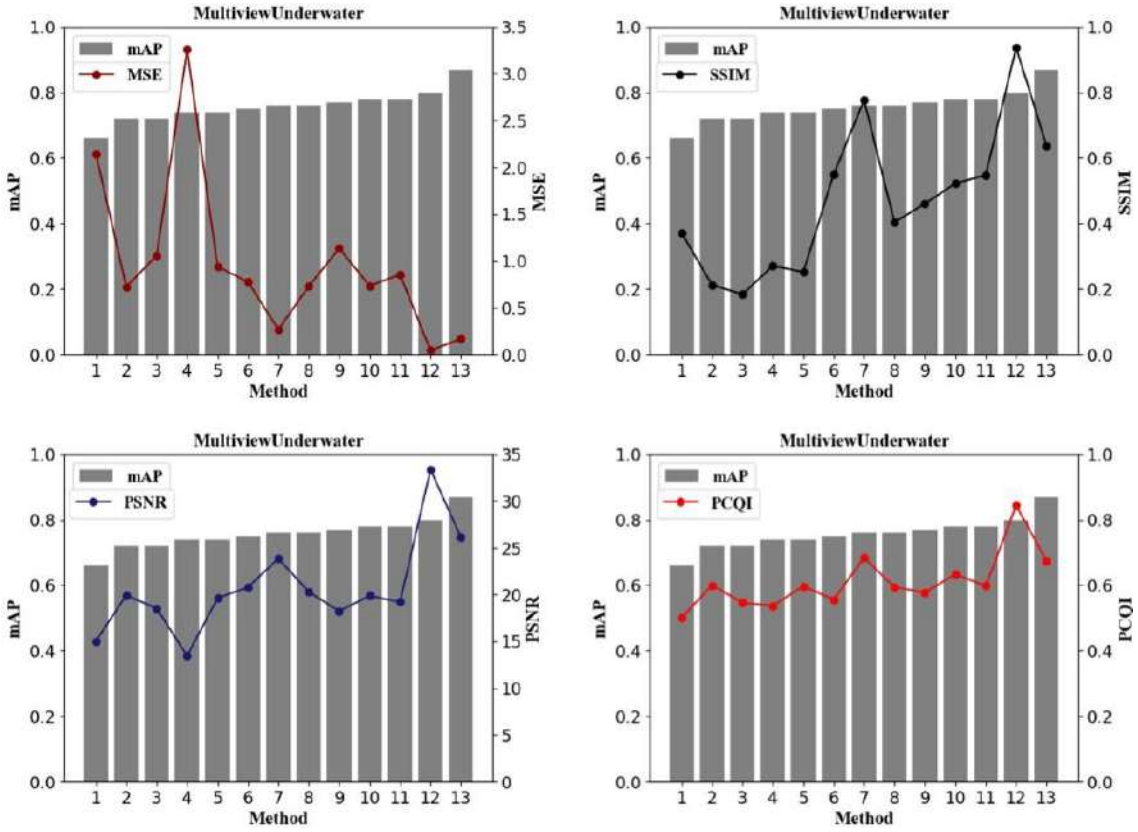

Fig. 3.24 Image quality evaluation metrics and mAP on MultiviewUnderwater. The histogram represents the mAP and the polyline represents different image quality evaluation metrics. Numbers 1 to 13 refer to thirteen UIE algorithms ordered according to increasing mAP values.

reference metrics, while other methods only work for some special image quality evaluation metrics.

Fig. 3.23 and Fig. 3.24 illustrate mAP and image quality evaluation metrics on ChinaMM and MultiviewUnderwater datasets, from which we investigate how the detection accuracy is related to different image quality evaluation metrics. There is no strong correlation between the mAP and the image quality evaluation metrics on the two datasets. For MultiviewUnderwater, Regression receives the best MSE and PCQI scores among the six physical model based methods, however, its detection accuracy is the worst among these methods. For ChinaMM, CycleGAN can greatly improve the UCIQE and UIQM scores, but its mAP (67.8%) is even lower than that of the raw underwater images (68.4%). On the OUC dataset, both GDCP and UDCP decrease the MSE and PSNR scores of the raw underwater images, but their mAP are even higher than these of the high quality reference images (86.6% mAP). Therefore, certain discrepancies exist between the image quality evaluation metrics and the detection accuracy. mAP may be biased to some image quality metrics such as high UICONM and UICM. For example, Blurriness, Histogram, Fusion and Retinex have top four UICONM scores, and achieve the top four detection accuracy among the non-deep learning based methods. OursPatch and OursOF receive significantly higher UICM scores and rank top two in terms of mAP. The detection task tends to favour the results with high contrasts (high UICONM) between the objects and the background, or that with the over-enhanced objects. One possible explanation lies in that high contrast suppresses the complicated background while bright color protrudes the objects. For illustration, we show the object detection results of two underwater images from MultiviewUnderwater and ChinaMM in Fig. 3.25 and Fig. 3.26.

We believe there is a gap between the image quality evaluation metrics for the low-level enhancement task and the accuracy metric for the high-level detection task. Underwater image enhancement task is usually evaluated using the image quality evaluation metrics. However, the objective of the low-level enhancement task typically differs from that of the high-level object detection task so that the enhancement algorithm can hardly recover

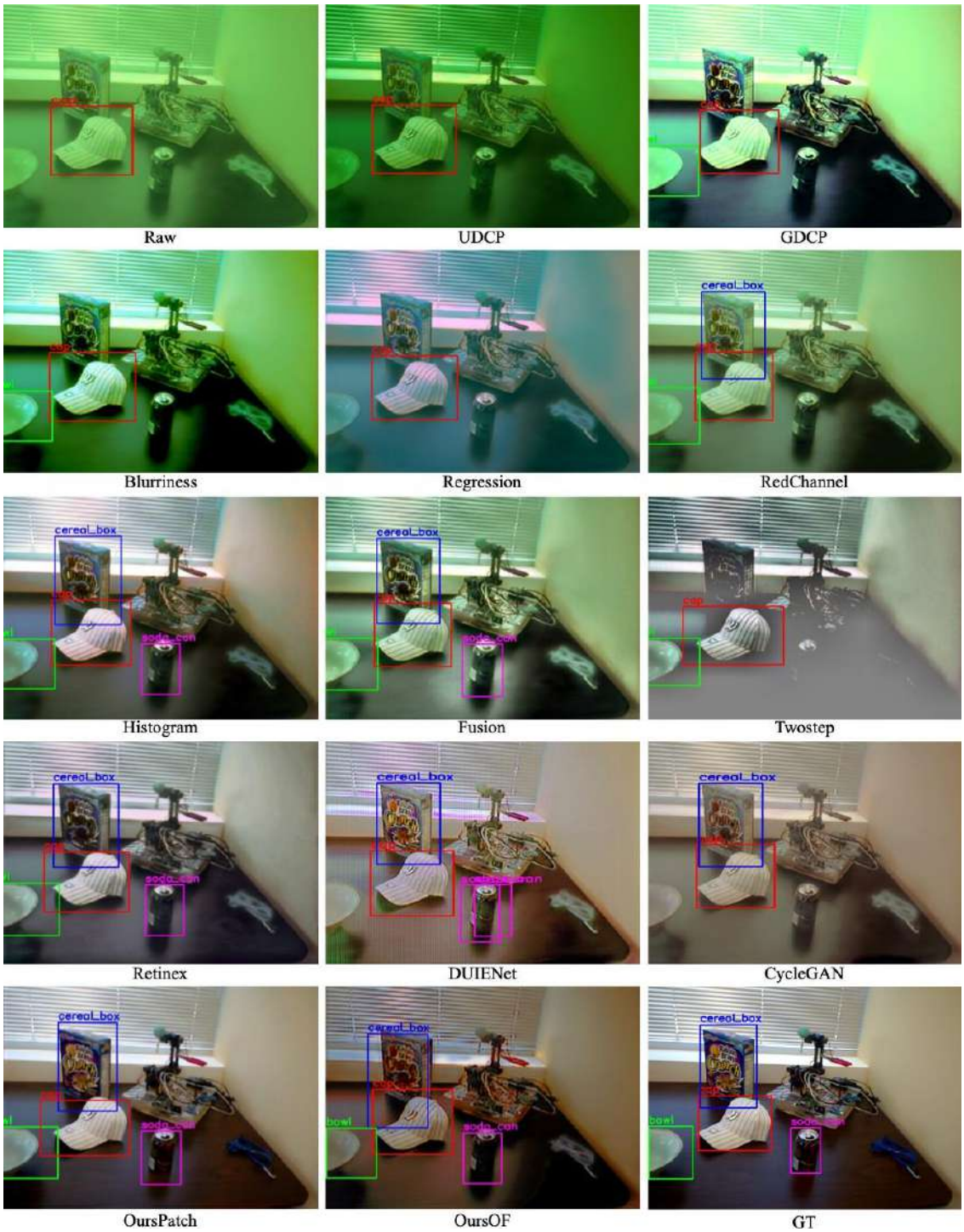

Fig. 3.25 Visualization of object detection results after having applied different UIE algorithms on MultiviewUnderwater.

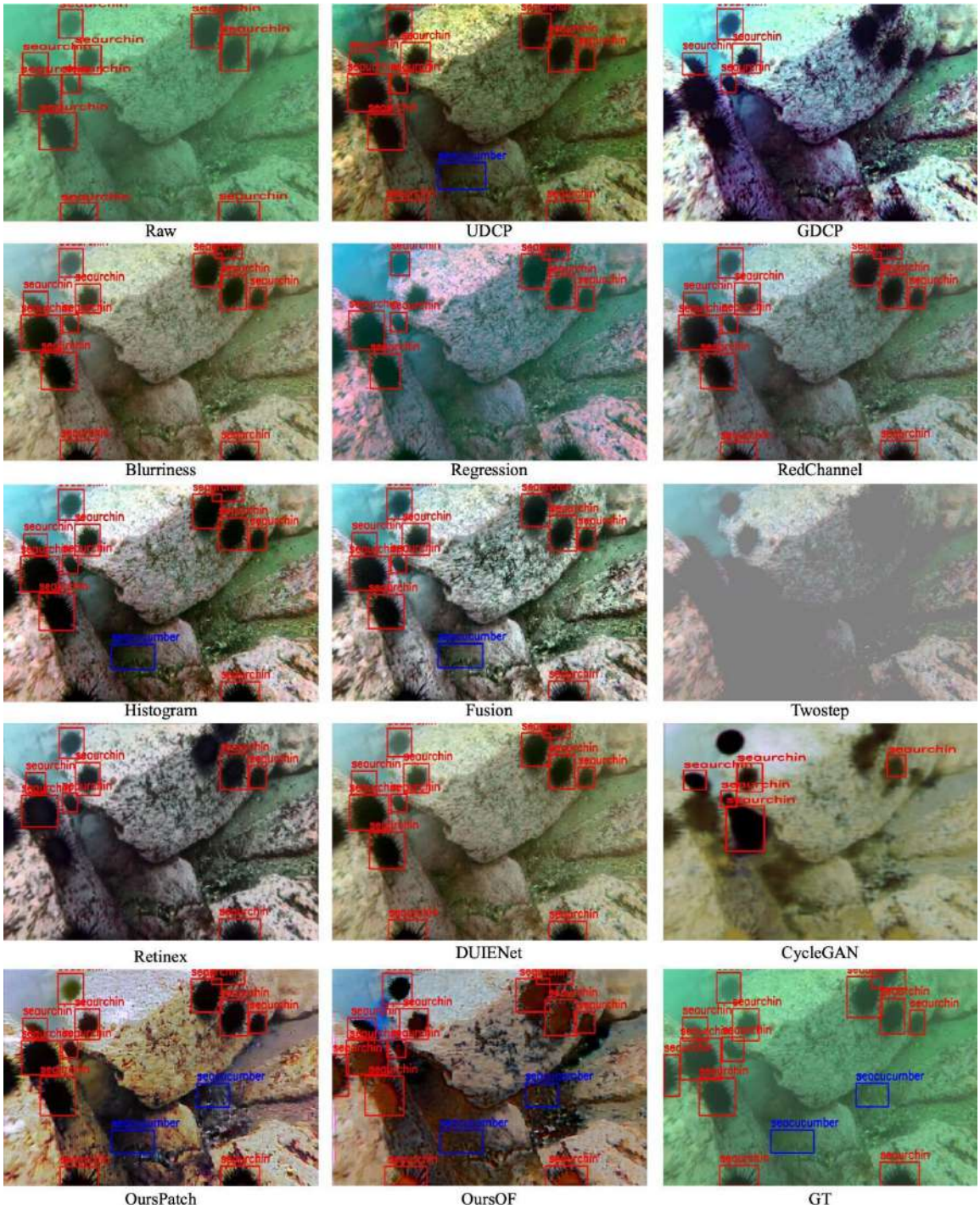

Fig. 3.26 Visualization of object detection results after having applied different UIE algorithms on ChinaMM.

features favoured by the high-level detection task. The interaction between the enhancement and detection tasks are important indeed.

3.5 Summary

In this chapter, we focus on addressing the underwater image enhancement task, and have proposed two detection-perceptual enhancement models, i.e., patch detection-based perceptual enhancement and object-focused detection based perceptual enhancement models. Different from previous works, we designed the novel detection perceptors for the enhancement models. With the help of the detection perceptors, our patch detection-based perceptual enhancement model can generate high quality in-air images with patch-level details, and our object-focused detection-based perceptual enhancement model can generate images which improves the detection accuracy of the following deep detectors. Moreover, to advance the generalisation ability of deep UIE algorithms, we have proposed a hybrid underwater image synthesis model to synthesise more realistic training images, and the enhancement model trained on them can learn more robust translation between the underwater and high quality in-air images, and generalise well on the real-world underwater scenes.

Chapter 4

Underwater Object Detection in Noisy Datasets

4.1 Introduction

Deep learning based object detection systems have demonstrated promising performance in various applications but still felt short of dealing with underwater object detection. This is because, firstly, underwater detection datasets are scarce and the objects in the available underwater datasets and real applications are usually small. Current deep learning based detectors cannot effectively detect small objects (see an example shown on top row of Fig. 4.1). Secondly, the images in the existing underwater datasets and real applications accompany considerable noisy data. In the underwater scenes, wavelength-dependent absorption and scattering [132] cause serious visibility loss, contrast decrease and color distortion, generating considerable noisy data. The noisy data exaggerates the challenge of inter-class similarity exist in object detection [133, 134], resulting in the confusion between the object classes and the background class. As shown on the bottom row of Fig. 4.1, the proposed SWIPENET trained on the noisy data cannot distinguish between the background and the objects.

In this chapter, we propose a deep ensemble detector which is effective in dealing with small objects and noisy data in the underwater scenes. To achieve the objectives, we propose a deep backbone network named Sample-WeIghted hyPEr Network (SWIPENET), which

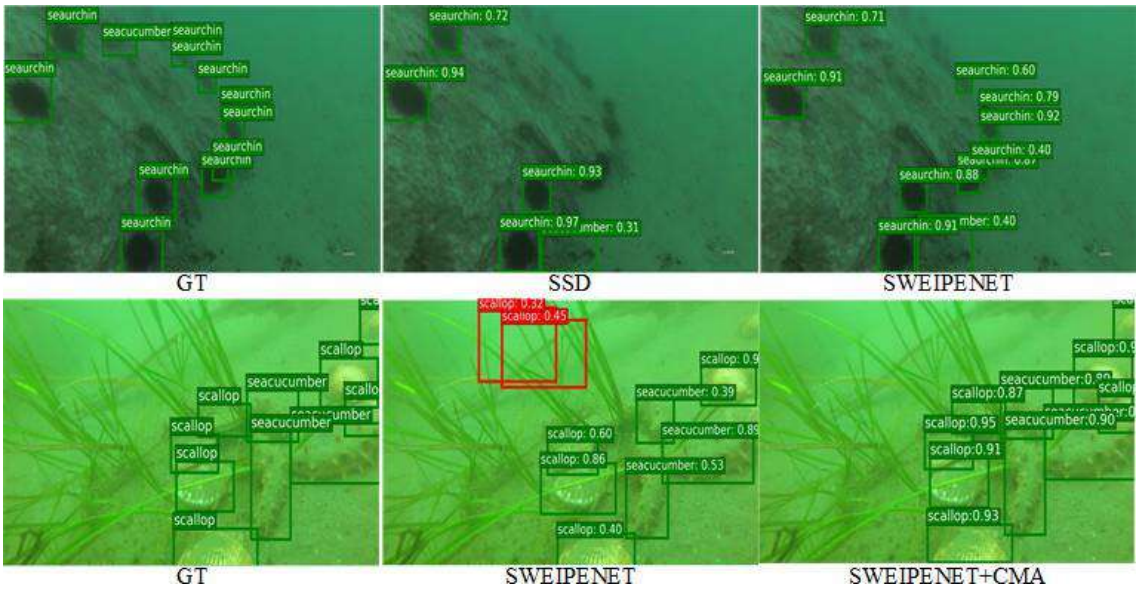

Fig. 4.1 Exemplar images with ground truth (GT) annotations, results of Single Shot MultiBox Detector (SSD) [112], our proposed SWIPENET and SWIPENET+CMA. The top row shows that SSD cannot detect all the small objects while our proposed SWIPENET outperforms SSD in this case. The bottom row shows our proposed SWIPENET treats the background as objects due to the existence of noisy data while our proposed SWIPENET+CMA performs better than the others.

fully takes advantage of multiple Hyper Feature Maps. To address the noisy data problem, we propose a novel sample-weighted detection loss function and a novel noise-robust training paradigm named Curriculum Multi-Class Adaboost (CMA), used to train the deep ensemble for underwater object detection. Indeed, the sample-weighted detection loss is used to control the influence of the training samples on SWIPENET. It works with the training paradigm CMA to train the proposed deep ensemble detector to reduce errors.

The proposed CMA training paradigm is inspired by the idea in the human education system that starts from learning easy tasks, and then gradually increase learning difficulty levels. This learning concept has been utilised to improve the generalisation ability and accelerate convergence in machine learning algorithms [135–137]. For example, Derenyi et al. [137] reported theoretical analysis where easy examples should be learnt first due to less noise. They treat the samples misclassified by the Bayesian classifier as noisy data and learn the easier samples first, then improve convergence and the generalisation ability. Motivated by these works, our CMA training paradigm consists of two training stages: Noise-eliminating (NECMA) and noise-learning (NLCMA) stages. In the noise-eliminating stage, a 'clean' detector (SWIPENET) of being free from the influence of noisy data is formulated by focusing on learning easy samples whilst ignoring learning hard and noisy data. Then, the previously learnt knowledge by the 'clean' detector is again used to ease the training of the detectors in the noise-learning stage which focuses on learning diverse noisy data. The parameters of the detectors in the noise-learning stage are initialised by those of the 'clean' detector, which help the deep detectors avoiding poor local optimum during training and improving the convergence speed and system generalisation. Finally, to achieve a balance between running time and detection accuracy, we present a selective ensemble algorithm to choose several detectors with a large diversity for the final ensemble.

4.2 Proposed SWIPENET+CMA Framework

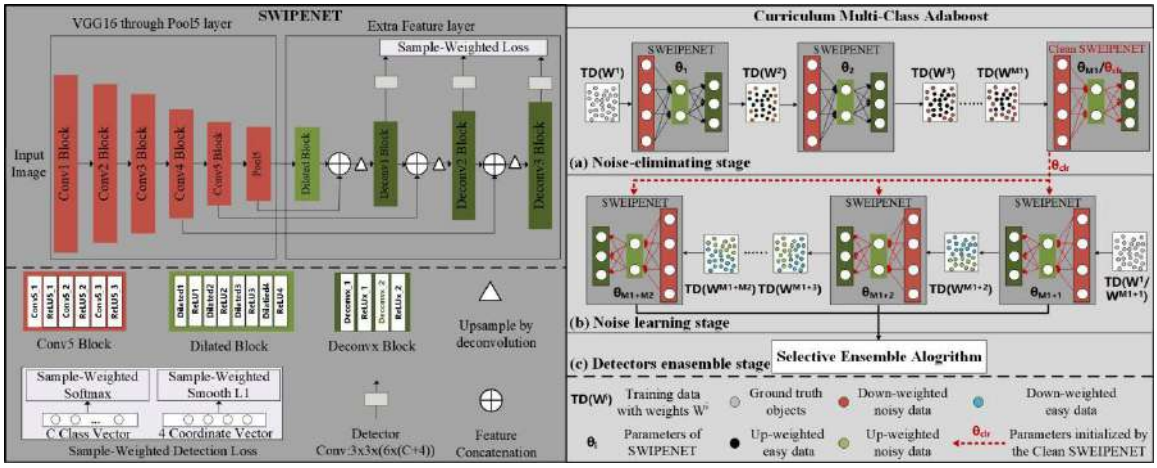

Fig. 4.2 The overview of our proposed SWIPENET+CMA detection framework. The left shows the structure of our proposed SWIPENET backbone, the right shows the CMA training paradigm that consist of the (a) Noise-eliminating stage (NECMA): gradually reduce the weights of the possible noisy data to obtain a 'clean' detector which is free from the influence of the noisy data. (b) Noise learning stage (NLCMA): learn diverse noisy samples by increasing their weights to boost the generalisation ability. The parameters of each detector in NLCMA are initialised by those of the 'clean' detector, that alleviates the local optimum problem and accelerate the convergence, and (c) Detectors ensemble stage: ensemble multiple detectors to boost the generalisation ability.

4.2.1 Sample-WeIghted hyPEr Network (SWIPENET)

Evidence shows that the down-sampling excises of Convolutional Neural Network result in strong semantics that lead to the success of classification tasks. However, this is not enough for the object detection task which not only needs to recognise the objects but also spatially locates its position. After we have applied several down-sampling operations, the spatial resolutions of the deep layers are too coarse to handle small object detection.

In this section, we propose the SWIPENET architecture that includes several high resolution and semantic-rich Hyper Feature Maps inspired by Deconvolutional Single Shot Detector (DSSD) [138], in which multiple down-sampling convolution layers are first constructed to extract high semantic feature maps. After several down-sampling operations, the feature maps are too coarse to provide sufficient information for accurate small object localization, therefore, multiple up-sampling deconvolution layers and skip connection are added to recover the high resolutions of the feature maps. However, the detailed information lost in

the down-sampling operations cannot be fully recovered even though the resolutions have been recovered. Different from DSSD, we design a dilated convolution block in SWIPENET to obtain large receptive fields without sacrificing detailed information that support object localization (large receptive fields lead to strong semantics [139, 140]). The proposed dilated convolution block consists of 4 dilated convolution layers with ReLU activation. Denote the input and output of i-th dilated convolution layer as F_I^i and F_O^i. The input of dilated block is the feature maps from the Pool5 layer (F_{P5}) and the output of dilated block F_O^4 is implemented using the following steps: Denote the i-th convolution kernel as $\vartheta_i \in \mathrm{R}^{K_i \times K_i \times C_i}$ (K_i is the kernel size and C_i is the channel number). Different from convolution operation (denote as $*$), dilated convolution operation first produce a dilated kernel ϑ_i^d by dilating the convolution kernel using the dilation function $F_D(.)$ parametrized with a dilation rate d_i.

$$\vartheta_i^d = F_D(\vartheta_i, d_i) \in \mathrm{R}^{K_i^d \times K_i^d \times C_i} \tag{4.1}$$

where

$$K_i^d = K_i + (K_i - 1) \times (d - 1) \tag{4.2}$$

$F_D(\vartheta_i, d_i)$ indicates insert d_i-1 zeros between neighbour values in each channel of ϑ_i. Then, the dilated kernel is applied to convolution on the input. Finally, ReLU activation function $F_R(.)$ is applied to the dilated convolution layer and produce the output of each dilated convolution layer.

$$F_O^i = F_R(F_I^i * F_D(\vartheta_i, d_i)), i = 1, 2, 3, 4 \tag{4.3}$$

where

$$F_I^i = \begin{cases} F_{P5} & \textbf{if } i = 1 \\ F_O^{i-1} & \textbf{otherwise} \end{cases} \tag{4.4}$$

Fig. 4.2 illustrates the overview of our proposed SWIPENET, which consists of multiple convolution blocks, a novel dilated convolution block, multiple deconvolution blocks and a novel sample-weighted loss. The front layers of the SWIPENET are based on the architecture of the standard VGG16 model [141] (truncated at the Pool5 layer). Then, we add the proposed dilated convolution block to extract high semantic while keep the resolutions of the feature

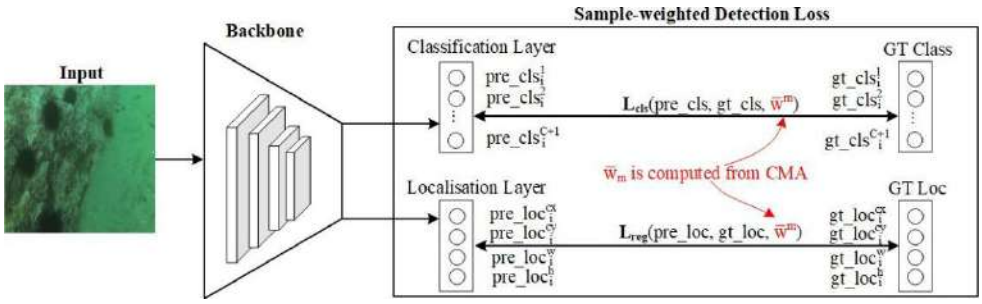

Fig. 4.3 The detailed explanation of sample-weighted detection loss.

maps. Finally, we up-sample the feature maps using deconvolution and add skip connection to construct multiple Hyper Feature Maps on the deconvolution layers. The prediction of SWIPENET deploys three different deconvolution layers, i.e. Deconv1_2, Deconv2_2 and Deconv3_2 (denoted as Deconvx_2 in Fig. 4.2), which increase in size progressively and allow us to predict the objects of multiple scales. At each location of the three deconvolution layers, we define 6 default boxes and use a 3×3 convolution kernel to produce $C + 1$-D class prediction (C indicates the number of the object classes and 1 indicates the background class) and 4-D coordinate prediction.

4.2.2 Sample-Weighted Detection Loss

We propose a novel sample-weighted detection loss function which can model sample weights in SWIPENET. The sample-weighted detection loss enables SWIPENET to focus on learning high weight samples whilst ignoring low weight samples. It cooperates with a novel sample re-weighting algorithm, namely Curriculum Multi-Class Adaboost, to reduce the influence of possible noise on SWIPENET by decreasing their weights.

Following the one-stage deep detector SSD [112], SWIPENET trains an object detector using default boxes on several layers. If the Intersection over Union (IoU) between the default box and its most overlapped object is larger than a pre-defined threshold, then the default box is a match to this object that works as its positive training sample. If a default box does not match any object, it will be regarded as a negative training sample. Technically, our sample-weighted detection loss L consists of a sample-weighted softmax loss L_{cls} for the

bounding box classification and a sample-weighted smooth L1 loss L_{reg} for the bounding box regression:

$$L = \frac{\alpha_1}{\ddot{N}} L_{cls}(pre\_cls, gt\_cls, \bar{w}) + \frac{\alpha_2}{\bar{N}} L_{reg}(pre\_loc, gt\_loc, \bar{w}) \tag{4.5}$$

where $\ddot{N}$ and $\bar{N}$ are the numbers of all the training samples and positive training samples respectively, α_1 and α_2 denote the weight terms of classification and regression losses. The sample-weighted softmax loss L_{cls} is formulated as

$$L_{cls} = -\sum_{i=1}^{\ddot{N}} \sum_{c=1}^{C+1} \bar{w}_i^m gt\_cls_i^c log(pre\_cls_i^c) \tag{4.6}$$

$$pre\_cls_i^c = \frac{e^{net_i^c}}{\sum_{c^-=1}^{C+1} e^{net_i^{c^-}}} \tag{4.7}$$

where $\bar{w}_i^m$ denotes the sample weight for the i-th sample computed in the m-th iteration of CMA by Eq. (4.12). Denote pre_cls_i and gt_cls_i as the predicted and ground truth class vectors for the i-th sample, these two vectors are $C+1$-D vectors (C object classes plus one background class). $pre\_cls_i^c$ and $gt\_cls_i^c$ denote the c-th element of the predicted and ground truth class vectors for the i-th sample (referring to Fig. 4.3 for better understanding). $gt\_cls_i^c = 1$ if the i-th sample belongs to the c-th class, $gt\_cls_i^c = 0$ otherwise. net_i^c is the classification prediction from the detection network. L_{reg} is the sample-weighted smooth L1 loss, formulated as follows:

$$L_{reg} = \sum_{i=1}^{\bar{N}} \sum_{l \in Loc} \bar{w}_i^m Smooth_{L_1}(pre\_loc_i^l - gt\_loc_j^l) \tag{4.8}$$

$$Smooth_{L1}(x) = \begin{cases} 0.5x^2 & \textbf{if } |x| < 1 \\ |x| - 0.5 & \textbf{otherwise} \end{cases} \tag{4.9}$$

$$pre\_loc_i^l = net_i^l, l \in Loc \tag{4.10}$$

pre_loc_i and gt_loc_i denote the predicted and ground truth coordinate vectors for the i-th sample, these two vectors are 4-D vectors (the coordinate information $Loc = (cx, cy, w, h)$ includes the coordinates of center (cx, cy) with width w and height h. $pre\_loc_i^l$ and $gt\_loc_i^l$ denote the l-th element of the predicted and the ground truth coordinate vectors for the i-th positive training sample respectively. net_i^l is the coordinate prediction from the detection network.

In the gradient based optimisation algorithm, the loss function plays a key role in providing the gradients for updating the model parameters in the back-propagation process. The sample's gradient magnitude in the derivative of the loss function determines its impact on the updating of the DNNs. In our proposed sample-weighted detection loss, the sample weight $\bar{w}$ is able to adjust the sample's gradient magnitude. Hence, we are able to investigate how the sample weight influences the sample's impact on the feature learning of DNNs. Denote the parameter of the detector as ϑ, the derivative of the sample-weighted detection loss $\frac{\partial L}{\partial \vartheta}$ is derived as:

$$\frac{\partial L}{\partial \vartheta} = \begin{cases} \alpha_1 \frac{1}{N^{+}} \sum_{i=1}^{N^{+}} \sum_{c=1}^{C+1} \bar{w}_i^m gt\_cls_i^c (pre\_cls_i^c - 1) \frac{\partial net_i^c}{\partial \vartheta} \\ \quad + \frac{\alpha_2}{N} \sum_{i=1}^{N} \sum_{l \in Loc} \bar{w}_i^m (pre\_loc_i^l - gt\_loc_j^l) \frac{\partial net_i^l}{\partial \vartheta} & \textbf{if } |pre\_loc_i^l - gt\_loc_i^l| < 1 \\ \alpha_1 \frac{1}{N^{+}} \sum_{i=1}^{N^{+}} \sum_{c=1}^{C+1} \bar{w}_i^m gt\_cls_i^c (pre\_cls_i^c - 1) \frac{\partial net_i^c}{\partial \vartheta} \\ \quad + \frac{\alpha_2}{N} \sum_{i=1}^{N^{+}} \sum_{l \in Loc} \bar{w}_i^m \frac{\partial net_i^l}{\partial \vartheta} & \textbf{otherwise} \end{cases} \tag{4.11}$$

From Eq. (4.11), we witness that the sample's gradient magnitude in the derivative is influenced by two factors. The first one is the accuracy of the predicted class and coordinates. For the i-th training sample with ground truth class c (i.e., $gt\_cls_i^c = 1$), the closer $pre\_cls_i^c$ and $pre\_loc_i^c$ to the ground truth, the smaller the gradient magnitude for the i-th sample. Second, the sample's weight $\bar{w}_i^m$. Suppose all of the training samples have the same prediction accuracy. The smaller the weight is, the smaller gradient magnitude is attached to the i-th sample. For example, if we assign a weight of 100 and 1 to the same positive sample respectively, then the gradient magnitude of the former one may be around 100 times that of the later one. The feature learning of DNN is dominated by high-weight samples while

the low-weight samples contribute far less to the update of the DNN features. Hence, the sample-weighted detection loss less counts on the low-weight samples.

4.2.3 Curriculum Multi-class Adaboost (CMA)

Underwater images suffer from the degradation of different levels, e.g. poor lighting, noise and blurs. If we train a detector on the dataset containing severely deteriorated images, the 'noisy data' are easy to confuse the object detector. Such example is illustrated in the bottom of Fig. 4.1.

Inspired by the human education system that learns from easy to hard samples, we here propose a noise-immune training paradigm, namely Curriculum Multi-class Adaboost (CMA), to train multiple deep detectors and then ensemble them into a unified model for underwater object detection in the underwater scenes with the data of considerable noise and large diversity. This strategy helps accelerate the convergence and improve the generalisation of the proposd architecture because the detector trained on easy data provides optimum initialisation for the following deep detectors. Good initialisation helps the proposed model to avoid the local optimum problem in training and to improve generalisation, which has been demonstrated in previous work [135–137].

4.2.3.1 The Overview of the CMA

CMA is developed based on Multi-Class Adaboost (MA) [57], which trains multiple base classifiers sequentially and assign a weight value α_m to each classifier according to its error rate E_m. When training each classifier, the samples misclassified by the preceding classifier are assigned a higher weight, allowing the following classifier to focus on learning these samples. Finally, all the weak base classifiers are combined to form an ensemble classifier with corresponding weight values.

Different form MA, our proposed CMA algorithm consists of two stages: noise-eliminating (denotes as NECMA) and noise-learning stages (denotes as NLCMA). In each training iteration of NECMA, we reduce the weights of the undetected objects as they are likely to be noisy data [137]. The sample-weighted detection loss enables the next SWIPENET to

only focus on learning the high-weight clean data. By gradually reducing the influence of the noisy data, the generalisation capability of the system is improved and a detector free from the influence of the noisy data is sought. However, after several iterations, the deep detector may over-fit over the clean, easy samples as their weights are too high after several rounds of re-weighting exercises, and the generalisation ability becomes deteriorated. Therefore, we terminate the noise-eliminating stage when the performance does not improve anymore, and the detector achieving the best detection accuracy is selected as the 'clean' detector.

The 'clean' detector can detect the clean, easy objects well but always fails to detect many hard objects. This is because although the undetected objects tend to be noisy data or outliers, but they also contain many hard object instances which haven't been detected because they are very similar with the backgrounds. Ignoring these hard object instances will limit the network's generalization ability on the hard object instance. Hence, we propose the NLCMA training stage, which focuses on learning diverse hard samples by increasing their weights. In practice, the parameters of each detector in NLCMA are initialised by those of the 'clean' detector. This strategy effectively alleviates the local optimum problem and significantly accelerate the convergence whilst boosting the generalisation ability.

The proposed CMA training paradigm can be found in Algorithm 1. It iteratively trains M detectors, including M_1 iterations for NECMA and M_2 iterations for NLCMA. We assume the best performing detector (i.e, the 'clean' detector S_{clr} parameterised by ϑ_{clr}) in NECMA is achieved in the M_1-th iteration, M_1 is experimentally obtained. Denote I_{train} as the training images with the ground truth objects $B = \{b_1, b_2, \ldots, b_N\}$, N is the number of the objects in the training set, $b_j = (cls, cx, cy, w, h)$ is the annotation of the j-th object. We denote w_j^m as the weight of the j-th object in the m-th iteration. Each object's weight is initialised to $\frac{1}{N}$ in the first iteration, i.e. $w_j^1 = \frac{1}{N}, j = 1, \ldots, N$.

In the m-th iteration of CMA, we firstly compute the weights of the positive training samples. If the i-th positive sample matches the j-th object during the training, we compute the i-th positive sample's weight $\bar{w}_i^m$ using Eq. (4.12).

$$\bar{w}_i^m = N * w_j^m, \; 0 < w_j^m < 1 \tag{4.12}$$

Algorithm 1 Noise-immune CMA training paradigm.

Input: Training images I_{train} with ground truth objects $B = \{b_1, \ldots, b_N\}$, testing images I_{test}.
Output: M SWIPENETs.

Initialise the object weights $w_j^1 = \frac{1}{N}, j = 1, \ldots, N$.
for $m = 1$ to M_1 **do**
    ·Compute the weights of positive samples using Eq. (8).
    ·Train the m-th SWIPENET G_m using Eq. (1).
    ·Compute the m-th SWIPENET's error rate E_m using Eqs. (9)-(10).
    ·Compute the m-th SWIPENET's weight α_m in the ensemble model using Eq. (11).
    ·Reduce the weights of the undetected objects and increase the weights of the detected objects using Eq. (12)).
end for
Obtain the parameter ϑ_{clr} of the M_1-th SWIPENET.
Initialize the object weights $w_j^{M_1+1} = \frac{1}{N}, j = 1, \ldots, N$.
for $m = M_1 + 1$ to M_2 **do**
    ·Compute the weights of positive samples using Eq. (8).
    ·Initialize the parameter of the m-th SWIPENET G_m using ϑ_{base}.
    ·Train the m-th SWIPENET G_m using Eq. (1).
    ·Compute the m-th SWIPENET's error rate E_m using Eqs. (9)-(10).
    ·Compute the m-th SWIPENET's weight α_m in the ensemble model using Eq. (11).
    ·Increase the weights of the undetected objects and decrease the weights of the detected objects using Eq. (13).
end for
return M SWIPENETs.

where w_j^m denotes the weight of the j-th object in the m-th iteration. The weight of the positive sample is N times that of its matched object. This is because the initial weight of each object in CMA is $\frac{1}{N}$ and the initial weight of each positive training sample in the sample-weighted detection loss is 1. Secondly, we use the re-weighted samples to train the m-th detector S_m. Thirdly, we run the m-th detector on the training set and receive the detection results $D_m = \{d_1, d_2, \ldots, d_i\}$ while $d_i = (cls, score, cx, xy, w, h)$ is the i-th predicted outcome, including the predicted class (cls), score ($score$) and coordinates (cx, cy, w, h). The error rate E_m of the m-th detector is computed based on the percentage of the undetected objects.

$$E_m = \sum_{j=1}^{N} w_j^m I(b_j) / \sum_{j=1}^{N} w_j^m \tag{4.13}$$

where

$$I(b_j) = \begin{cases} 0 \text{ if } \exists d \in D_m, s.t. b_j.cls == d.cls \wedge IoU(b_j, d) \geq \vartheta \\ 1 \text{ otherwise} \end{cases} \tag{4.14}$$

In Eq. (4.14), if there exists a detection d which belongs to the same class as the j-th ground truth object b_j (i.e. $b_j.cls == d.cls$) and the Intersection over Union (IoU) between the detection and the j-th object is larger than the threshold ϑ (0.5 here), we set $I(b_j) = 0$, indicating the j-th object has been detected and $I(b_j) = 1$ is the undetected. Fourthly, we compute the m-th detector's weight α_m using Eq. (4.15), which is used when we ensemble different detectors.

$$\alpha_m = log\frac{1 - E_m}{E_m} + log(C - 1) \tag{4.15}$$

$$w_j^m \leftarrow \frac{w_j^m}{z_m} exp(\alpha_m(1 - I(b_j))) \tag{4.16}$$

where C is the number of the object classes. Finally, we update each object's weight w_j^m and train the following detector. In the first M_1 iterations of NECMA stage, we reduce the weights of the undetected objects by Eq. (4.16) that enables the next detector to pay less attention to possible noisy data. In the last M_2 iterations of the NLCMA stage, we increase the weights of the undetected objects by Eq. (4.17), whereas the detector turns to learning

the diverse hard data. z_m is a normalisation constant. The same iteration repeats again till all M detectors have been trained.

$$w_j^m \leftarrow \frac{w_j^m}{z_m} exp(\alpha_m I(b_j)) \tag{4.17}$$

It is noticed that when CMA changes from NECMA to NLCMA, i.e., in the $M_1 + 1$-th iteration, we must re-initialise the weight of each object as $\frac{1}{N}$. In each iteration of NLCMA, we initialise the parameter of each detector with the parameter ϑ_{clr} of the 'clean' SWIPENET. These initialisations help the system avoid local maximum (or minimum) problem, whilst efficiently converging to stationary points.

4.2.3.2 Selective Ensemble Algorithm

An ensemble model may be more accurate than a single model, but brings in additional computational overhead. Recent references [142–144] have pointed out that the ensemble of selective deep models may not only be more compact but also stronger in the generalization ability than that of the overall deep models. To reduce the computational costs, we first select a few detectors with large diversity. If the results of two different detectors look similar, the ensemble model based on the two detectors does not have the complementary ability. We need to determine which detector is used and how many detectors are incorporated in the final ensemble model.

We here propose a greedy selection algorithm to select candidate detectors for the final ensemble. Firstly, we construct a candidate ensemble set E to add up the selected detectors, and initialise it with the detector achieving the highest detection accuracy among all the M_2 detectors in NLCMA as these detectors have not been confused by noisy data. Then, we gradually select a single detector D_{m*} having the largest diversity with all the detectors in the candidate ensemble set and add it to the ensemble set, as formulated in Eq. (4.18).

$$D_{m*} = \underset{m, D_m \notin E}{\text{argmax}} \sum_{D_n \in E} Q_{mn} \tag{4.18}$$

Here, we apply the commonly used Q statistic [145] to measuring the diversity of two detectors' performance.

$$Q_{mn} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \tag{4.19}$$

Q_{mn} denotes the diversity between the performance of detectors D_m and D_n. N^{11} and N^{00} are the numbers of the objects detected and missed by the two detectors respectively. N^{01} is the total number of the objects missed by D_m and detected by D_n, N^{10} is the total number of the objects detected by D_m and missed by D_n. Maximum diversity is achieved at $Q_{mn} = -1$ when the two detectors make different predictions (i.e., $N^{11} = N^{00} = 0$), and the minimum diversity is achieved at $Q_{mn} = 1$ when the two detectors generate identical predictions (i.e., $N^{01} = N^{10} = 0$).

After all the candidate detectors have been selected, we ensemble them into a unified ensemble detector according to their weights computed by Eq. (4.15) in CMA and their diversity weight in the ensemble set. We assign a higher weight to the detector with a larger diversity. This enables the ensemble detector to detect diverse objects in the underwater scenes, where a large diversity exists due to the changed illuminations, water depth and object-camera distance. We compute the diversity weight div_m of detector D_m as its average diversity with all the detectors in the ensemble set (by Eq. (4.20)).

$$div_m = \sum_{D_n \in E, n \neq m} Q^*_{mn} / (|E| - 1) \tag{4.20}$$

The value of Q_{mn} lies in [-1,1]. For better representing the weights of the detection model, we normalise Q_{mn} as Q^*_{mn} using Eq. (4.21). The value of Q^*_{mn} lies in [0,1], and the larger diversity the large value of the diversity weight.

$$Q^*_{mn} = 0.5(1 - Q_{mn}) \tag{4.21}$$

The final weight λ_i of detector D_i is formulated as

$$\lambda_i = \frac{div_i * \alpha_i}{\sum_{m=1}^{M^*} div_m * \alpha_m} M^*, i = 1, \ldots, M^* \tag{4.22}$$

In the testing stage, we use the weights to re-score the detection boxes. M^* denotes the number of the selected detectors, and $M^*/\sum_{m=1}^{M^*} div_m * \alpha_m$ in Eq. (4.22) is a normalisation term, scaling the score of the box to fall in [0,1] after re-scoring. In particular, we first run all M^* selected SWIPENETs on the testing set I_{test} and produce a M^* detection set Det_m.

$$Det_m = D_m(I_{test}),\ m = 1, 2, \ldots, M^* \tag{4.23}$$

Afterwards, we re-score each detection d in Det_m using λ_m.

$$d.score = \lambda_m d.score, d \in Det_m \tag{4.24}$$

Finally, we combine all the detections and utilise Non-Maximum Suppression [146] to remove the overlapped detections by Eq. (4.25), achieving the final detection results Det.

$$Det = NonMaximumSuppression(\bigcup_{m=1}^{M^*} Det_m) \tag{4.25}$$

4.3 Experiments Setup

To demonstrate the effectiveness of the proposed method, we conduct comprehensive evaluations on four underwater datasets URPC2017, URPC2018, URPC2019 and ChinaMM [9]. The former three datasets come from the Underwater Robot Picking Contest. The underwater robot picking contest datasets were generated by National Natural Science Foundation of China and Dalian Municipal People's Government. The Chinese website is http://www.cnurpc.org/index.html and the English website is http://en.cnurpc.org/. The contest holds annually from 2017, consisting of online and offline object detection contests. In this paper, we use URPC2017, URPC2018 and URPC2019 datasets from the online object detection contest. To use the datasets, participants need to communicate with zhuming@dlut.edu.cn and sign a commitment letter for data usage: http://www.cnurpc.org/a/js/2018/0914/102.html. In this section, we first introduce the experimental datasets. Then, we describe the implementation details.

4.3.1 Datasets

The URPC2017 and ChinaMM datasets have 3 object categories, including seacucumber, seaurchin and scallop. URPC2017 contains 18,982 training images and 983 testing images. ChinaMM contains 2,071 training images and 676 validation images. The URPC2018 and URPC2019 datasets have 4 object categories, including seacucumber, seaurchin, scallop and starfish. URPC2018 and URPC2019 have published the training set, but the testing set is not publicly available. Hence, we randomly split the training set of URPC2018 into 1,999 training images and 898 testing images, and split the training set of URPC2019 into 3,409 training images and 1,000 testing images. All four datasets provide underwater images and box level annotations.

4.3.2 Implementation Details

All the experiments are conducted on a server with an Intel Xeon CPU @ 2.40GHz and a single Nvidia Tesla P100 GPUs with a 16 GB memory. For our proposed detection framework, we implement it using the Keras framework, and train it with the Adam optimisation algorithm. We use an image scale of 512x512 as the input for both training and testing. On URPC2017, the batch-size is 16, and the learning rate is 0.0001. Our models often diverge when we use a high learning rate due to unstable gradients, and all the detectors in the ensemble achieve the best performance after running 120 epochs. On URPC2018 and URPC2019, the batch-size is 16. We first train each detector in the ensemble with a learning rate 0.001 for 80 epochs, and then train them with a learning rate 0.0001 for another 40 epochs. On ChinaMM, the batch-size is 16, and the learning rate is 0.001. Each detector in the ensemble runs 120 epochs. The source code will be made available at:https://github.com/LongChenCV/SWIPENET+CMA.

Table 4.1 mAP indicates mean Average Precision(%).

Dataset	Network	Skip	Dilation	mAP
URPC2017	UWNET1		✓	40.4
	UWNET2			38.3
	SWIPENET	✓	✓	42.1
URPC2018	UWNET1		✓	61.2
	UWNET2			58.1
	SWIPENET	✓	✓	62.2
URPC2019	UWNET1		✓	55.0
	UWNET2			54.2
	SWIPENET	✓	✓	57.6
ChinaMM	UWNET1		✓	73.9
	UWNET2			71.0
	SWIPENET	✓	✓	76.1

4.4 Ablation Studies

In this section, we conduct the ablation experiments to investigate the influence of different components on our SWIPENET+CMA framework, including the skip connection, the dilated convolution block and the CMA training paradigm. In the next section, we compare our method against several state-of-the-art (SOAT) detection frameworks on four datasets.

4.4.1 Ablation Studies on the Skip Connection and Dilated Convolution

To investigate the influence of skip connection, we design the first baseline network UWNET1 which has the same structure as SWIPENET except that it does not contain skip connection between the low and high layers. The second network UWNET2 replaces the dilated convolution block in UWNET1 with standard convolution block to learn the influence of the dilated convolution block. Table 4.1 shows the performance comparison of different networks on four datasets, we observe that SWIPENET performs better than UWNET1. The gains come from the skip connection which passes fine detailed information of the lower layers such as object boundary to the high layers that are important for object localisation. Compared to UWNET2, UWNET1 performs better because the dilated convolution block in UWNET1 brings much semantic information to the high layers which enhances the

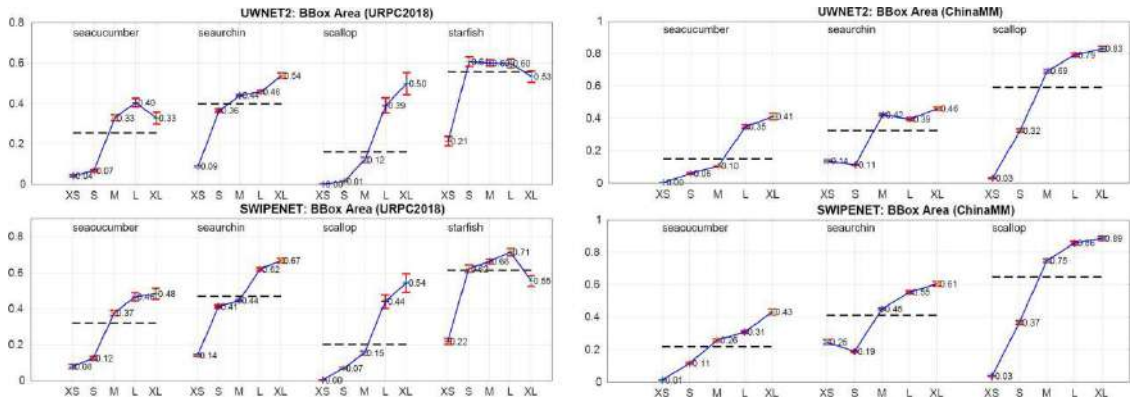

Fig. 4.4 The mean Average Precision of UWNET2 and SWIPENET for objects with different object sizes on URPC2018 and ChinaMM. The object size is measured as the pixel area of the bounding box. XS (bottom 10%)=extra-small; S (next 20%)=small; M (next 40%)=medium; L (next 20%)=large; XL (next 10%)=extra-large.

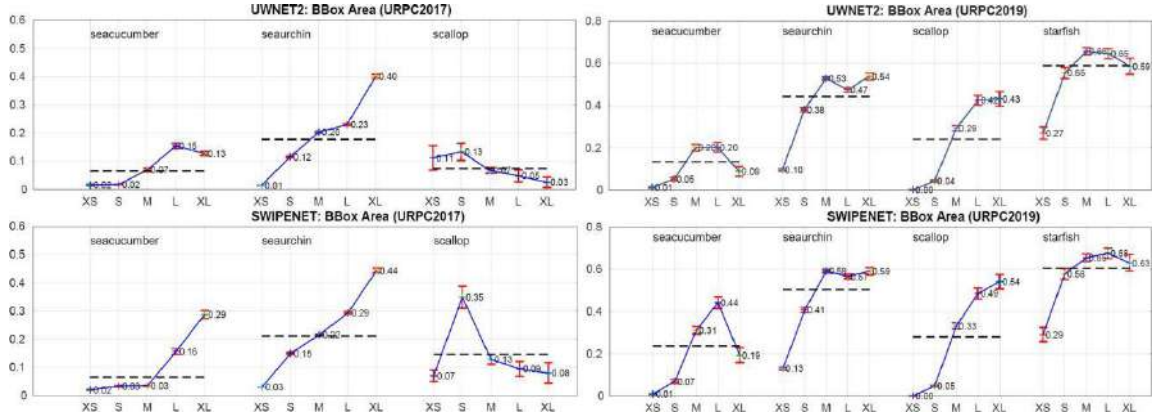

Fig. 4.5 The mean Average Precision of UWNET2 and SWIPENET for objects with different object sizes on URPC2017 and URPC2019. The object size is measured as the pixel area of the bounding box. XS (bottom 10%)=extra-small; S (next 20%)=small; M (next 40%)=medium; L (next 20%)=large; XL (next 10%)=extra-large.

Table 4.2 The red numbers indicate the results of the 'clean' SWIPENETs.

Dataset	Stage	NECMA				
	Iteration	1	2	3	4	5
URPC2017	Single	42.1	44.2	45.3	40.5	37.2
	Ensemble	42.1	45.0	46.3	45.3	44.2
URPC2018	Single	62.2	63.3	62.4	61.2	59.3
	Ensemble	62.2	64.5	64.0	62.8	62.1
URPC2019	Single	57.6	58.5	57.2	56.9	56.1
	Ensemble	57.6	59.9	59.0	59.0	59.5
ChinaMM	Single	76.1	77.5	78.3	76.5	74.8
	Ensemble	76.1	78.5	79.9	77.8	78.5

classification ability. We also present the mean Average Precision (mAP) of UWNET2 and SWIPENET for the objects with different sizes in Fig. 4.4 and Fig. 4.5, from which we observe the skip connection and dilated convolution block largely improves the small object detection accuracy. For example, for small objects (S) of seacucumber, seaurchin and scallop categories, SWIPENET improves 5%~6% mAP over UWNET2 on URPC2018 and ChinaMM.

4.4.2 Ablation Studies on CMA

In this subsection, we investigate the influence of CMA, including NECMA and NLCMA, on the final detection results. In our experiments, the number of the iterations of NECMA is set to 5 and the number of the iterations of NLCMA is set to 7. Tables 4.2 and 4.3 show the performance of the single model and the ensemble model after each iteration on the testing set of four datasets.

The role of NECMA. From Table 4.2, in the noise-eliminating stage (NECMA), we observe that the 'clean' SWIPENET is achieved in the 3rd iteration on URPC2017 and ChinaMM, and in the 2nd iteration on URPC2018 and URPC2019. So we set $M_1 = 3$ on UPRC2017 and ChinaMM, and $M_1 = 2$ on URPC2018 and URPC2019. The 'clean' SWIPENETs perform much better than the detectors in the 1st iteration. We assume this is because the noisy data (backgrounds but labelled as objects categories in the annotation process) confuse the

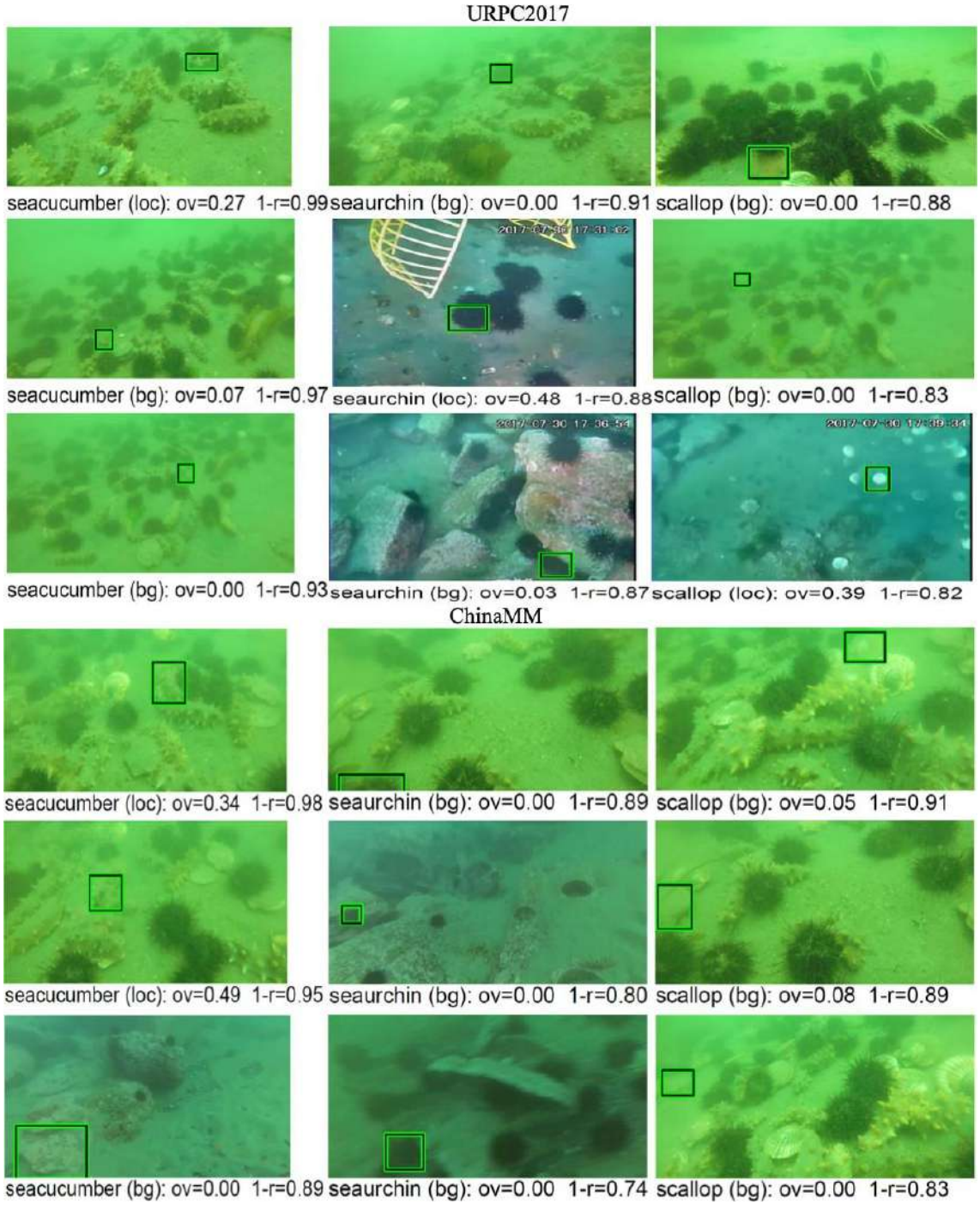

Fig. 4.6 Examples of top false positives of the SWIPENET without CMA. We show the top three false positives (FPs) for all categories on URPC2018 and ChinaMM. The text indicates the type of error ("loc"=localization; "bg"=confusion with backgrounds), the amount of overlap ("ov") with a true object, and the fraction of correct examples that are ranked lower than the given false positive ("1-r", for 1-recall). Localization errors are due to insufficient overlaps.

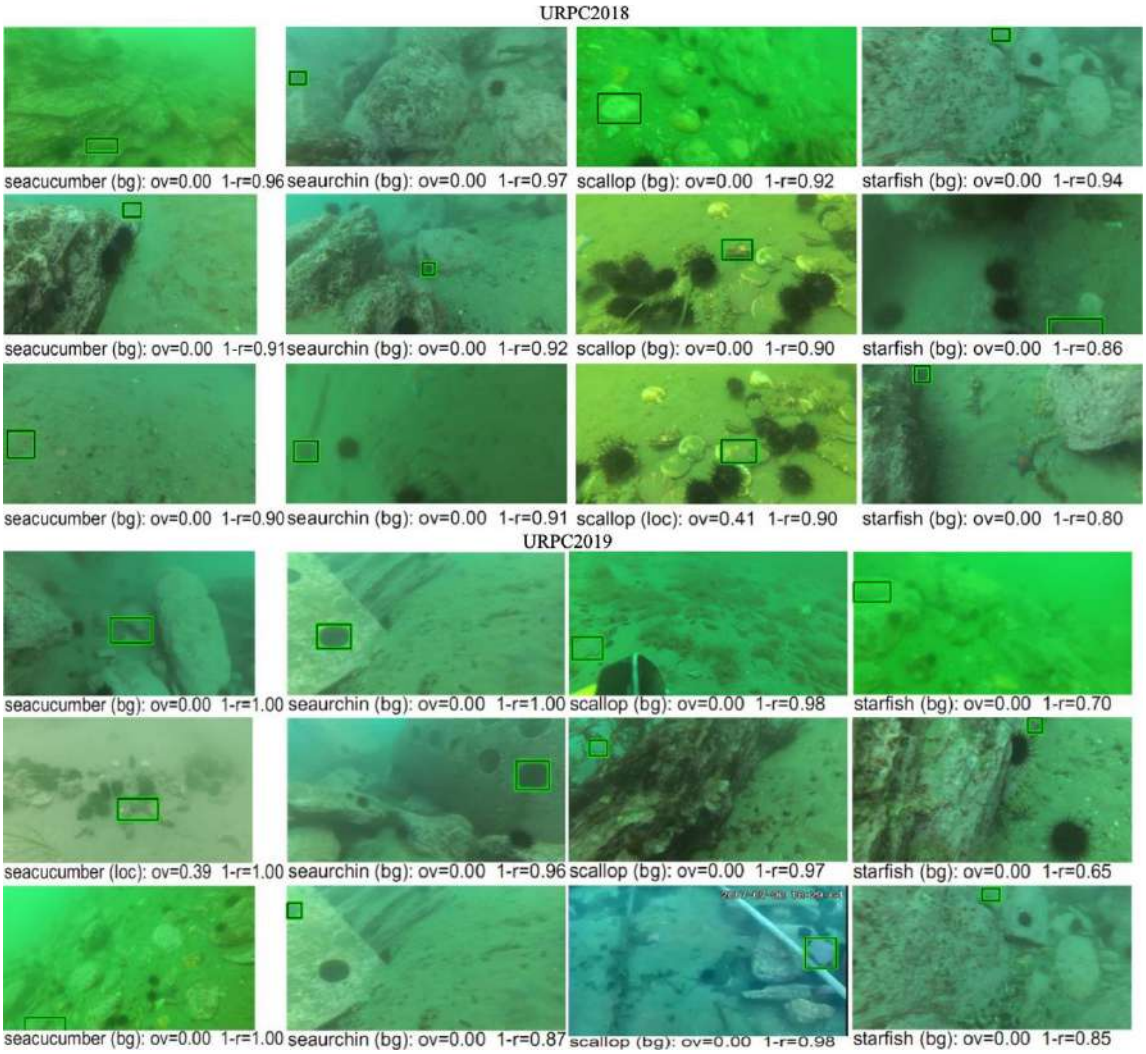

Fig. 4.7 Examples of top false positives of SWIPENET without CMA. We show the top three false positives (FPs) for all categories on URPC2017 and URPC2019. The text indicates the type of error ("loc"=localization; "bg"=confusion with backgrounds), the amount of overlap ("ov") with a true object, and the fraction of correct examples that are ranked lower than the given false positive ("1-r", for 1-recall). Localization errors are due to insufficient overlaps (less than 0.5).

Table 4.3 The performance (mAP(%)) of SWIPENET in each iteration of NLCMA on test set of four datasets.

Dataset	Stage	NLCMA						
	Iteration	1	2	3	4	5	6	7
URPC2017	Single	47.5	47.2	46.2	47.9	**48.0**	47.0	47.6
	Ensemble	47.5	48.6	49.8	52.3	**52.5**	52.5	52.5
URPC2018	Single	65.0	64.8	**65.3**	64.5	64.5	63.9	64.3
	Ensemble	65.0	65.4	66.9	67.5	**68.0**	68.0	68.0
URPC2019	Single	**61.8**	61.5	61.6	61.0	59.5	61.5	61.0
	Ensemble	61.8	62.4	**63.9**	63.9	63.9	63.9	63.9
ChinaMM	Single	80.4	79.8	**82.3**	81.4	79.5	80.0	79.3
	Ensemble	80.4	81.9	83.4	**85.6**	85.5	85.6	85.6

detectors in the 1st iteration. Fig. 4.6 and Fig. 4.7 show the top three false positives for the 1st detector, i.e. the SWIPENET trained without CMA, we can see that the background error (detecting the backgrounds as the objects) has much influence on the detectors than the localisation error (inaccurate localisation). To further verify this assumption, we use the detection analysis tool of [128] to analyse the false positives of the 1st detector and the 'clean' detector in NECMA. Fig. 4.8 and Fig. 4.9 show the distribution of the top-ranked false positives for each category on four datasets. We can see that the 1st detector cannot well distinguish the objects with complex background and generate much more background errors than the 'clean' detector. NECMA gradually reduces the influence of the noisy data on the single detector by decreasing their weights, and the background error clearly decreases in the detection results of the 'clean' SWIPENET. However, the performance of the single detectors after the 'clean' SWIPENET is less satisfactory. This is because most of the detected objects are continuously up-weighted and the detectors over-fit over these high-weight objects.

The role of NLCMA. In the noise-learning stage (NLCMA), we initialise each detector using the parameter learned in the 'clean' SWIPENET. This strategy provides a good initialisation for the following detectors which avoid getting stuck in poor local minima during the training. With this initialisation strategy, the detectors converge much faster during the training, shown in Fig. 4.10 (we also take the testing set as the validation set and investigate

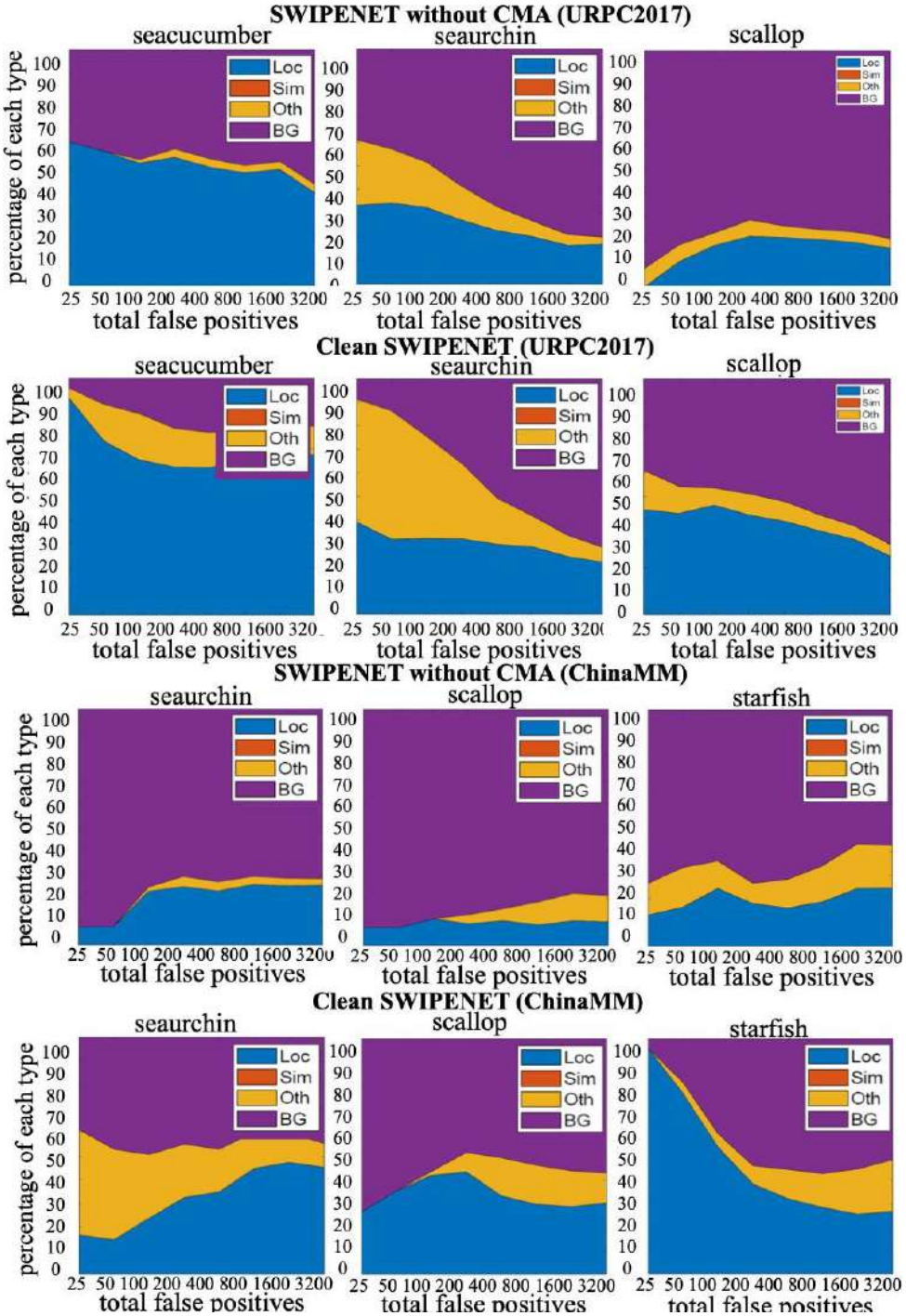

Fig. 4.8 The distribution of top-ranked false positive types of the SWIPENET without CMA and the 'clean' SWIPENET for each category on URPC2018 and ChinaMM. The false positive types include localisation error (Loc), confusion with similar categories (Sim), with others (Oth), or with background (BG).

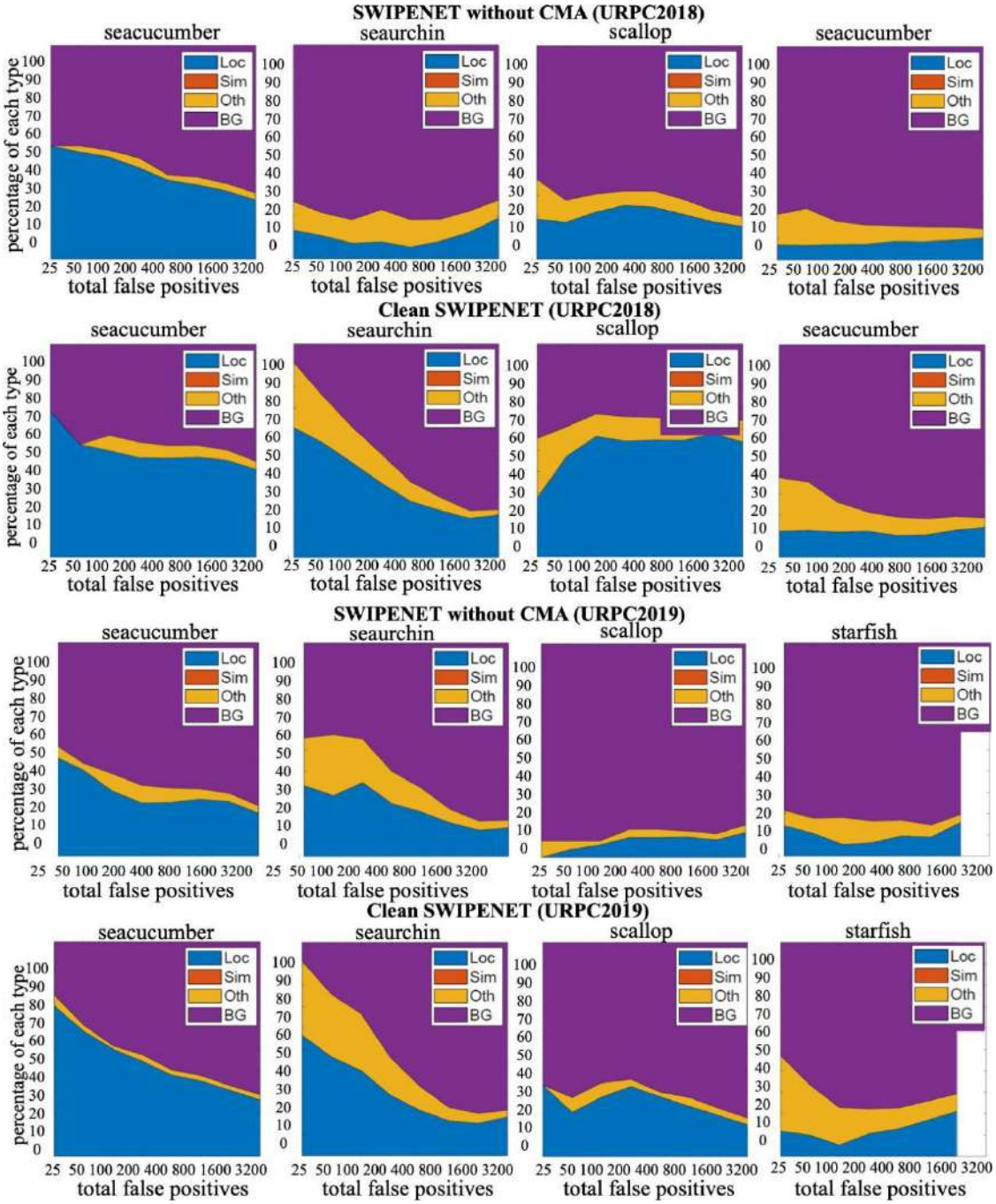

Fig. 4.9 The distribution of top-ranked false positive types of the 1st detector in NECMA (top) and the 'clean' SWIPENET (bottom) for each category on URPC2017 and URPC2019. The false positive types include localisation error (Loc), confusion with similar categories (Sim), with others (Oth), or with background (BG).

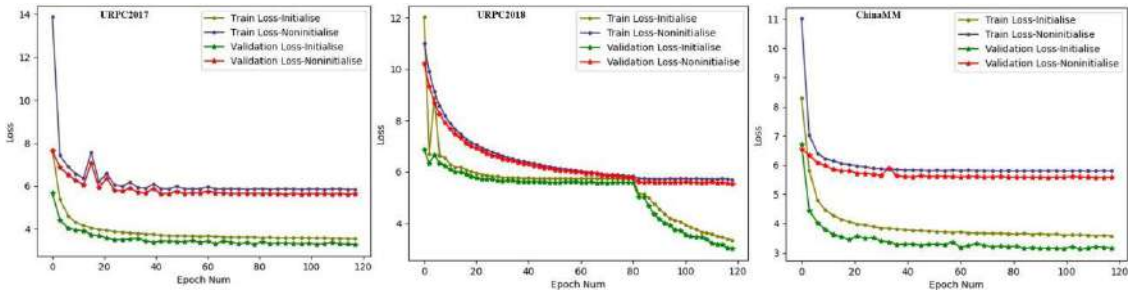

Fig. 4.10 The learning curve of SWIPENETs with and without initialisation by the 'clean' SWIPENET.

the influence of this initialisation strategy on the validation loss). From Table 4.3, we can see all the detectors in NLCMA perform better than the 'clean' detectors. This is because the detectors in NECMA take all the undetected objects as the noisy data and ignore learning them, however, in addition to the noisy data, the undetected objects also contain many hard targets, which are hard to be detected due to their minor discrepancies with the backgrounds. The 'clean' detector trained by NECMA can only detect the easy objects well but mis-detect many hard targets that limits the generalization of the detector. Different from detectors in NECMA, the detectors in NLCMA are able to detect the hard targets with the help of the 'clean' SWIPENET. The fundamental knowledge learnt by the 'clean' SWIPENET helps the following detectors identify the minor discrepancies between the hard targets and the backgrounds.

4.4.3 Ablation Studies on the Selective Ensemble Algorithm.

We investigate the influence of selective ensemble algorithm (SE) on the performance of the final ensemble detector. Fig. 4.11 shows the performance of the ensemble detector with different numbers of the selected detectors. The SE algorithm reduces the number of the detectors in the final ensemble. For example, the ensemble detector without SE achieves the best mAP on URPC2017 and URPC2018 when we ensemble five detectors, but the ensemble detector with SE achieves the same mAP by only integrating three selected detectors on URPC2017 and two selected detectors on URPC2018. This demonstrates some of the detectors do not help boosting the final performance in the ensemble. Few detectors

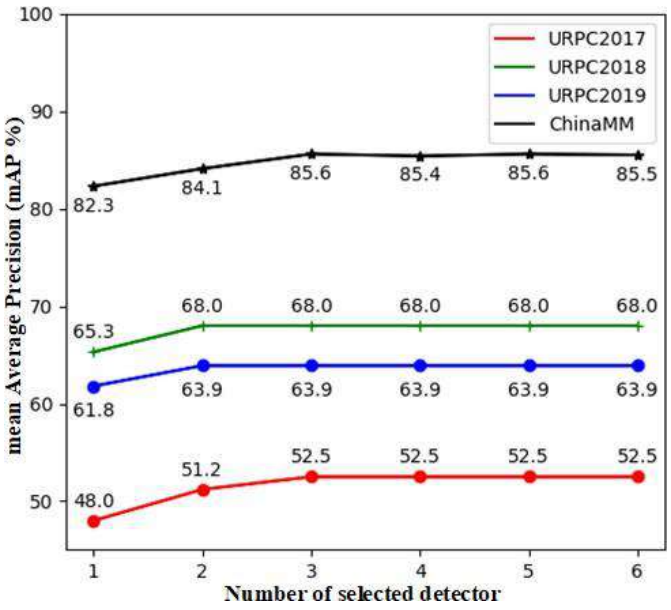

Fig. 4.11 The performance of the ensemble with different numbers of detectors.

with large diversity are sufficient to achieve the best performance. The selective ensemble algorithm surely helps reduce the computational overhead during testing due to the reduced number of the detectors.

4.5 Comparison with SOAT detection frameworks

In this section, we compare our proposed deep detector with other state-of-the-art (SOAT) detection frameworks. Since our SWIPENET are specially designed to improve small object detection, we first compare our proposed method with latest small object detection methods. Then, we compare SWIPENET+CMA framework with underwater object detection frameworks used in recent literatures.

4.5.1 Comparison with Small Object Detection Frameworks

Following the latest small object detection work [147], we select DSSD [138], RetinaNet [54], FCOS [148], FRCNN-FPN [149], and layer fusion strategy S-alpha [147] as the small object detection comparison methods. For fair comparison, we only compare our single models

Table 4.4 Comparison with small object detection frameworks on URPC2017

Dataset	URPC2017			
Methods	seacucumber	seaurchin	scallop	mAP
DSSD	13.2	70.3	26.4	36.6
FCOS	23.6	**75.2**	33.8	44.2
RetinaNet	19.2	72.9	28.9	40.3
FRCNN-FPN	25.3	73.7	26.3	41.8
RetinaNet with S-α	27.2	74.6	31.6	44.5
FRCNN-FPN with S-α	18.7	75.0	39.6	44.4
SWIPENET-noCMA	43.6	51.3	31.2	42.1
SWIPENET-Single	**46.6**	55.8	**41.6**	**48.0**

Table 4.5 Comparison with small object detection frameworks on URPC2018.

Dataset	URPC2018				
Methods	seacucumber	seaurchin	scallop	starfish	mAP
DSSD	48.4	75.3	38.2	64.0	56.5
FCOS	43.2	76.5	47.5	69.4	59.1
RetinaNet	52.5	74.9	43.1	69.8	60.1
FRCNN-FPN	57.7	76.9	38.1	70.6	60.9
RetinaNet with S-α	54.4	76.5	**52.4**	71.7	63.8
FRCNN-FPN with S-α	**59.1**	77.0	39.2	71.4	61.7
SWIPENET-noCMA	46.4	**84.0**	40.2	78.2	62.2
SWIPENET-Single	54.8	81.5	46.6	**78.4**	**65.3**

Table 4.6 Comparison with small object detection frameworks on URPC2019.

Dataset	URPC2019				
Methods	seacucumber	seaurchin	scallop	starfish	mAP
DSSD	22.8	79.8	43.2	75.3	55.3
FCOS	36.8	81.2	51.0	75.8	61.2
RetinaNet	34.8	79.5	50.4	74.8	59.8
FRCNN-FPN	40.2	81.8	54.5	75.9	63.1
RetinaNet with S-alpha	42.2	78.7	54.0	76.1	62.8
FRCNN-FPN with S-alpha	**46.3**	**82.8**	**54.9**	75.7	**64.9**
SWIPENET-noCMA	28.9	79.9	48.3	73.3	57.6
SWIPENET-Single	41.7	81.8	50.4	73.2	61.8

Table 4.7 Comparison with small object detection frameworks on ChinaMM.

Dataset	ChinaMM			
Methods	seacucumber	seaurchin	scallop	mAP
DSSD	54.5	82.0	79.4	72.0
FCOS	57.7	83.1	78.7	73.2
RetinaNet	59.6	82.0	81.0	74.2
FRCNN-FPN	58.0	82.1	81.6	73.9
RetinaNet with S-α	60.8	82.0	82.7	75.2
FRCNN-FPN with S-α	62.0	82.4	82.7	75.7
SWIPENET-noCMA	63.0	83.5	81.9	76.1
SWIPENET-Single	**77.0**	**84.7**	**85.2**	**82.3**

SWIPENET-noCMA (the SWIPENET trained without CMA) and SWIPENET-Single (the best single model achieved in the CMA) with other detection frameworks without considering the ensemble model.

Implementation details. For RetinaNet and FCOS, we use ResNet50 [150] as the backbone network. For DSSD and FRCNN-FPN, we use their original backbone networks. Following [147], we use FRCNN-FPN and RetinaNet with layer fusion strategy S-alpha as the detection frameworks. Both use ResNet50 [150] backbone. The comparison methods are tuned to have the best performance.

The experimental results on URPC2017, URPC2018 and ChinaMM are shown in Tables 4.4, 4.5 and 4.7, from which we obeserve SWIPENET-noCMA performs much better than DSSD, this is because multiple down-sampling operations lost many useful features, which are importance for accurate small object localization, these features cannot fully recovered by up-sampling operations once lost. The dilated convolution block in SWIPENET retains these features that benefits object localisation. On three datasets, our SWIPENET-Single achieves the best performance, its advantage comes from the SWIPENET backbone and the noisy eliminating strategy. It is worth noting that FCOS and RetinaNet and FRCNN-FPN frameworks apply much deeper backbones (ResNet50) than our SWIPENET, but SWIPENET-noCMA still achieves better performance than the former three frameworks on URPC2018 and ChinaMM, this demonstrates the multiple Hyper Features in SWIPENET

Table 4.8 Comparison with underwater object detection frameworks on URPC2017.

Dataset		URPC2017			
Methods	Backbone	seacucumber	seaurchin	scallop	mAP
SSD	VGG16	38.4	52.9	15.7	35.7
YOLOv3	DarkNet53	28.4	50.3	22.4	33.7
FRCNN	VGG16	27.2	45.0	31.9	34.7
FRCNN	ResNet50	31.0	41.4	33.5	35.3
FRCNN	ResNet101	26.2	47.7	32.5	35.5
FRCNN	FPN	25.3	73.7	26.3	41.8
IMA	SWIPENET	44.4	52.4	42.1	46.3
RetinaNet	ResNet50	19.2	72.9	28.9	40.3
FCOS	ResNet50	23.6	75.2	33.8	44.2
FreeAnchor	ResNet50	21.8	74.7	27.7	41.4
GHM	ResNet50	23.0	74.3	33.9	43.7
SWIPENET-Single	SWIPENET	46.6	55.8	41.6	48.0
SWIPENET-CMA	SWIPENET	**49.1**	**62.3**	**46.1**	**52.5**

is able to detect multi-scale objects well. FRCNN-FPN with S-alpha achieves the best performance on URPC2019 as shown in Table 4.5, this is because the layer fusion strategy S-alpha greatly boost the performance of small object detection, but it cannot solve the noise problem.

4.5.2 Comparison with Underwater Object Detection Frameworks

We also compare our method against several detection frameworks have ever applied for underwater object detection in recent literatures [33, 13], we only select the comparision methods whose source code is public availiable online, including IMA [33], SSD [112], YOLOv3 [26], FRCNN [24], RetinaNet [54], FCOS [148], FreeAnchor [151] and GHM [152].

Implementation details. For SSD, we use VGG16 [141] as the backbone. For Faster RCNN, we use four backbones including VGG16, ResNet50 [150], ResNet101 [150] and FPN [149]. For YOLOv3, we use its original DarkNet53 network. RetinaNet, FCOS,

Table 4.9 Comparison with underwater object detection frameworks on URPC2018.

Dataset		URPC2018				
Methods	Backbone	seacucumber	seaurchin	scallop	starfish	mAP
SSD	VGG16	44.2	84.4	35.8	78.1	60.6
YOLOv3	DarkNet53	35.7	83.0	34.0	77.9	57.7
FRCNN	VGG16	43.3	83.0	32.0	74.5	58.2
FRCNN	ResNet50	41.1	83.2	34.5	77.2	59.0
FRCNN	ResNet101	44.3	82.5	34.7	77.5	59.8
FRCNN	FPN	57.7	76.9	38.1	70.6	60.9
IMA	SWIPENET	52.8	84.1	42.9	78.0	64.5
RetinaNet	ResNet50	52.5	74.9	43.1	69.8	60.1
FCOS	ResNet50	43.2	76.5	47.5	69.4	59.1
FreeAnchor	ResNet50	46.2	72.3	42.5	71.4	58.1
GHM	ResNet50	52.4	78.4	42.1	71.5	61.1
SWIPENET-Single	SWIPENET	54.8	81.5	46.6	78.4	65.3
SWIPENET-CMA	SWIPENET	**56.4**	**84.6**	**50.9**	**79.9**	**68.0**

Table 4.10 Comparison with underwater object detection frameworks on URPC2019.

Dataset		URPC2019				
Methods	Backbone	seacucumber	seaurchin	scallop	starfish	mAP
SSD	VGG16	24.3	80.1	46.4	74.3	56.3
YOLOv3	DarkNet53	18.1	78.1	40.4	73.3	52.5
FRCNN	VGG16	20.9	79.1	43.5	73.2	54.2
FRCNN	ResNet50	22.8	79.8	43.2	75.3	55.3
FRCNN	ResNet101	25.4	79.1	46.4	74.6	56.4
FRCNN	FPN	40.2	81.8	54.5	75.9	63.1
IMA	SWIPENET	34.1	80.7	50.5	75.6	60.2
RetinaNet	ResNet50	34.8	79.5	50.4	74.8	59.8
FCOS	ResNet50	36.8	81.2	51.0	75.8	61.2
FreeAnchor	ResNet50	32.7	73.7	48.1	75.5	57.5
GHM	ResNet50	38.3	80.6	**53.2**	75.2	61.8
SWIPENET-Single	SWIPENET	41.7	81.8	50.4	73.2	61.8
SWIPENET-CMA	SWIPENET	44.8	81.8	53.0	**76.1**	63.9

Table 4.11 Comparison with underwater object detection frameworks on ChinaMM.

Dataset		ChinaMM			
Methods	Backbone	seacucumber	seaurchin	scallop	mAP
SSD	VGG16	47.3	80.3	78.1	68.6
YOLOv3	DarkNet53	33.1	80.2	77.9	63.7
FRCNN	VGG16	38.5	77.9	77.1	64.5
FRCNN	ResNet50	41.0	81.0	78.1	66.7
FRCNN	ResNet101	51.7	81.5	79.5	70.9
FRCNN	FPN	58.0	82.1	81.6	73.9
IMA	SWIPENET	68.3	83.3	84.5	78.7
RetinaNet	ResNet50	59.6	82.0	81.0	74.2
FCOS	ResNet50	57.7	83.1	78.7	73.2
FreeAnchor	ResNet50	41.9	80.6	76.9	66.4
GHM	ResNet50	53.7	82.1	82.3	72.7
SWIPENET-Single	SWIPENET	77.0	84.7	85.2	82.3
SWIPENET-CMA	SWIPENET	**82.2**	**87.1**	**87.6**	**85.6**

FreeAnchor and GHM all use ResNet50 [150] as the backbones. The comparison methods are tuned to have the best performance.

Tables 4.8, 4.9 and 4.11 show the experimental results on URPC2017, URPC2018 and ChinaMM, where our proposed SWIPENET-CMA achieves the best performance than other comparison methods. On three datasets, FRCNN with FPN performs better than FRCNN with ResNet101, ResNet50 and VGG16, where the deeper backbone FPN plays a critical role. SWIPENET-Single, the best single SWIPENET trained using CMA, outperforms the other frameworks by a large margin on three datasets, demonstrating the superiority of our proposed CMA in dealing with noisy data. It performs even better than the ensemble model trained with the IMA algorithm. This is because IMA regards all the undetected objects as noisy data and ignore learning them, which loses considerable effective training samples. Although the undetected objects tend to be noisy data or outliers, they also contain many hard object instances. Ignoring these hard object instances, IMA can only detect the easy objects well but cannot detect many hard objects. Similarly, GHM avoids learning both noisy data and hard objects, it can avoid the influence of the noisy data but cannot generalize well on the hard object instances. RetinaNet is easily to overfit on the noisy data because it employed

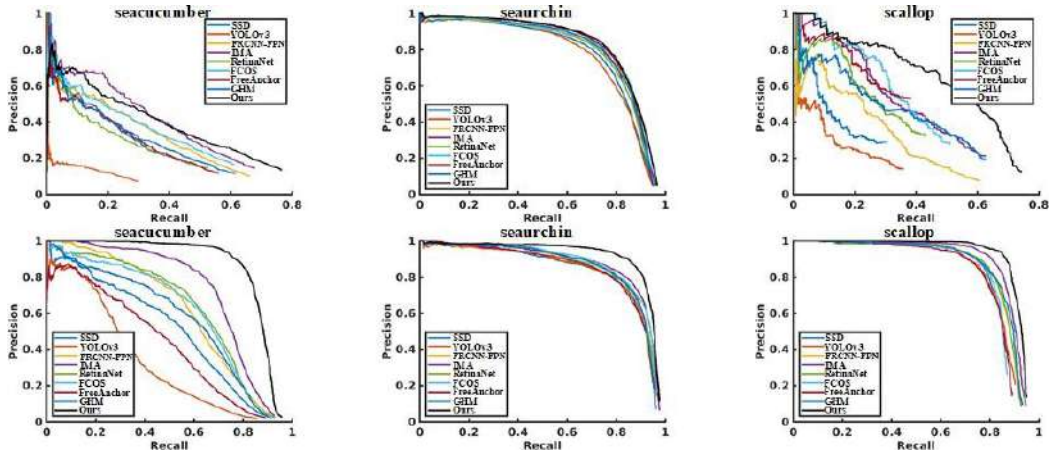

Fig. 4.12 Precision/Recall curves of different detection methods on URPC2017 (top row) and ChinaMM (bottom row).

the focal loss to train the detection network which emphasis on learning hard samples and also noisy data. Different from IMA and GHM, NLCMA stage of CMA focuses on learning possible hard object instances by increasing their weights, that improve the generalization on hard objects instances. SWIPENET-CMA further improves the SWIPENET-Single. The gain comes from its capacity to detect the diverse hard object instances. Fig. 4.12 and Fig. 4.13 show the Precision/Recall curves of different detection methods on four datasets, where we observe SWIPENET-CMA (black curve) achieves the best performance across all the object categories on URPC2017, URPC2018 and ChinaMM. On URPC2019, as shown in Tables 4.10, our proposed method ranks the 2nd place in all comparison methods. The best performance is achieved by the method FRCNN-FPN with S-alpha, which is specially designed for small object detection. We think two factors explain why FRCNN-FPN with S-alpha performs better than our proposed method on URPC2019: First, FRCNN-FPN with S-alpha applied the much deeper backbone Feature Pyramid Networks (FPN) and advanced layer fusion strategy S-alpha to boost the performance of small object detection. Second, URPC2019 has more accurate annotations (less noisy labels) than URPC2017 and URPC2018. The challenge from small objects is much bigger than that from the noisy data on URPC2019. With less influence of the noisy data, FRCNN-FPN with S-alpha performs better than our proposed method.

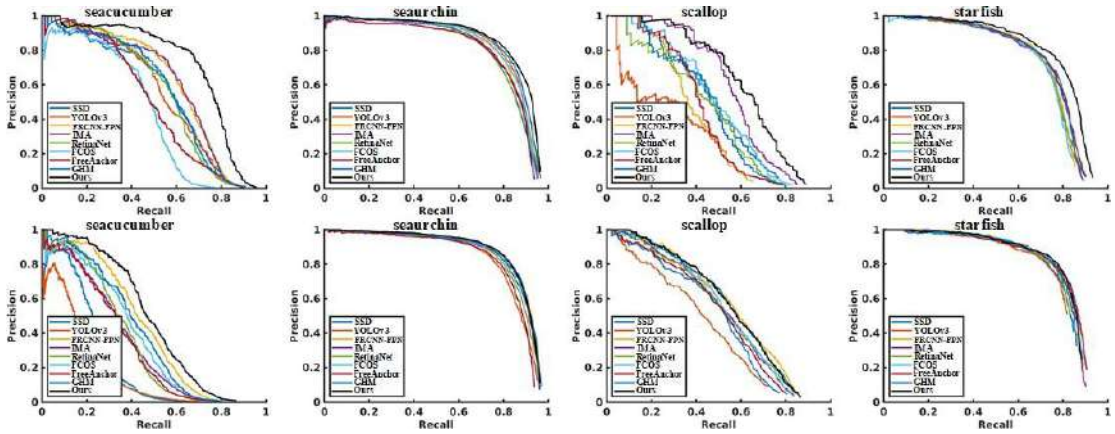

Fig. 4.13 Precision/Recall curves of different detection methods on URPC2018 (top row) and URPC2019 (bottom row).

Fig. 4.14 present visualization of object detection results of different detection frameworks on URPC2017, URPC2018 and URPC2019, we observe that most of the detection frameworks cannot detect all the objects, some of them detected the backgrounds as the objects. Among them, SWIPENET-CMA performs best.

4.5.3 Comparison with Representative Learning Paradigms

CMA combines the learning tricks from Multi-Class Adaboost [57] and Curriculum Learning [60], hence, we also conduct additional experiments to further compare our CMA learning paradigm with these two learning paradigms.

Implementation details. SWIPENET+MA train multiple detectors using the Multi-Class Adaboost algoithm and finally ensemble them into a unified model, focusing on learning undetected samples by up-weighting their weights. **SWIPENET+Curriculum** first trains a detector on the easy samples, then fine-tunes the detector of hard samples, since curriculum paradigm needs to define the easy and hard training samples: Similar to [137] that takes misclassified samples as the hard samples, we take the undetected objects as hard samples and the detected objects as easy samples. Specially, we first train a detector on all the training data, then we test the detector on the training data, the detected objects as easy and undetected objects as hard samples.

Fig. 4.14 Visualization of object detection results of different detection frameworks on URPC2017 (top images), URPC2018 (middle images) and URPC2019 (bottom images). From left to right are raw underwater images with ground truth, results of SSD, YOLOv3, FRCNN-FPN, IMA, RetinaNet, FCOS, FreeAnchor, GHM and our SWIPENET+CMA.

Table 4.12 The performance (mAP(%)) of SWIPENET in each iteration of different training paradigm on the test set of URPC2017, URPC2018 and ChinaMM.

Dataset	Iteration	1	2	3	4	5	6	7	8
URPC2017	CMA	42.1	45.0	46.3	47.5	48.6	49.8	52.3	**52.5**
	MA	**42.1**	41.0	40.5	39.2	39.5	38.8	40.2	39.8
	Curriculum	42.1	41.0	**43.9**	-	-	-	-	-
URPC2018	CMA	62.2	64.5	65.0	65.4	66.9	67.5	**68.0**	68.0
	MA	**62.2**	62.0	61.0	61.2	60.1	58.8	60.2	59.3
	Curriculum	62.2	62.1	**63.8**	-	-	-	-	-
URPC2019	CMA	57.6	59.9	61.8	62.4	**63.9**	63.9	63.9	63.9
	MA	**57.6**	56.2	57.0	57.6	56.9	56.8	55.8	56.3
	Curriculum	57.6	56.9	**60.8**	-	-	-	-	-
ChinaMM	CMA	76.1	78.5	79.9	80.4	81.9	83.4	**85.6**	85.5
	MA	76.1	**77.0**	76.5	76.0	75.5	75.7	75.0	74.7
	Curriculum	76.1	75.5	**78.2**	-	-	-	-	-

Table 4.12 shows the performance comparison of different training paradigms. Our CMA performs much better than the other training paradigms on all four datasets. After the 1st iteration, MA enable the detectors to focus on learning the hard data that degrade the system performance. This is because these hard data contain many noisy data confuse the detectors. On the four datasets, Curriculum decays the performance in the 2nd iteration but boosts the performance in the 3rd iteration. This is because Curriculum trains the detector using insufficient easy samples in the 2nd iteration. After having fine-tuned over the remaining hard samples, the performance is better than that in the 1st iteration. The gains come from the easy-to-hard training strategy and sufficient training data. However, CMA still performs much better than Curriculum. This is because the underwater datasets contain considerable diverse data resources due to frequently changing illuminations and environments, the ensemble model is able to learn diverse data and performs much better than the single model trained using the Curriculum paradigm whose generalisation ability is limited.

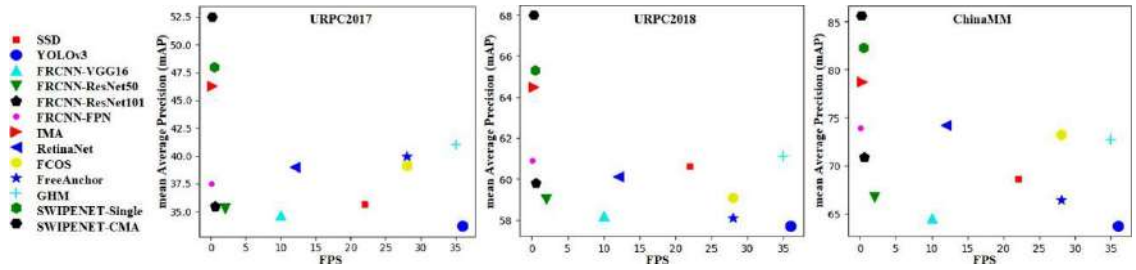

Fig. 4.15 Running time (Frames Per Second, FPS) vs mean Average Precision (mAP) of different detection frameworks.

4.6 Summary

This chapter aims to address the noisy data problem in underwater object detection. We have presented a new neural network architecture, called Sample-WeIghted hyPEr Network (SWIPENET), for small underwater object detection. Moreover, a sample re-weighting algorithm named Curriculum Multi-Class Adaboost (IMA) had been presented to deal with the noise issue. We also provide theoretical analysis on the ability of the sample-weighted detection loss in detail. To achieve the balance between the detection accuracy and the computational cost, we propose a selective ensemble algorithm to choose the best detector trained with large data diversity. Our proposed method well-handles the noise issue in underwater object detection and achieves the state-of-the-art performance on the challenging underwater datasets. However, since it is an ensemble deep model, the time complexity is much higher than current popular single models (as shown in Fig. 4.15). Hence, in our future work, reducing the computational complexity of our proposed method is of vital importance.

Chapter 5

Underwater Object Detection in Imbalanced Datasets

5.1 Introduction

Autonomous underwater vehicles (AUVs) [153] and remotely operated vehicles (ROVs) [154] equipped with intelligent underwater object detection systems play an important role in marine environment monitoring, underwater navigation, marine organism capturing and other fields. Robust underwater object detection is an indispensable technology for AUVs and ROVs to fulfill these tasks.

However, there are still several challenges in the field of underwater object detection: (1) Existing underwater object detection datasets contain considerable label noise and the distribution of label noise is highly imbalanced. The imbalance noise distribution is usually generated in the data annotation process [155, 156], whereas some object classes receive more erroneous labels than the others. For example, in the competition underwater object detection datasets URPC2017 and URPC2018 [1], the scallop class contains more incorrect labels than the others as scallops are visually similar to sands and stones in terms of color and texture. On the other hand, sea urchins are of black color that is largely different from the background, and

[1]URPC17 and URPC18 are two public competition underwater object detection detection from the underwater robot picking contest, which can be downloaded on the website of the underwater robot picking contest http://www.cnurpc.org/index.html

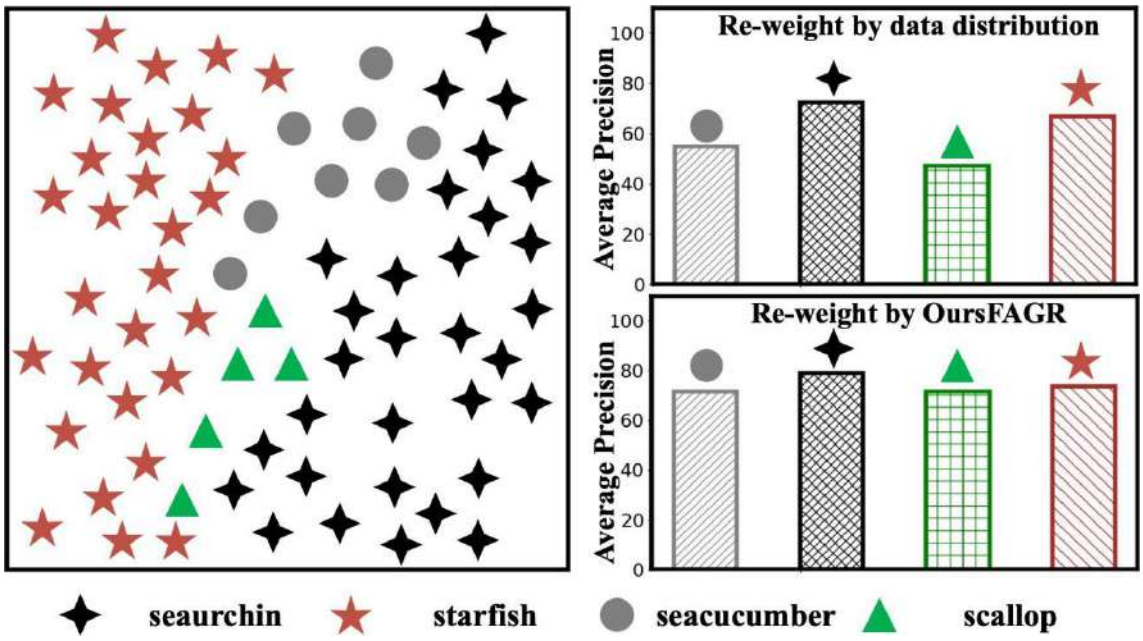

Fig. 5.1 Imbalance data distributions are commonly witnessed in large scale real-world underwater object detection datasets. Our precision distribution based re-balancing method (OursFAGR) outperforms the standard data distribution based re-balancing methods, e.g. [69], for the object detection task.

therefore there are much less incorrect labels for this class. These incorrectly labelled objects are considered as the outliers that challenge the established underwater object detection networks [33]. (2) Imbalance data distributions in the real-world underwater object detection datasets greatly degrade the performances of the existing detection systems. Detection and classification networks commonly suffer from the imbalance problem on the class imbalance datasets where the number of the samples of each class in the datasets is not balanced [62, 63]. Learning models trained on these datasets usually result in highly imbalanced performance over different classes, where the over-represented classes receive much better results than the under-represented classes [157–159].

The class imbalance problem has been extensively studied in the classification tasks, where large accuracy discrepancies have been produced due to the imbalance data distributions [73, 160–162]. Most classification neural networks exhibit biases towards learning the majority classes but ignore learning the minority class that bring high imbalance classification accuracy. In these studies, data distribution-based re-balancing strategies (e.g.

re-sampling [163, 164, 69, 165] and re-weighting [67, 68]) have been recruited to enhance the outcomes of the established approaches by generating a roughly uniform data distribution. In the detection tasks, most previous works focused on alleviating the foreground-background (object-background) imbalance problem rather than the foreground-foreground (object-object) imbalance problem. Zhou et al. [166] formed an attention driven loss to re-weight the foreground and background to address the class imbalance problem. Song et al. [167] exploited an attention-based hard negative mining strategy to filter out the background samples to re-balance the foreground and the background classes. Aiming to address the foreground-foreground class imbalance problem, Liu et al. [168] exploited the Poisson GAN to generate more samples of the minority classes to re-balance the class distribution.

However, these techniques cannot achieve satisfactory performance in noisy imbalanced underwater object detection datasets, mainly due to three reasons: (1) all these works assume that the datasets are clean without any label noise. However, in real-world scenarios, considerable label noise exists and imbalance noise exists in different classes; (2) There are intrinsic differences between detection and classification tasks [73]. Different from the classification tasks, object detection aims to estimate the locations and classes of the objects at the same time. The inherent localisation difficulties of the classes may also lead to imbalance detection. In practice, hard classes may need more training samples or training iterations to achieve the same detection precision as easy classes. Hence, data distribution based class imbalance techniques developed for the classification tasks cannot be directly applied to the object detection tasks due to the ignorance of critical factors such as imbalance noise distributions and diversity of classes; (3) Underwater object detection task faced much less foreground-background imbalance problem than the foreground-foreground imbalance problem, because the underwater creatures follow the gregarious behaviors and they are densely distributed in the underwater images without much excessive background samples. In this paper, we focused on addressing the foreground-foreground class imbalance problem. Different from the previous works, (1) we discover that the imbalanced label noise distributions exaggerate the imbalance detection problem, hence, we propose a noise removal (NR) algorithm to alleviate the influence of the imbalanced label noise distributions, (2) we

re-weight the classes based on the precision distribution rather than the data distribution and this is more suitable for addressing the class imbalance problem in the detection tasks.

5.2 Proposed Method

5.2.1 Noise Removal (NR)

5.2.1.1 Theoretical Analysis of Label Noise

We first give theoretical analysis on how label noise affects deep detection networks in the gradient optimisation. We choose Single Shot Detector (SSD) [112] as our basic detection framework, before conducting detailed analysis, we need to review some basic knowledge of SSD and define the mathematical symbols.

Denote the training set of the object detection dataset as $O = \{O_1, O_2, \ldots, O_j\}$, where object $O_j = \{cls, x, y, \Delta x, \Delta y\}$ is labeled using the ground-truth bounding box, which is a rectangular region defined by the object class cls and coordinates $(x, y, \Delta x, \Delta y)$. In the training stage, SSD does not directly use the objects as real training samples, instead, it deploys default boxes (also named anchors in [169]) on several convolutional layers. If the Intersection over Union (IoU) between a default box and its most overlapping object is larger than the pre-defined threshold. Then, the default box is a match to this ground-truth object, working as its positive training sample. If the default box does not match any object, it will be regarded as the negative training sample. Denote the default boxes/training samples set as $S = \{S_{1,1}, S_{1,2}, \ldots, S_{n,i}\}$, where $S_{n,i}$ denotes the i-th training sample of the n-th class. The default box $S_{n,i} = \{cls, loc\}$ is defined by its class cls and its relative coordinate loc. $loc = (cx, cy, \Delta x, \Delta y)$ denotes the coordinate information of the training sample that includes the coordinate of center (cx, cy) with width Δx and height Δy. Technically speaking, the detection loss L of SSD consists of a classification loss L_{cls} for the default box classification and a regression loss L_{reg} for the default box localisation with the following form:

$$L = \frac{\alpha_1}{N} L_{cls} + \frac{\alpha_2}{N} L_{reg} \tag{5.1}$$

where $\ddot{N}$ and $\bar{N}$ are the numbers of all the training samples and the positive training samples, respectively. Denote C as the number of the object classes, and N_n $(n = 1, 2, \ldots, C+1)$ as the number of the training samples belonging to the n-th class, we have the $C+1$-th class as the background/negative class. Afterwards, we have $\ddot{N} = \sum_{n=1}^{C+1} N_n$ and $\bar{N} = \sum_{n=1}^{C} N_n$. α_1 and α_2 denote the weight terms of the classification and regression losses, respectively. The classification loss L_{cls} is a softmax loss between the predicted class p and the ground-truth class y, defined as:

$$L_{cls} = -\sum_{n=1}^{C+1}\sum_{i=1}^{N_n}\sum_{c=1}^{C+1} y(c|S_{n,i}) \log p(c|S_{n,i}) \tag{5.2}$$

where

$$p(c|S_{n,i}) = \frac{e^{net(c|S_{n,i})}}{\sum_{j=1}^{C+1} e^{net(j|S_{n,i})}}, c = 1, 2, \ldots, C+1 \tag{5.3}$$

$p(c|S_{n,i})$ denotes the probability of $S_{n,i}$ being predicted as the c-th class, $y(c|S_{n,i})$ denotes the true probability of $S_{n,i}$ being the c-th class. $y(c|S_{n,i}) = 1$ if $S_{n,i}$ belongs to the c-th ground-truth class, and $y(c|S_{n,i}) = 0$ otherwise. $net(c|S_{n,i})$ denotes the classification prediction from the detection network for $S_{n,i}$ at the c-th class (referring to Figure 5.2 for more details).

L_{reg} is a smooth L1 loss between the predicted coordinate p^- and the ground-truth coordinate g, defined as follows:

$$L_{reg} = \sum_{n=1}^{C}\sum_{i=1}^{N_n}\sum_{l \in loc} SmoothL_1(p^-(l|S_{n,i}) - g(l|S_{n,i})) \tag{5.4}$$

where

$$SmoothL_1(x) = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{5.5}$$

$$p^-(l|S_{n,i}) = net(l|S_{n,i}), l \in loc \tag{5.6}$$

$p^-(l|S_{n,i})$ denote the l-th element of the predicted coordinate vector for $S_{n,i}$, and $net(l|S_{n,i})$ denotes the l-th element of the coordinate prediction from the detection network.

We conduct theoretical analysis to discuss what factors result in imbalance detection. SSD is maintained by gradient-based optimisation [170], in which the loss function provides necessary gradients for updating the model parameters in the back-propagation mode. The gradient magnitude of a class determines how much impact it has on the updating of a

DNN [171, 172]. The classes generating larger gradient magnitudes have more impacts on the updating of a DNN, and the DNN focuses on learning these classes while ignoring other classes. Hence, we can explain why some classes are under-represented by investigating the classes' gradient magnitude in the derivation. We denote the parameter of the detector as ϑ, and its derivative is (the detailed derivation can be found in the Supplementary):

$$\frac{\partial L}{\partial \vartheta} = \begin{cases} \frac{\alpha_1}{N} \sum_{n=1}^{C+1} \sum_{i=1}^{N_n} \sum_{c=1}^{C+1} y(c|S_{n,i})(p(c|S_{n,i}) - 1) \frac{\partial net(c|S_{n,i})}{\partial \vartheta} \\ + \frac{\alpha_2}{N} \sum_{n=1}^{C} \sum_{i=1}^{N_n} \sum_{l \in loc} (p^-(l|S_{n,i}) - g(l|S_{n,i})) \frac{\partial net(l|S_{n,i})}{\partial \vartheta} \\ \qquad \textbf{if } |p^-(l|S_{n,i}) - g(l|S_{n,i})| < 1 \\ \frac{\alpha_1}{N} \sum_{n=1}^{C+1} \sum_{i=1}^{N_n} \sum_{c=1}^{C+1} y(c|S_{n,i})(p(c|S_{n,i}) - 1) \frac{\partial net(c|S_{n,i})}{\partial \vartheta} \\ \pm \frac{\alpha_2}{N} \sum_{n=1}^{C} \sum_{i=1}^{N_n} \sum_{l \in loc} \frac{\partial net(l|S_{n,i})}{\partial \vartheta} \qquad \textbf{otherwise} \end{cases} \tag{5.7}$$

To simplify Eq. (5.7), we denote gra_cls_n and gra_loc_n as the gradients generated from the classification and regression losses by the training samples belonging to the n-th class. Then, Eq. (5.7) can be re-formulated as Eq. (5.8).

$$\frac{\partial L}{\partial \vartheta} = \frac{\alpha_1}{N} \sum_{n=1}^{C+1} gra\_cls_n + \frac{\alpha_2}{N} \sum_{n=1}^{C} gra\_loc_n \tag{5.8}$$

where

$$\begin{aligned} gra\_cls_n &= \sum_{i=1}^{N_n} \sum_{c=1}^{C+1} y(c|S_{n,i})(p(c|S_{n,i}) - 1) \frac{\partial net(c|S_{n,i})}{\partial \vartheta} \\ &= \sum_{i=1}^{N_n} (p(n|S_{n,i}) - y(n|S_{n,i})) \frac{\partial net(n|S_{n,i})}{\partial \vartheta} \end{aligned} \tag{5.9}$$

$$gra\_loc_n = \begin{cases} \sum_{i=1}^{N_n} \sum_{l \in loc} (p^-(l|S_{n,i}) - g(l|S_{n,i})) \frac{\partial net(l|S_{n,i})}{\partial \vartheta} \\ \qquad \textbf{if } |p^-(l|S_{n,i}) - g(l|S_{n,i})| < 1 \\ \pm \sum_{i=1}^{N_n} \sum_{l \in loc} \frac{\partial net(l|S_{n,i})}{\partial \vartheta} \qquad \textbf{otherwise} \end{cases} \tag{5.10}$$

Eqs. (5.9) and (5.10) show that the gradient magnitude of the n-th class relies on three factors. First, large N_n (i.e., the number of the training samples belonging to the n-th class) bring large gradient magnitude. Suppose all the samples generate the same gradient from both classification and regression losses, the gradient magnitude of the majority classes will be much larger than that of the minority classes. Hence, the majority classes have more impacts on the updating of the detector, and the detector focuses on learning the majority class while ignoring the minority classes, leading to the large precision discrepancies.

Second, large $|p(n|S_{n,i}) - y(n|S_{n,i})|$ and $|p^-(l|S_{n,i}) - g(l|S_{n,i})|$ values bring a large gradient magnitude. For incorrectly labelled data, outliers identified by the deep detection network,

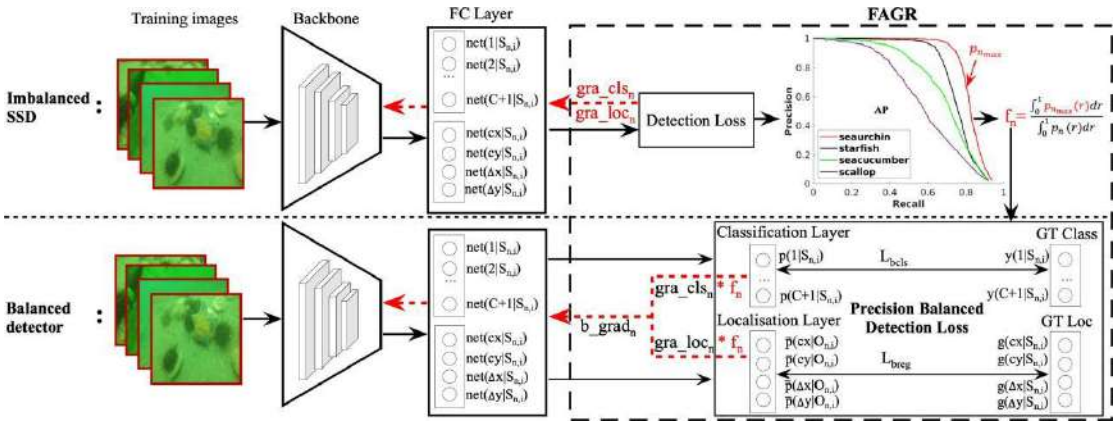

Fig. 5.2 The pipeline of the proposed FAGR. First, FAGR computes the gradient-adjustment coefficient f_n for the n-th class based on the precision discrepancies generated by the imbalanced SSD. Then, FAGR applies f_n to adjust the gradient magnitude of the n-th class and provides a detection balanced gradient b_grad_n for the n-th class so that the detector pays equally attentions on learning all the classes. Dash red arrows indicate the gradients' direction.

the predicted class $p(n/S_{n,i})$ and coordinate $p(l/S_{n,i})$ are largely different from the human-labelled class $y(n/S_{n,i})$ and coordinate $g(l/S_{n,i})$. Hence, the incorrectly labelled data are likely to produce a large gradient magnitude that makes the deep detection network over-fitting. The deep detector trained over the data cannot distinguish their human-labelled classes from their true classes and make accurate coordinate predictions. For example, the scallop class contains considerable background label noise, it will learn confusing feature representations from both the scallop instances and background instances, which cannot be used to make accurate predication for the scallops.

Finally, the terms $\frac{\partial net(n/S_{n,i})}{\partial \vartheta}$ and $\frac{\partial net(l/S_{n,i})}{\partial \vartheta}$ also influence the gradient magnitude of each class. Since there is no closed form solution to determine whether or not this factor affects the precision, we conduct the empirical analysis in the experimental section to further investigate whether there are additional factors that influences imbalance detection.

5.2.1.2 The Proposed Noise Removal (NR) Algorithm

The existence of label noise largely imparts the performance of the detection networks, and the imbalance label noise also exaggerate the imbalanced detection problem. Hence, it is

necessary to propose a noise removal (NR) algorithm to remove the influence of label noise on the underwater object detection network.

To alleviate the influence of label noise on the detection networks, a straight way is to pick out the label noise and filter them out. To pick out the incorrectly labelled objects, we first train a deep noise-immune detector using the Invert Multi-class Adaboost (IMA) algorithm proposed in [33], which is designed to remove the influence of noisy data on the deep models. Then, we run the noise-immune detector on the training set $O = \{O_1, O_2, \ldots, O_j\}$ and receive the predictions $P = \{P_1, P_2, \ldots, P_i\}$. We take the detected objects as the clean data and the undetected objects as the noisy data. This is because the detector trained by IMA focuses on learning clean data and ignores learning the noisy data, and most of the clean data can be detected while the noisy data hard to be detected. In the object detection field, the evaluation metric Intersection over Union (IoU) is commonly used when we determine whether a object has been detected or not. If there exists a prediction, and the Intersection over Union (IoU) between this prediction and the object O_j is larger than a predefined IoU threshold σ , O_j is regarded as the detected object. The IoU threshold σ is an important hyper-parameter that controls how many objects will be removed for different classes, we investigate how different settings of σ affect the final performance in the experiments.

Label noise has been picked out, however, to remove them from the training set is not an easy thing, because each image contains multiple objects including noisy objects and clean objects, directly deleting the images containing noisy objects will lead to the loss of clean objects. To avoid this negative effect, we set the weight of the clean objects as 1 and the weight of the noisy objects as 0 using Eq. 5.11 so that the noisy objects cannot contribute the updating of the network parameters.

$$I(O_j) = \begin{cases} 1 \text{ if } \exists P \in \boldsymbol{P} : IoU(O_j, P) \geq \sigma \\ 0 \text{ otherwise} \end{cases} \tag{5.11}$$

$$IoU(O_j, P) = \frac{O_j \cap P}{O_j \cup P} \tag{5.12}$$

In practice, SSD extracts default boxes around the objects as the real training samples. The default box $S_{n,i}$ will be classified as the positive and negative training sample: if $S_{n,i}$ does not match any object with an IoU higher than 0.5, it is a negative training sample; if it matches an object with an IoU higher than 0.5, it is a positive training sample. $S_{n,i}$ is matched to its most overlapped object O_{j^*}. We define a novel noise-removal term $I_{n,i}$ for the default box $S_{n,i}$: if $S_{n,i}$ matches a noisy object, its weight is set as the noisy object's weight (i.e., 0) so that this noisy training sample cannot feed gradients to update the network parameters. The weights of other default boxes are set as 1.

$$I_{n,i} = \begin{cases} I(O_{j^*}) \textbf{ if } \exists O \in O : IoU(S_{n,i}, O) \geq 0.5 \\ 1 \textbf{ otherwise} \end{cases} \tag{5.13}$$

where

$$\begin{aligned} O_{j^*} &= \underset{O_j}{\operatorname{argmax}} IoU(O_j, S_{n,i}) \\ s.t. \quad & IoU(O_j, S_{n,i}) \geq 0.5 \end{aligned} \tag{5.14}$$

The noise-removal term $I_{n,i}$ will be incorporated into our final detection loss (formulated as Eqs. (5.19) and (5.20)) to remove the influence of label noise.

5.2.2 Factor-agnostic Gradient Re-weighting (FAGR)

We have explained that the imbalance detection problem cannot be well-addressed by data distribution based re-balancing methods in the previous analyses. However, it is still hard to figure out all the influential factors of the imbalance detection. Evenly, it is sure that the combination effect of all the possible factors leads to the imbalance detection precision, which motivates us to re-weight the classes according to the precision discrepancy.

To enable the detector to treat all the classes equally in terms of detection precision, we facilitate precision balanced gradients for all the classes so that the detector focuses on learning low-precision classes and then generating balanced precision for all the classes. One way to emphasise on learning low-precision classes is to increase their gradient magnitudes during the optimisation. Here, we propose a factor-agnostic gradient re-weighting (FAGR)

algorithm that re-weights all the classes according to the class precision discrepancies. FAGR modifies the gradients of all the classes, and multiplies a gradient-adjustment coefficient f_n to the n-th class. This coefficient fine-tunes the gradient magnitudes of low-precision classes and produces detection balanced gradients *b_grad*, formulated as Eq. (5.15) for all the classes.

$$b\_grad_n = f_n * grad\_cls_n + f_n * grad\_loc_n \tag{5.15}$$

where $grad\_cls_n$ and gra_loc_n are the gradients generated by the samples belonging to the n-th class from the standard detection loss defined by Eqs. (5.9) and (5.10).

Since f_n is responsible for balancing the precision of all the classes, the precision discrepancies should be taken into account when we design the metric. As shown in Figure 5.2, we first train an imbalanced SSD on the imbalanced data set, then we compute the precision discrepancies in the detection results. The average precision AP_n of the n-th class is defined as the area under the precision-recall curve $p_n(r)$ and can be computed using Eq. (5.16) (refer to Figure 5.2 for better more dteails).

$$AP_n = \int_0^1 p_n(r)dr, n = 1, 2, \dots C \tag{5.16}$$

Then, we select the class with the best AP as the base class n_{max} , keep its gradient magnitude unchanged and adjust the gradient magnitudes of the other low-precision classes. This strategy helps improve the influences of the low-precision classes on the detector with the intention to achieve better precision.

$$n_{max} = \underset{n}{\operatorname{argmax}} AP_n, n = 1, 2, \dots C \tag{5.17}$$

Finally, we derive the n-th class's gradient-adjustment coefficient f_n using Eq. (5.18).

$$f_n = \frac{AP_{n_{max}}}{AP_n} = \frac{\int_0^1 p_{n_{max}}(r)dr}{\int_0^1 p_n(r)dr}, n = 1, 2, \dots C \tag{5.18}$$

We leverage f_n as the gradient-adjustment coefficient for two reasons: (1) It can minimise the class precision discrepancy by attaching higher importance for the classes with lower

precision. (2) It is factor-agnostic and we do not need to interpret what factors result in the precision discrepancy and to what extent.

Finally, we incorporate the noise-removal term $I_{n,i}$ and gradient-adjustment coefficient f_n into our final class-balanced detection loss to train a precision-balanced detector. The classification branch and localization branch of our final precision balanced detection loss are formulated as Eqs. (5.19) and (5.20), respectively.

$$L_{bcls} = -\sum_{n=1}^{C+1}\sum_{i=1}^{N_n} I_{n,i} f_n \sum_{j=1}^{C+1} y(c/S_{n,i}) log p(c/S_{n,i}) \tag{5.19}$$

$$L_{breg} = \sum_{n=1}^{C}\sum_{i=1}^{N_n} I_{n,i} f_n \sum_{l \in Loc} SmoothL_1(g(l/S_{n,i}) - p^{-}(l/S_{n,i})) \tag{5.20}$$

The proposed precision balanced detection loss enables FAGR to control the contribution of each class to the deep detection networks by adjusting the gradient magnitude of each class.

Models	Backbone	seacucumber	seaurchin	scallop	starfish	mAP
YOLOv3	DarkNet53	43.5	60.6	41.4	54.2	49.9
SSD	VGG16	47.5	60.3	47.0	57.9	53.2
F-RCNN	VGG16	46.8	58.8	45.9	57.0	52.1
F-RCNN	ResNet50	46.9	60.1	46.9	57.1	52.8
F-RCNN	ResNet101	47.1	60.2	47.2	57.5	53.0
SSD+NR	VGG16	50.6	**61.9**	51.3	58.4	55.8
SSD+NR+FAGR	VGG16	**55.8**	61.5	**56.5**	**59.5**	**58.3**

Table 5.1 Performance of different deep detection networks on Balance18. NR and FARG indicate the noise removal and factor-agnostic re-weighting strategies, respectively.

5.3 Experimental Setup

To demonstrate the effectiveness of the proposed NR+FAGR algorithm, we conduct comprehensive evaluations over four datasets, including two datasets (URPC2017 and URPC2018) with both imbalanced data distribution and imbalanced label noise distribution, and two datasets (Balance18 [173] and PASCAL VOC2007Noise) with balanced data distribution

but imbalanced label noise distribution. In this section, we first introduce the experimental datasets. Then, we describe the implementation details.

5.3.1 Datasets

URPC2017 and URPC2018 are two public competition underwater object detection detection from the underwater robot picking contest[2]. These two datasets have highly imbalanced data distributions and label noise distributions. URPC17 contains three classes (seacucumber, seaurchin, and scallop classes) while URPC18 contains four classes (seacucumber, seaurchin, scallop, and starfish classes).

Balance18 [173] has a balanced data distribution but an imbalanced label noise distribution. The dataset contains four classes, including seacucumber, seaurchin, scallop and starfish. Each class has similar number of object instances, however, the label noise in different classes is largely different. The scallop and seacucumber classes have much more label noise than the other classes.

PASCAL VOC2007Noise also has balanced data distribution but imbalanced label noise distribution, it is generated from the PASCAL VOC2007 dataset [174] which is commonly used to evaluate different detection frameworks in general object detection field. We manually added imbalanced label noise into different classes of PASCAL VOC2007 to produce PASCAL VOC2007Noise, the noise rate is set to 0% and 40% to ensure the clean labels are dominant.

5.3.2 Implementation Details

Our proposed NR and FAGR are implemented in the SSD framework using Python and Keras (the source code will be published on Github). In our proposed SSD+NR+FAGR framework, we first remove the label noise using the NR algorithm, then we apply the FAGR algorithm to train our balanced detector. To implement the proposed FAGR algorithm, we first train a class-imbalanced SSD without using any class imbalance strategy, and then we use FAGR to

[2]URPC17 and URPC18 can be downloaded on the website of the underwater robot picking contest http://www.cnurpc.org/index.html

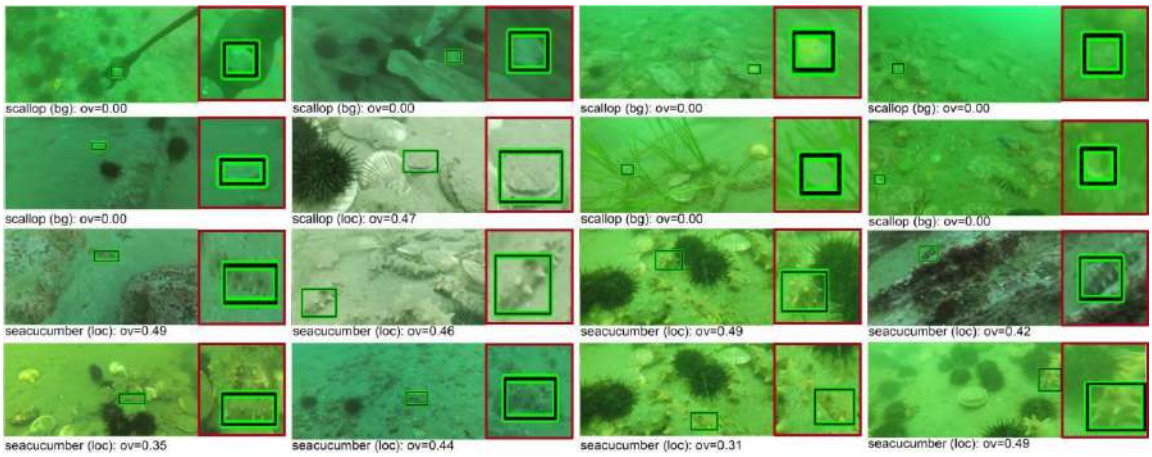

Fig. 5.3 False positives of SSD for the scallop and seacucmber classes on Balance2018. The text indicates the type of error ("loc"=localization; "bg"=confusion with backgrounds), the amount of overlap ("ov") with a ground truth object. The patches in the red boxes show the details of the false positives.

compute the gradient-adjustment coefficient for each class. Finally, the gradient-adjustment coefficient support the training of the second detector which is class-balanced. Both two detectors are initialised by the VGG16 backbone pretrained on the ImageNet dataset [175]. We train each detector on URPC2017 for 120 epochs and URPC2018 for 80 epochs using a batch size of 16 and a learning rate of 0.0001 on URPC2017 and 0.001 on URPC18. All the experiments are conducted on a single NVIDIA Tesla P100 GPU with a 16 GB memory.

5.4 Experiments on Balance18 and VOC2007Noise

A number of previous works assumed that imbalanced data distribution accounts for the imbalanced classification and detection. However, this assumption does not hold in noisy scenes where imbalance label noise exists. In our analysis, we discovered imbalanced label noise distributions also lead to the imbalanced detection problem. To empirically verify our analysis and investigate whether or not there are other factors lead to imbalanced detection, we conducted experiments using several general detection frameworks on Balance18 and VOC2007Noise datasets, which have balanced data distributions but imbalanced label noise distribution. We choose three well-performing general detection frameworks with various

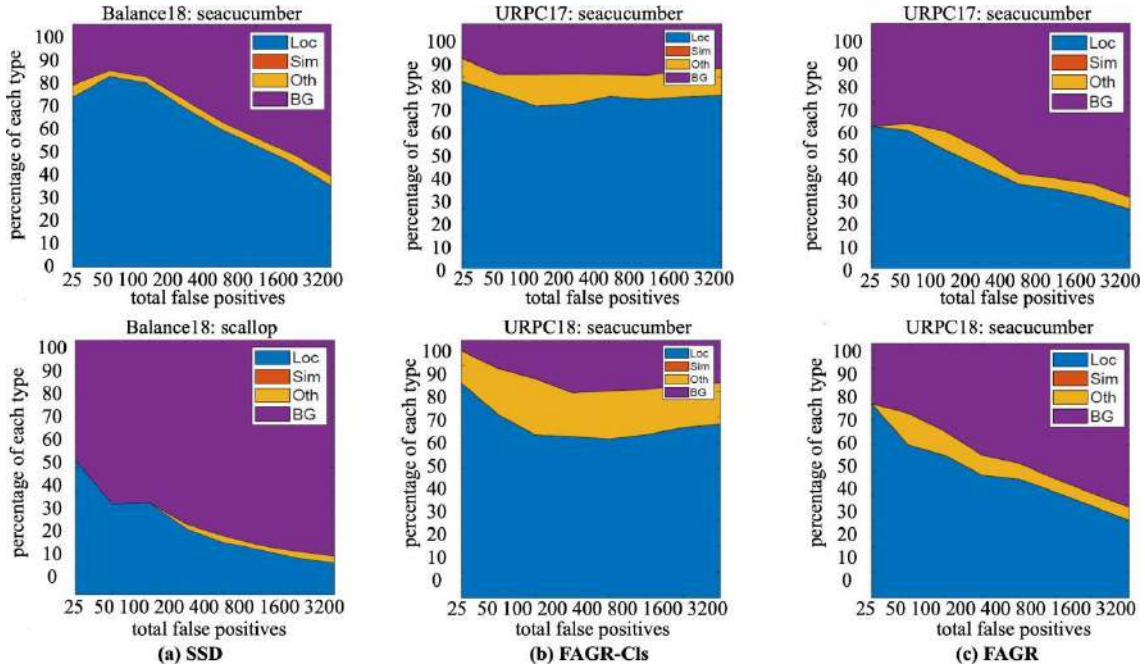

Fig. 5.4 The distribution of the false positive types of SSD (a) on Balance18, FAGR-Cls (b) and FAGR (c) on URPC17 and URPC18.

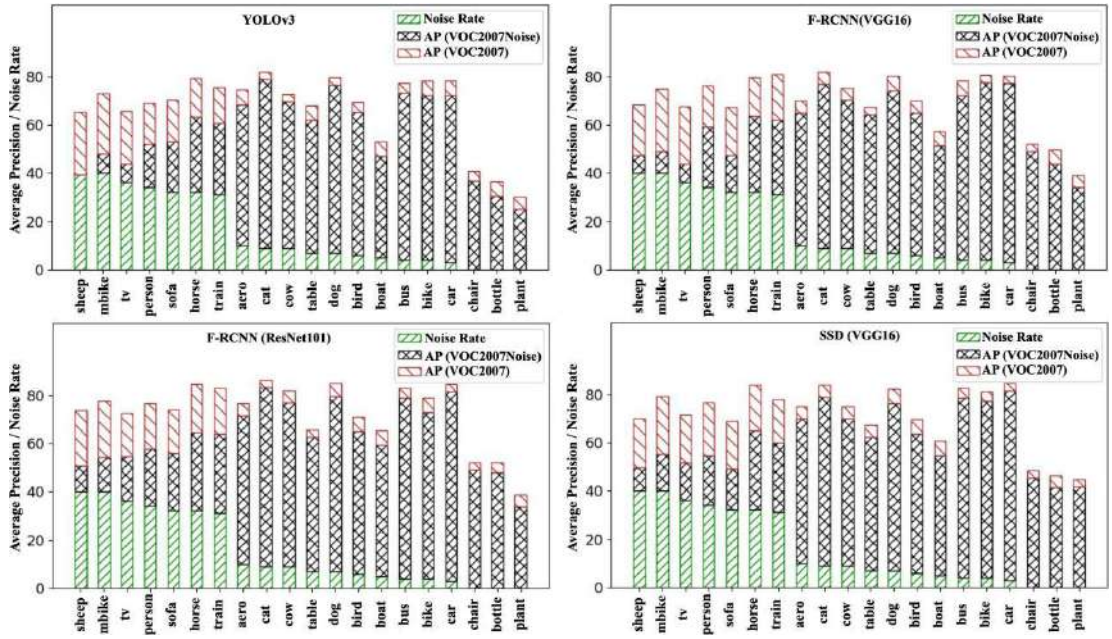

Fig. 5.5 The average precision (AP) of each class achieved by different detection networks on VOC2007Noise and VOC2007. The noise rates of the classes have been visualised using the green colors.

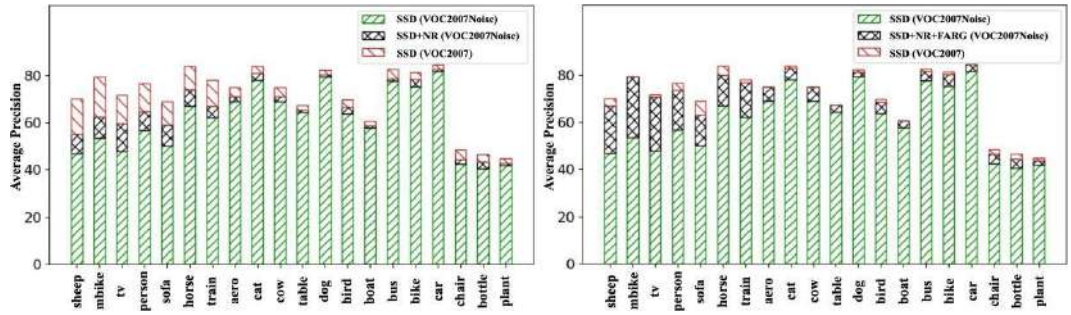

Fig. 5.6 Performance of SSD with and without NR (left) and FARG (right) on VOC2007Noise. We also list the performance of SSD on VOC2007 as the reference.

Dataset	Method	σ	θ_1	θ_2	θ_3	θ_4	seacucumber	seaurchin	scallop	starfish	mAP
URPC2017	SSD	-	-	-	-	-	38.4	52.9	15.7	-	35.7
	NR	0.1	10.5%	2.1%	12.2%	-	46.5	52.0	38.0	-	45.5
	NR	0.2	25.6%	5.3%	31.4%	-	49.8	54.5	47.1	-	50.5
	NR	0.3	39.7%	18.7%	42.3%	-	47.4	49.4	45.7	-	47.5
	FAGR	-	-	-	-	-	50.6	60.2	49.8	-	53.5
	NR+FAGR	0.2	25.6%	5.3%	31.4%	-	55.7	60.5	55.4	-	57.2
URPC2018	SSD	-	-	-	-	-	44.2	84.4	35.8	78.1	60.6
	NR	0.1	8.5%	2.6%	11.7%	2.2%	55.6	80.0	52.7	75.9	66.1
	NR	0.2	15.8%	3.5%	18.6%	2.9%	59.0	79.8	56.9	76.4	68.0
	NR	0.3	22.3%	10.3%	27.6%	12.5%	54.9	76.5	50.2	73.8	63.9
	FAGR	-	-	-	-	-	67.8	78.6	66.6	72.2	71.3
	NR+FAGR	0.2	15.8%	3.5%	18.6%	2.9%	71.6	79.0	**71.4**	73.6	**73.9**

Table 5.2 Impacts of NR and σ on our proposed detection network. θ_1, θ_2, θ_3 and θ_4 denote the percentage of training samples having been removed for the seacucumber, seaurchin, scallop and starfish classes under the threshold σ.

Setting				URPC17			
Method	Cls	Loc	α_1:α_2	seacucumber	seaurchin	scallop	mAP
FAGR-No	×	×	-	38.4	52.9	15.7	35.7
FAGR-Loc	✓	×	-	48.4	58.6	54.8	53.9
FAGR-Cls	×	✓	-	55.9	59.7	47.6	54.4
FAGR	✓	✓	1:1	55.7	60.5	55.4	**57.2**
FAGR	✓	✓	1:2	56.1	58.2	51.2	55.2
FAGR	✓	✓	2:1	51.8	**61.0**	**56.5**	56.4
FAGR	✓	✓	1:3	**59.5**	60.2	50.5	56.7
FAGR	✓	✓	3:1	52.7	59.9	56.4	56.3

Table 5.3 Impacts of FAGR on our proposed detection network. FAGR-Cls/FAGR-Loc indicates that FAGR is only added on the classification/localisation head. α_1 and α_2 are the weight terms of the classification and localisation heads, respectively.

Setting				URPC18				
Method	Cls	Loc	α_1:α_2	seacucumber	seaurchin	scallop	starfish	mAP
FAGR-No	×	×	-	44.2	**84.4**	35.8	**78.1**	60.6
FAGR-Loc	✓	×	-	66.9	78.2	68.4	72.4	71.5
FAGR-Cls	×	✓	-	69.5	77.9	62.1	70.9	70.1
FAGR	✓	✓	1:1	71.6	79.0	**71.4**	73.6	**73.9**
FAGR	✓	✓	1:2	72.2	79.3	68.8	70.3	72.7
FAGR	✓	✓	2:1	68.8	79.2	68.1	73.0	72.3
FAGR	✓	✓	1:3	**75.8**	79.9	65.8	72.4	73.5
FAGR	✓	✓	3:1	66.6	80.0	70.1	72.6	72.3

Table 5.4 Impacts of FAGR on our proposed detection network. FAGR-Cls/FAGR-Loc indicates that FAGR is only added on the classification/localisation head. α_1 and α_2 are the weight terms of the classification and localisation heads, respectively.

backbones, including YOLOv3 [26], F-RCNN [169] with VGG16, ResNet50 and ResNet101 backbones, and SSD [33].

Table 5.1 reports the performance of different detection frameworks on Balance18. Balance18 has balanced data distributions for all the object classes, however, all of the detection networks still suffer from the severe imbalanced detection problem. The precision of the seacucumber and scallop classes is far behind that of the seaurchin class (more than 10% AP for all of the detection networks). The empirical evidence shows that imbalanced detection is not solely caused by the imbalanced data distributions.

To further explore the factors that cause the low precision of the seacucumber and scallop classes on Balance18, we applied the detection analysis tool of [128] to analysing the false positives of the best performing detector SSD. The false positives are due to four error types: (1) Localisation error caused by insufficient overlaps with the ground-truth annotations (Loc); (2) confusion with similar classes (Sim); (3) confusion with other classes (Oth); (4) confusion with the background class (BG). We present false positives of these two classes in Figure 5.3, where SSD frequently treats the stones and sands as the scallops and cannot localise the sea cucumbers accurately. Figure 5.4(a) also shows the distribution of the false positive types of SSD for the seacucumber and scallop classes on Balance18. We witness that, without label noise removal, SSD has more localisation errors on the seacucmber class

and more background errors on the scallop class. For the seacucumber class, from the outcomes of the underwater datasets, we observe that they follow the gregarious behaviors and severe occlusions exist in the captured images. Hence, many label noise has been inevitablely generated during the annotation stage due to the inaccurate location annotation. For the scallops, some of them are very similar to the background. In the annotation, many background targets have been wrongly labelled as the scallop class, and this label noise confuses the detector that cannot distinguish the scallops from the background. This explains why SSD yields more background errors and produces low precision on the scallop class. After applying the NR algorithm to SSD, we observed that the detection frameworks SSD+NR and SSD+NR+FARG achieved much balanced precision for all the object classes as shown in Table 5.1. This empirical evidence reveals that the imbalanced label noise distribution may exaggerate the imbalanced detection problem in the noisy imbalanced dataset.

Figure 5.5 presents the performance of the general detection networks trained on VOC2007Noise and VOC2007. On VOC2007Noise, all of the detection networks suffer from the serious imbalanced detection problem even though VOC2007Noise has a balanced data distribution. VOC2007Noise is generated by manually adding imbalanced label noise into different classes of VOC2007. From Figure 5.5, we observe that classes with higher label noise rates suffer from higher AP decrease. This demonstrates that the performance decrease comes from the label noise, and the imbalanced label noise distribution enlarges the precision discrepancies of different classes. This is because the label noise disturbs the feature learning of the noisy classes, and the features learned from the deep network are suitable to the clean classes (i.e., classes with less label noise) but may not be suitable to the noisy classes (i.e., classes with considerable label noise). Moreover, we discover that some clean classes, such as the plant, bottle and chair classes, still generate low precision on the clean VOC2007 data set. We believe the low precision arises from the classification and localisation difficulties of the classes themselves.

We also apply the NR and FARG algorithm to training SSD on VOC2007Noise, and the performance comparison of SSD with and without NR and FARG algorithms on VOC2007Noise are presented in Figure 5.6. We observe that the imbalanced detection

Dataset	URPC17						
Method	seacucumber	seaurchin	scallop	mAP	AP_r	AP_c	AP_f
NoR	38.4	52.9	15.7	35.7	27.1	-	52.9
FRR [69]	44.5	51.0	38.8	44.8	41.7	-	51.0
ENR [68]	43.9	52.9	41.5	46.1	42.7	-	52.9
Focal [54]	19.2	72.9	28.9	40.3	24.1	-	72.9
BAGS [73]	49.5	57.5	49.9	52.3	49.7	-	57.5
FreeAnchor [151]	21.8	74.7	27.7	41.4	24.8	-	74.7
GHM [152]	23.0	74.3	33.9	43.7	28.5	-	74.3
FCOS [148]	23.6	**75.2**	33.8	44.2	28.7	-	**75.2**
IMA [33]	44.4	52.4	42.1	46.3	43.3	-	52.4
CMA [7]	49.1	62.3	46.1	52.5	47.6	-	62.3
Ours	**55.7**	60.5	**55.4**	**57.2**	**55.6**	-	60.5

Table 5.5 Comparisons with different imbalance algorithms and detection frameworks on URPC2017. AP_r, AP_c and AP_f indicate the AP for the rare, common and frequent classes, respectively.

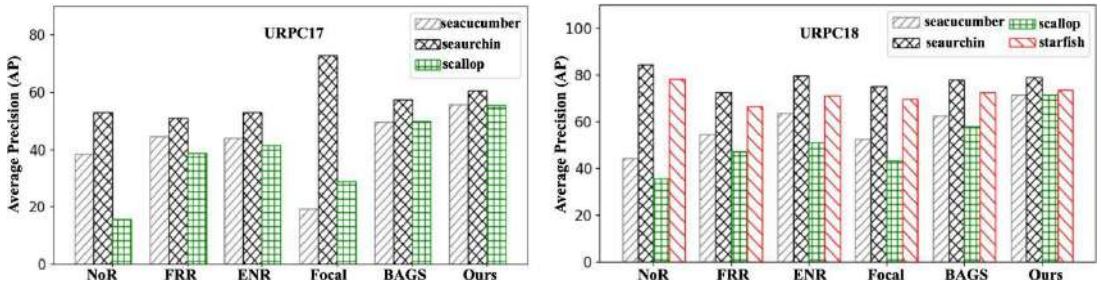

Fig. 5.7 The average precision of each class achieved by different class-imbalance algorithms on URPC17 and UPRC18.

problem has been alleviated after removing label noise, and the detection network SSD+NR trained on the clean training data achieves relative balanced precision for all the classes. However, large precision discrepancies still exist in different classes. This is because the NR algorithm removes many training samples of the noisy classes and this generates an imbalanced data distribution. After deploying both NR and FAGR algorithms, SSD achieved very similar performance on VOC2007Noise and VOC2007, demonstrating that imbalanced detection has been well-addressed by NR and FAGR.

Dataset	URPC18							
Method	seacucumber	seaurchin	scallop	starfish	mAP	AP_r	AP_c	AP_f
NoR	44.2	84.4	35.8	78.1	60.6	40.0	78.1	84.4
FRR [69]	54.8	72.5	47.2	66.6	60.3	51.0	66.6	72.5
ENR [68]	63.7	79.6	50.9	70.9	66.3	57.3	70.9	79.6
Focal [54]	52.5	74.9	43.1	69.8	60.1	47.8	74.9	69.8
BAGS [73]	62.6	77.9	57.9	72.5	67.7	60.3	72.5	77.9
FreeAnchor [151]	46.2	72.3	42.5	71.4	58.1	50.3	71.4	72.3
GHM [152]	52.4	78.4	42.1	71.5	61.1	51.6	71.5	78.4
FCOS [148]	43.2	76.5	47.5	69.4	59.1	53.3	69.4	76.5
IMA [33]	52.8	84.1	42.9	78.0	64.5	47.9	78.0	84.1
CMA [7]	56.4	**84.6**	50.9	**79.9**	68.0	53.7	**79.9**	**84.6**
Ours	**71.6**	79.0	**71.4**	73.6	**73.9**	**71.5**	73.6	79.0

Table 5.6 Comparisons with different imbalance algorithms and detection frameworks on URPC2018. AP_r, AP_c and AP_f indicate the AP for the rare, common and frequent classes, respectively.

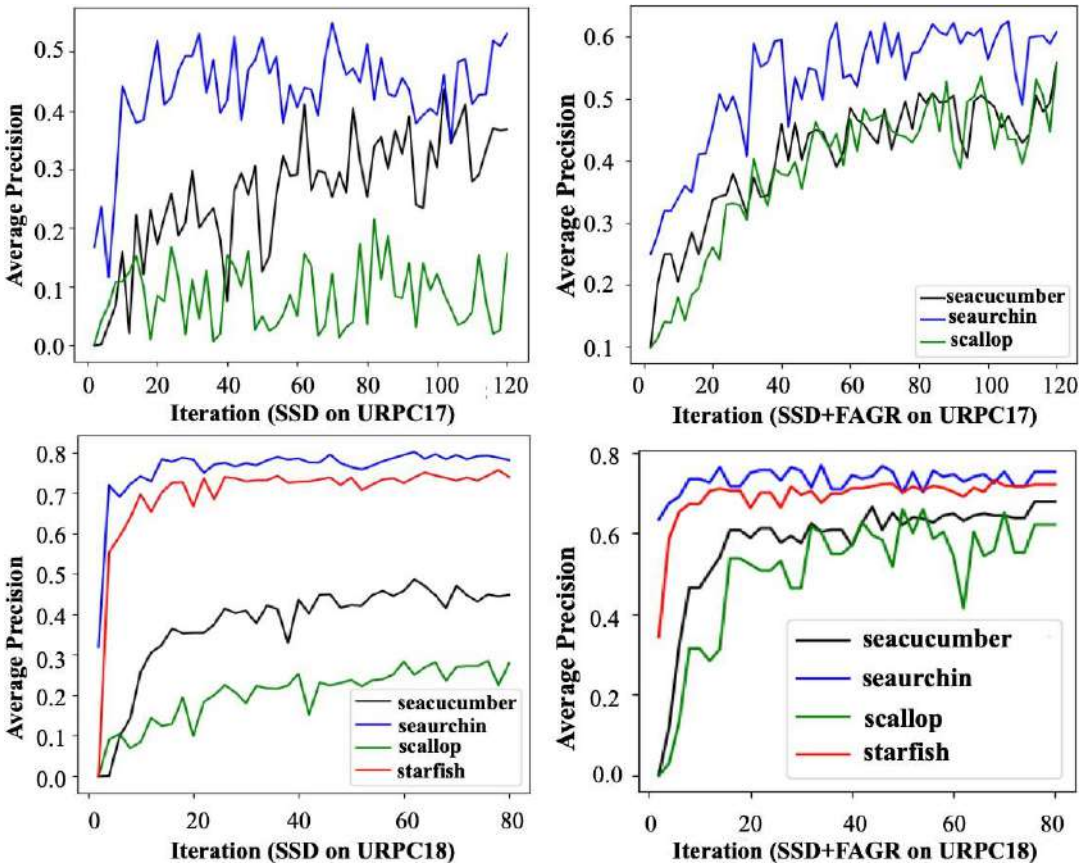

Fig. 5.8 The precision at each training iteration of SSD with and without FAGR on URPC17 and URPC18.

5.5 Experiments on URPC2017 and URPC2018

On the URPC2017 and URPC2018 datasets, we first conduct ablation studies to investigate the influences of NR and FAGR algorithms on our complete detection framework. Then, we compare our proposed method with other state-of-the-art (SOTA) class imbalance algorithms and detection networks.

5.5.1 Ablation Study of NR

We conduct ablation experiments to investigate the influence of NR algorithm on our final detection network. In addition, the predefined IoU threshold σ is an important hyper-parameter in the NR algorithm. For an object, if there is no prediction to match it with an IoU higher than σ, the object will be regarded as the noisy data and removed. Larger σ may remove more noisy and clean training samples while smaller σ may leave most of the noisy data in the training set. Hence, σ decides how many training samples will be removed. We denote β_1, β_2, β_3 and β_4 as the percentage of the training samples that are removed for the seacucumber, seaurchin, scallop and starfish classes under the setting of σ.

Table 5.2 shows the impacts of NR and σ on our proposed detection framework. SSD with the NR algorithm (SSD+NR) performs much better than SSD without the NR algorithm due to the removal of the label noise. The set of σ is of vital importance to the final performance. Using small σ =0.1, most of the noisy data have not been removed that still hurts the precision of the detection network. Using large σ =0.3, excessive training samples have been removed including the clean training samples, leading to the loss of considerable effective training samples. On the two datasets, the detection network achieves the best performance when σ =0.2. Moreover, we found most of the removed noisy data come from the scallop and seacucumber classes, that explains why these two classes achieve much lower precision than the other classes.

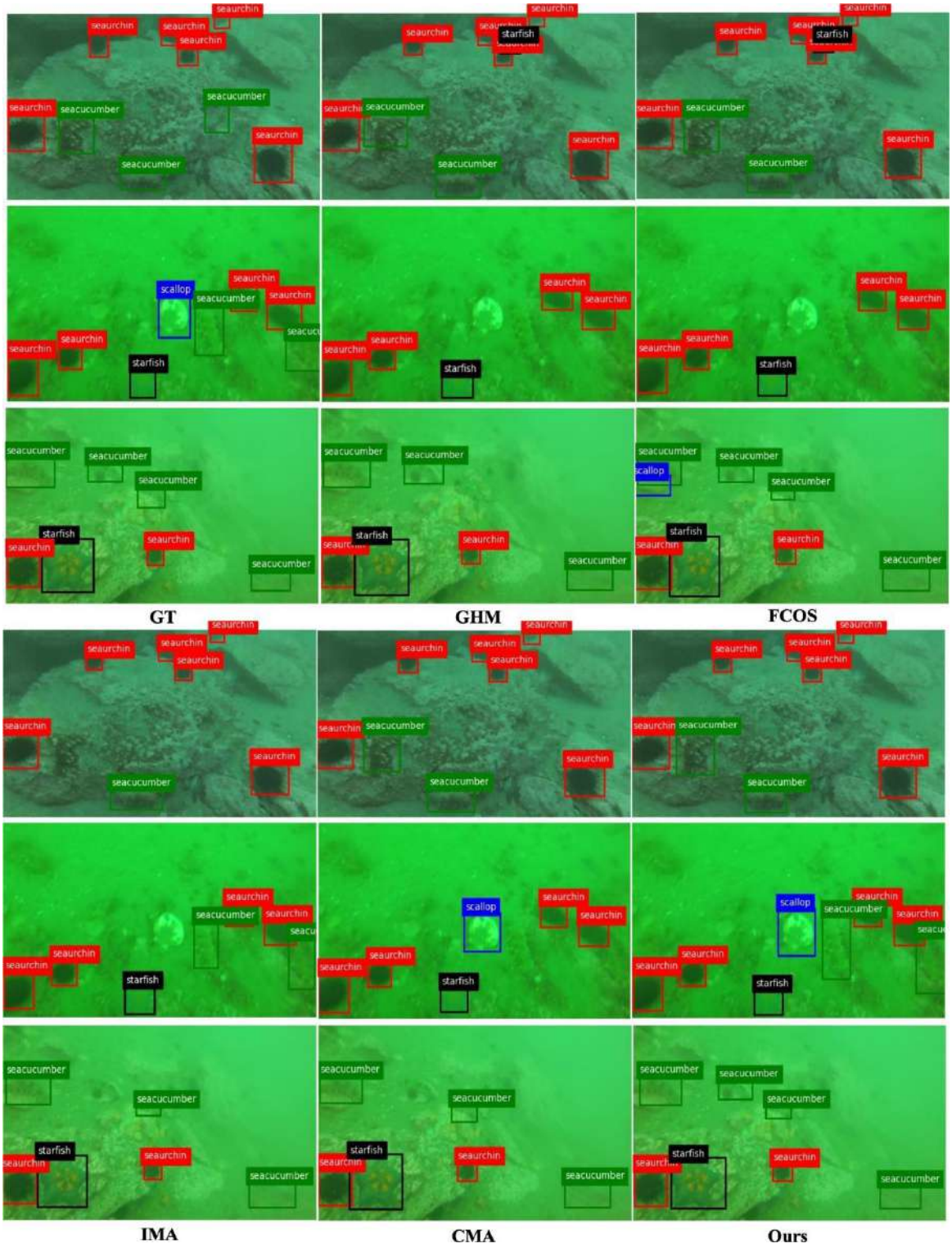

Fig. 5.9 Visualisation comparison with the SOTA detection frameworks on URPC17.

5.5.2 Ablation Study of FARG

Current class imbalance algorithms designed for object detection, such as BAGS [73] and Focal loss [54], only improve the classification head of the detection network without improving the localisation head. Hence, we conduct ablation experiments to investigate whether or not adding FAGR onto the localisation head helps us to address the imbalanced detection problem. We denote FAGR-Cls/FAGR-Loc as FAGR is only added onto the classification/localisation heads, and FAGR-No indicates the detection network without using the FAGR algorithm. In addition, we conduct experiments to choose the best hyper-parameters α_1 and α_2, i.e., the weight terms of the classification and localisation heads, respectively.

Table 5.4 shows the impacts of FAGR on our proposed detection framework. We observe that applying FAGR on both the heads helps us to boost the detection precision, which demonstrates that FAGR is able to enhance the classification and localisation accuracy for the under-represented classes. FAGR increases the weights of the under-represented classes so that the deep network biases to learn better features representations to achieve accurate classification and localisation. FAGR performs much better than FAGR-Cls and FAGR-Loc. Figure 5.4(b) and (c) also show the distribution of the false positives of FAGR and FAGR-Cls for the seacucumber class, which contains much more label noise due to the inaccurate location annotations in the URPC17 and URPC18 datasets. We observe that FAGR makes much less localisation errors than FAGR-Cls, because FAGR magnifies the weights of the under-represented classes in the localisation branch so that the features related to the under-represented classes have been enhanced. Moreover, FAGR achieves the best performance when we set α_1 equal to α_2, and all these experimental results demonstrate that improving the localization head is as important as improving the classification head for the imbalanced detection problem.

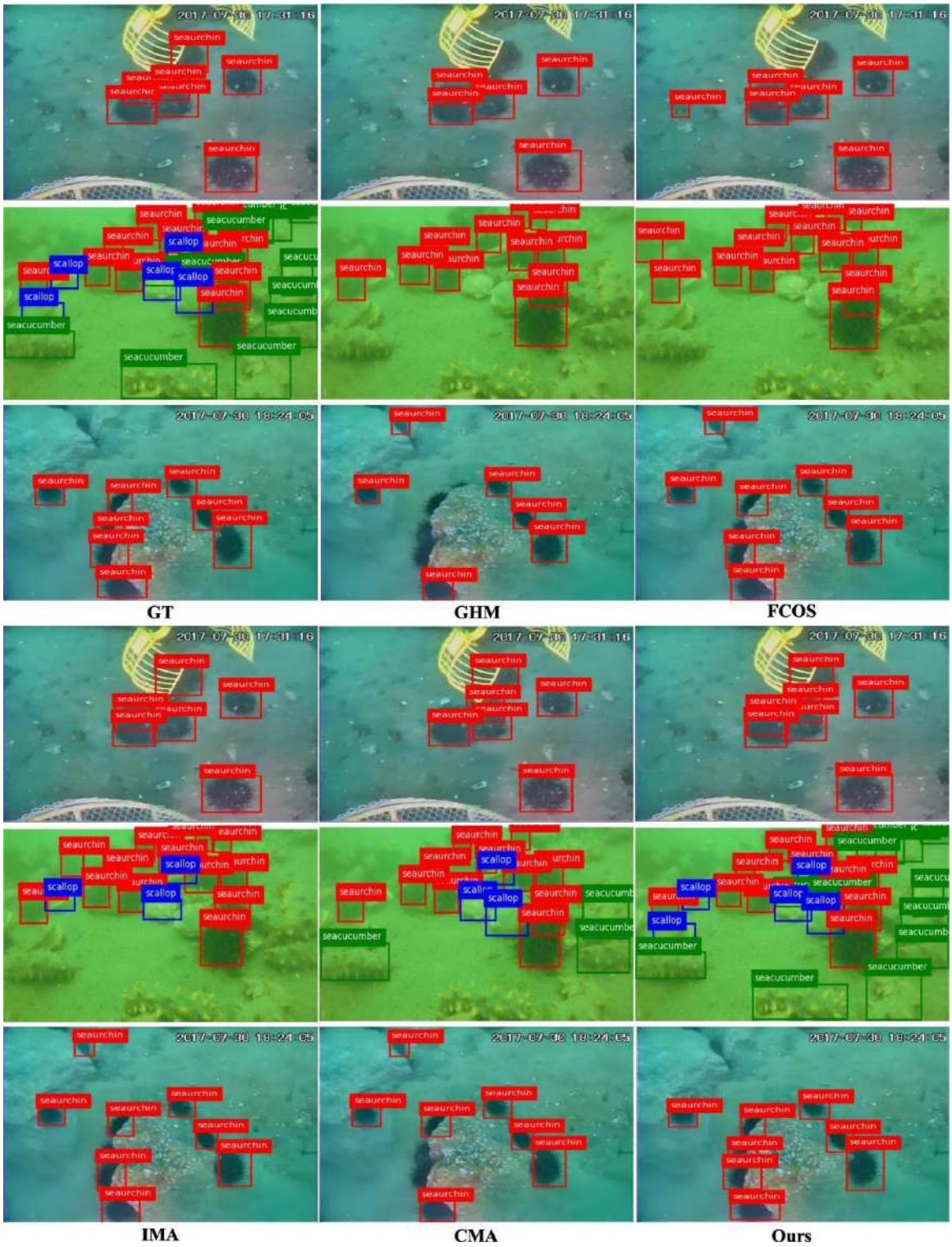

Fig. 5.10 Visualisation comparison with other state-of-the-arts detection frameworks on URPC18.

5.5.3 Comparison with Class-imbalance Algorithms

Following [73], in addition to metrics including mean Average Precision (mAP), AP_r (AP for rare classes), AP_c (AP for common classes), and AP_f (AP for frequent classes) have also been reported when compared with other SOTA class imbalance algorithms. According to the training instance numbers, we take the seacucumber and scallop classes as rare classes, the starfish class as the common class and the seaurchin class as the frequent class. We transfer multiple class imbalance algorithms designed for classification to the SSD framework, including inverse class frequency-based re-balancing (FRR) algorithm [69] and the effective number-based re-balancing (ENR) algorithm [68]. We also deploy two SOTA class imbalance methods designed for object detection, including BAGS [73] and Focal loss [54]. For all the comparison methods, we train and test the detection networks on the underwater datasets, and choose the best hyper-parameter settings for the underwater datasets. We also present the results of the detector without re-balancing (NOR) as the baseline.

The performance of different class imbalance algorithms are shown in Table 5.6 and Figure 5.7, from which we observe that Focal holds the largest precision discrepancies among different classes. This is because the focal loss has been designed to emphasise on learning the hard training samples, however, the noisy data are also hard training samples. The deep network with focal loss may overfit on the noisy data and cannot generate accurate predications for the classes with more label noise. The ability of FRR and ENR to reduce the precision discrepancies is also limited, and large precision discrepancies still exist after the re-balancing operation. This is because the two algorithms re-balance the classes, however, other important interfering factors, such as imbalance label noise, have not been addressed by FRR and ENR. Among the comparison methods, BAGS achieves the best balanced AP, although its performance falls behind that of our proposed method. This is because BAGS only improves the classification head but ignores the localisation head, and the inherent localisation difficulties of the classes are largely different, resulting in imbalanced detection. From Figure 5.7, we observe our proposed FAGR algorithm achieves the best balanced average precision for all the classes, whilst re-weighting all the classes fairly and learning optimal features for all the classes. We also present the precision in each training iteration of

SSD with and without FAGR in Figure 5.8, from which we observe that FAGR gradually minimises the precision discrepancies between different classes during training, and the learning biases in the imbalanced dataset have been well-addressed by FAGR.

5.5.4 Comparison with SOTA Detection Frameworks

In this paper, we also compare the proposed framework with several top performing detection frameworks on URPC17 and URPC18, including two SOTA detection frameworks (CMA [7] and IMA [33]) particularly designed for underwater object detection, and three top-performing general detection frameworks FreeAnchor [151], GHM [152] and FCOS [148]. The performance of different detection networks are shown in Table 5.6, among the comparison methods, IMA and CMA perform much better than the other detection networks. This is because these two detection frameworks explicitly alleviates the influence of the noisy data and ensembles the results of several detectors for complementary performance. CMA also achieves better performance than IMA because CMA is able to make full use of all the available training instances while IMA treats uncertain training samples as noise instead of learning them, leading to the loss of possible clean training instances. In our proposed detection framework, we apply the Noise-Removal (NR) strategy to remove possible noisy data, different from IMA, we remove a few training samples using the IoU threshold σ, that avoids excessive loss of training samples. The existence of label noise greatly degrades the performance of the general detection networks FreeAnchor, GHM and FCOS, which do not learn robust features for the classes with label noise. As shown in Figures 5.9 and 5.10, the general detection networks GHM and FCOS almost cannot detect scallops and seacucumbers. These two classes have more label noise that diverts the general detection networks to learn discriminative features. Moreover, all the detection networks without a class imbalance algorithm suffer from the severe class imbalance problem: the feature learning biases to the classes which have more training samples and less label noise, which achieve much higher precision than the others. Our proposed NR reduces the influence of label noise, and our FAGR applies the gradient-adjustment coefficients to increasing the gradient magnitudes of the low-precision classes, encouraging the detection networks to learn better feature

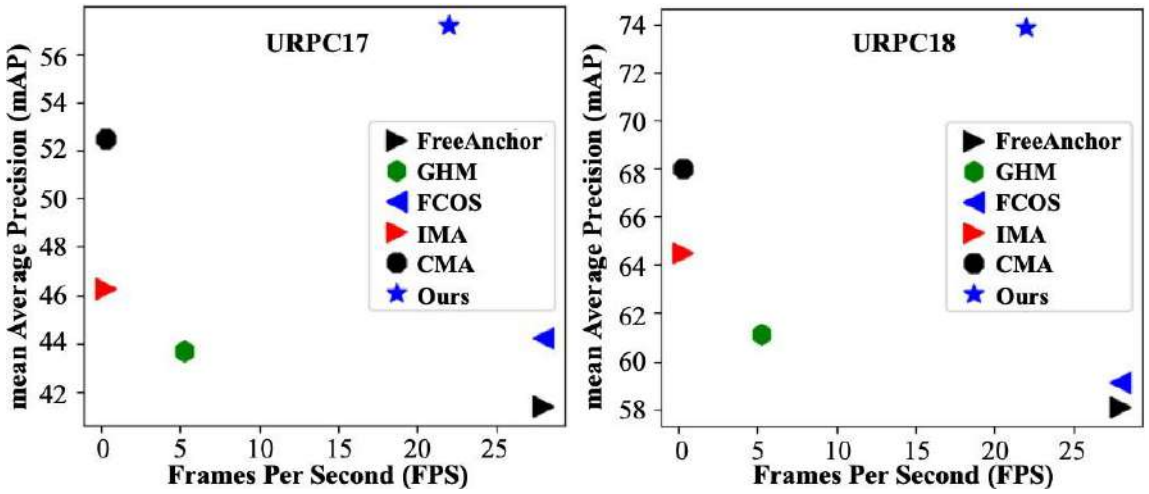

Fig. 5.11 The running time of different detection networks on URPC17 and URPC18.

representations. FAGR re-weights the classes implicitly considering the combinations of all possible factors. This explains why FAGR greatly improves the generalisation performance of the detection network even though it has not explicitly considered all possible factors.

Figure 5.11 shows the running time comparisons of different detection networks. All the experiments are conducted on a single NVIDIA Tesla P100 GPU with a 16GB memory. Our proposed method runs at the speed of 22 FPS with the input image size of 512 ×512, which are much faster than the other SOTA underwater object detection frameworks IMA and CMA. Among all the detection frameworks, FCOS and FreeAnchor are able to achieve a real-time running speed, but our proposed detection framework achieves the best speed-precision trade-off. Our proposed algorithm outperforms the other detection frameworks on mean Average Precision (57.2% mAP on URPC17 and 73.9% mAP on URPC18) while maintaining a pseudo real-time speed.

5.6 Summary

In summary, we provide theoretical analysis and empirical evidence that imbalance data distributions are not the only cause of imbalance detection. As we know, this is the first work to report that imbalanced label noise distributions also lead to the imbalance detection

problem. To alleviate the influence of label noise on the detection neural network, we propose a noise removal (NR) algorithm to filter out label noise in the datasets. We propose a factor-agnostic gradient re-weighting (FAGR) algorithm to address the imbalance detection problem. FAGR produces the precision balanced gradients of all the classes and re-balance the precision distributions. Extensive experiments show that our proposed framework, by re-balancing precision distributions, performs much better than those of re-balancing data distributions (see Figure 5.1).

Chapter 6

Conclusions and Perspectives

In this thesis, we have presented and evaluated several contributions for underwater object detection, undrwater image enhancement and underwater image synthesis. To conclude our work, we summarise our key contributions and discuss our experimental findings. Finally, we present future perspectives, indicating interesting directions for future research in this field.

6.1 Key Contributions

In this thesis, we aim at developing intelligent computer vision systems for robust underwater object detection and underwater image enhancement. To achieve high performance and generalisation capacity, we developed several deep learning techniques in our research. We have reviewed a large number of previous works, discussed the advantages and disadvantages of these works, and aimed to address the limitations of these works. Moreover, we have identified several challenges, such as the noisy data problem and the class imbalance problem, that hinder the development of underwater image enhancement and underwater object detection. In this thesis, we have proposed several novel solutions to address these challenges in underwater object detection. The contribution of this thesis can be summarised as follows:

First, we have proposed a new solution for the underwater image enhancement task. Different from previous works, the objective of our proposed underwater image enhancement method is to assist the later underwater object detection task rather than improving the visual

quality of the degraded underwater images. To achieve this objective, we have proposed two detection perceptual enhancement models, each of which consists of an enhancement model and a detection perceptor. One enhancement model can generate visually pleasing images on the patch-level, while the other one can generate detection-favouring images that help improving the detection accuracy. To our knowledge, this is the first practice for underwater image enhancement, aiming to generate detection-favouring rather than visually pleasing images. Moreover, we have proposed a novel underwater image synthesis model for generating more training data for the underwater image enhancement model. Different from previous works, our proposed hybrid underwater image synthesis model incorporates both physical priors and data-driven cues. The hybrid synthesis model fully takes into account image characteristics such as color distortion, haze-effects and diversity, enabling our perceptual enhancement models to be generalised to handle real-world underwater scenes. The proposed hybrid synthesis and perceptual enhancement models are incorporated into a unified framework named HybridDetectionGAN, and can be jointly optimised in an end-to-end pattern. We evaluated the underwater image enhancement algorithms using extensive evaluation metrics, including four full-reference image quality evaluation metrics (MSE, PSNR, SSIM and PCQI) and two evaluation metrics (UCIQE and UIQM). These metrics are complementary to each other and provide reliable quantitative evaluations of different underwater image enhancement algorithms. The extensive evaluations show our proposed perceptual enhancement models outperform several state-of-the-art UIE algorithms on both synthetic and real-world underwater datasets.

Second, the performance of underwater object detection frameworks is greatly degraded by the existence of noisy data in underwater object detection datasets, hence, we focus on addressing the noisy data problem in Chapter 4. We offer a compelling insight on the training strategy of deep detectors in underwater scenes where noisy data exists. Specially, we have proposed a novel noise-immune deep detection framework which consists of a backbone network SWIPENET and a powerful training paradigm CMA. The SWIPENET+CMA framework trains a robust deep ensemble detector for the object detection task in the underwater scenes with heterogeneous noisy data and small objects. SWIPENET fully takes advantage of

both high resolution and semantic-rich Hyper Feature Maps that significantly boost small object detection. Moreover, a novel sample-weighted detection loss is designed for the proposed SWIPENET, which controls the influence of the training samples on SWIPENET according to their weights. We also provided theoretical analysis on the ability of the sample-weighted detection loss in detail. To achieve the balance between the detection accuracy and the computational cost, we proposed a selective ensemble algorithm to choose the best detector trained with large data diversity. Experiments on four underwater object detection datasets showed that the proposed SWIPENET+CMA framework achieved better or competitive accuracy in object detection against several state-of-the-art approaches.

Third, the class imbalance problem is another serious challenge for underwater object detection framework. From the optimisation perspective, we have discovered and provided evidence for the fact that imbalanced data distributions are not the only factor leading to the imbalance detection problem but imbalanced noise distributions also contribute to the problem. To address imbalance detection problems, we proposed a noise removal (NR) algorithm to remove label noise, and then a factor-agnostic gradients re-weighting (FAGR) algorithm to re-weight the classes according to the precision distributions. We have demonstrated that re-balancing the precision distribution brings significant generalisation improvements to deep detection networks than re-balancing the data distribution. This attempt provides a new perspective for addressing the imbalanced detection problem in noisy imbalanced scenarios. FAGR implicitly considers the integrating effects of all possible factors leading to the class imbalance problem, thus it is a general and simple solution to address the imbalance detection problem. Extensive experiments on three underwater object datasets and one general object detection dataset demonstrate the effectiveness of our proposed NR+FAGR algorithm in handling the class imbalance problem in noisy underwater object detection datasets.

6.2 Future Work

In this thesis, we have presented our new solutions on underwater image enhancement and underwater object detection. Our proposed frameworks aim to achieve robust enhancement and detection performance in underwater scenes, and extensive experimental results show they can achieve high accuracy on related tasks and have potential to facilitate the research of ocean scientists and biologists. However, there is still some space for improvement in our future work.

First, we need to propose a unified enhancement-detection framework, which combines low-level enhancement and high-level detection in an end-to-end framework. Our proposed enhancement and detection models carry out related tasks separately, i.e., we first apply image enhancement algorithms to the underwater images, followed by object detection using the enhanced images. This cascaded independent pipeline is not an optimal solution because the image transmission between the enhancement and detection models requires additional time expenses. Therefore, in our further work, we aim to construct a unified enhancement-detection framework, which combines low-level enhancement and high-level detection in an end-to-end framework instead of two cascaded independent processes. The end-to-end enhancement-detection framework may have two advantages: (1) Two models are jointly optimised that they can achieve better optimisation solution than the cascaded independent pipeline. (2) The running speed can be accelerated as the time expenses of the image transmission between the enhancement and detection models greatly decrease. Besides, we will propose a novel multi-task loss function to examine the unified enhancement-detection framework.

Second, the reduction in computational complexity is highly demanded for real-time applications in the underwater scenes. In Chapter 4, we have proposed a deep detection framework which can well-handle the noise issue in the underwater scenes and achieves the state-of-the-art performance on the challenging underwater datasets. However, since it is a deep ensemble model, its time complexity is high. Moreover, although deep neural networks are very powerful, a large number of weights in DNNs consumes considerable storage and memory bandwidth. In the future, we plan to incorporate several model compression

algorithms into our framework to save considerable memory and computational costs. Model compression algorithms have been proposed to address efficient training and inference in deep neural networks. They attempt to reduce the size of a network by binarising the weights, pruning redundant and non-informative weights, and designing compact blocks. Inspired by previous works, we will propose a deep learning compression algorithm to reduce the storage and energy required to run inference on deep networks so that they can be deployed on AUVs and ROVs for real time underwater applications.

Finally, we plan to extend our enhancement-detection framework for underwater video analysis in the future. For videos and sequential underwater data analysis, it is often better to use Recurrent Neural Networks (RNNs). Compare to CNNs, the hidden layers in RNNs are self-connected, which means the information feedback of the hidden layer is not just imported to the output layer, but also imported into the hidden layer of the next input of the sequence. In RNNs, the output of the current input is related to the information learned from the previous sequential data, thus gives RNNs the ability to use their internal memory to process arbitrary sequences of inputs. Hence, we plan to combine CNNs and RNNs based architectures into a unified deep learning framework for end-to-end underwater video analysis.

References

[1] T. Li, Q. Yang, S. Rong, L. Chen, and B. He, "Distorted underwater image reconstruction for an autonomous underwater vehicle based on a self-attention generative adversarial network," *Applied Optics*, vol. 59, no. 32, pp. 10 049–10 060, 2020.

[2] M. Moniruzzaman, S. M. S. Islam, M. Bennamoun, and P. Lavery, "Deep learning on underwater marine object detection: A survey," in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 2017, pp. 150–160.

[3] Y. Zhou, Q. Wu, K. Yan, L. Feng, and W. Xiang, "Underwater image restoration using color-line model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 3, pp. 907–911, 2018.

[4] D. Mallet and D. Pelletier, "Underwater video techniques for observing coastal marine biodiversity: a review of sixty years of publications (1952–2012)," *Fisheries Research*, vol. 154, pp. 44–62, 2014.

[5] A. Sahoo, S. K. Dwivedy, and P. Robi, "Advancements in the field of autonomous underwater vehicle," *Ocean Engineering*, vol. 181, pp. 145–160, 2019.

[6] I. Carlucho, M. De Paula, S. Wang, Y. Petillot, and G. G. Acosta, "Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning," *Robotics and Autonomous Systems*, vol. 107, pp. 71–86, 2018.

[7] L. Chen, F. Zhou, S. Wang, J. Dong, N. Li, H. Ma, X. Wang, and H. Zhou, "Swipenet: Object detection in noisy underwater images," *arXiv preprint arXiv:2010.10006*, 2020.

[8] P. I. Macreadie, D. L. McLean, P. G. Thomson, J. C. Partridge, D. O. Jones, A. R. Gates, M. C. Benfield, S. P. Collin, D. J. Booth, L. L. Smith *et al.*, "Eyes in the sea: unlocking the mysteries of the ocean using industrial, remotely operated vehicles (rovs)," *Science of the Total Environment*, vol. 634, pp. 1077–1091, 2018.

[9] R. Liu, X. Fan, M. Zhu, M. Hou, and Z. Luo, "Real-world underwater enhancement: Challenges, benchmarks, and solutions under natural light," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 12, pp. 4861–4875, 2020.

[10] X. Ye, Z. Li, B. Sun, Z. Wang, R. Xu, H. Li, and X. Fan, "Deep joint depth estimation and color correction from monocular underwater images based on unsupervised adaptation networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 3995–4008, 2019.

[11] C. Li, S. Anwar, and F. Porikli, "Underwater scene prior inspired deep underwater image and video enhancement," *Pattern Recognition*, vol. 98, p. 107038, 2020.

[12] J. Y. Chiang and Y.-C. Chen, "Underwater image enhancement by wavelength compensation and dehazing," *IEEE transactions on image processing*, vol. 21, no. 4, pp. 1756–1769, 2011.

[13] Z. Wang, C. Liu, S. Wang, T. Tang, Y. Tao, C. Yang, H. Li, X. Liu, and X. Fan, "Udd: an underwater open-sea farm object detection dataset for underwater robot picking," *arXiv preprint arXiv:2003.01446*, 2020.

[14] O. Beijbom, P. J. Edmunds, D. I. Kline, B. G. Mitchell, and D. Kriegman, "Automated annotation of coral reef survey images," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 1170–1177.

[15] D. Kim, D. Lee, H. Myung, and H.-T. Choi, "Artificial landmark-based underwater localization for auvs using weighted template matching," *Intelligent Service Robotics*, vol. 7, no. 3, pp. 175–184, 2014.

[16] M.-C. Chuang, J.-N. Hwang, and K. Williams, "A feature learning and object recognition framework for underwater fish images," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1862–1872, 2016.

[17] K. Blanc, D. Lingrand, and F. Precioso, "Fish species recognition from video using svm classifier," in *Proceedings of the 3rd ACM International Workshop on Multimedia Analysis for Ecological Data*, 2014, pp. 1–6.

[18] S. Villon, M. Chaumont, G. Subsol, S. Villéger, T. Claverie, and D. Mouillot, "Coral reef fish detection and recognition in underwater videos by supervised machine learning: Comparison between deep learning and hog+ svm methods," in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 2016, pp. 160–171.

[19] A. Rova, G. Mori, and L. M. Dill, "One fish, two fish, butterfish, trumpeter: Recognizing fish in underwater video," in *MVA*, 2007.

[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[22] S. Choi, "Fish identification in underwater video with deep convolutional neural network: Snumedinfo at lifeclef fish task 2015." in *CLEF (Working Notes)*, 2015.

[23] X. Li, M. Shang, H. Qin, and L. Chen, "Fast accurate fish detection and recognition of underwater images with fast r-cnn," in *OCEANS 2015-MTS/IEEE Washington*. IEEE, 2015, pp. 1–5.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[25] H. Yang, P. Liu, Y. Hu, and J. Fu, "Research on underwater object recognition based on yolov3," *Microsystem Technologies*, vol. 27, no. 4, pp. 1837–1844, 2021.

[26] A. Farhadi and J. Redmon, "Yolov3: An incremental improvement," *Computer Vision and Pattern Recognition*, *cite as*, 2018.

[27] M. S. A. C. Marcos, M. N. Soriano, and C. A. Saloma, "Classification of coral reef images from underwater video using neural networks," *Optics express*, vol. 13, no. 22, pp. 8766–8771, 2005.

[28] M. Elawady, "Sparse coral classification using deep convolutional neural networks," *arXiv preprint arXiv:1511.09067*, 2015.

[29] O. Py, H. Hong, and S. Zhongzhi, "Plankton classification with deep convolutional neural networks," in *2016 IEEE Information Technology*, *Networking*, *Electronic and Automation Control Conference*. IEEE, 2016, pp. 132–136.

[30] H. Lee, M. Park, and J. Kim, "Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning," in *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016, pp. 3713–3717.

[31] J. Dai, R. Wang, H. Zheng, G. Ji, and X. Qiao, "Zooplanktonet: Deep convolutional network for zooplankton classification," in *OCEANS 2016-Shanghai*. IEEE, 2016, pp. 1–6.

[32] A. Mahmood, M. Bennamoun, S. An, F. Sohel, F. Boussaid, R. Hovey, G. Kendrick, and R. B. Fisher, "Coral classification with hybrid feature representations," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 519–523.

[33] L. Chen, Z. Liu, L. Tong, Z. Jiang, S. Wang, J. Dong, and H. Zhou, "Underwater object detection using invert multi-class adaboost with deep learning," in *2020 International Joint Conference on Neural Networks*, *IJCNN 2020*. Institute of Electrical and Electronics Engineers (IEEE), 2020.

[34] W.-H. Lin, J.-X. Zhong, S. Liu, T. Li, and G. Li, "Roimix: Proposal-fusion among multiple images for underwater object detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics*, *Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 2588–2592.

[35] B. Fan, W. Chen, Y. Cong, and J. Tian, "Dual refinement underwater object detection network," in *Computer Vision–ECCV 2020: 16th European Conference*, *Glasgow*, *UK*, *August 23–28*, *2020*, *Proceedings*, *Part XX 16*. Springer, 2020, pp. 275–291.

[36] M. P. Hayes and P. T. Gough, "Broad-band synthetic aperture sonar," *IEEE Journal of Oceanic Engineering*, vol. 17, no. 1, pp. 80–94, 1992.

[37] E. Galceran, V. Djapic, M. Carreras, and D. P. Williams, "A real-time underwater object detection algorithm for multi-beam forward looking sonar," *IFAC Proceedings Volumes*, vol. 45, no. 5, pp. 306–311, 2012.

[38] N. Strachan, "Recognition of fish species by colour and shape," *Image and vision computing*, vol. 11, no. 1, pp. 2–10, 1993.

[39] D.-J. Lee, R. B. Schoenberger, D. Shiozawa, X. Xu, and P. Zhan, "Contour matching for a fish recognition and migration-monitoring system," in *Two-and Three-Dimensional Vision Systems for Inspection*, *Control*, *and Metrology II*, vol. 5606. International Society for Optics and Photonics, 2004, pp. 37–48.

[40] C. Spampinato, Y.-H. Chen-Burger, G. Nadarajan, and R. B. Fisher, "Detecting, tracking and counting fish in low quality unconstrained underwater videos." *VISAPP (2)*, vol. 2008, no. 514-519, p. 1, 2008.

[41] R. Larsen, H. Olafsdottir, and B. K. Ersbøll, "Shape and texture based classification of fish species," in *Scandinavian Conference on Image Analysis*. Springer, 2009, pp. 745–749.

[42] P. X. Huang, B. J. Boom, and R. B. Fisher, "Underwater live fish recognition using a balance-guaranteed optimized tree," in *Asian Conference on Computer Vision*. Springer, 2012, pp. 422–433.

[43] X. Li, M. Shang, J. Hao, and Z. Yang, "Accelerating fish detection and recognition by sharing cnns with objectness learning," in *OCEANS 2016-Shanghai*. IEEE, 2016, pp. 1–5.

[44] L. Chen, Z. Jiang, L. Tong, Z. Liu, A. Zhao, Q. Zhang, J. Dong, and H. Zhou, "Perceptual underwater image enhancement with deep learning and physical priors," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[45] M. Jian, Q. Qi, J. Dong, Y. Yin, W. Zhang, and K.-M. Lam, "The ouc-vision large-scale underwater image database," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2017, pp. 1297–1302.

[46] M. Jian, Q. Qi, J. Dong, Y. Yin, and K.-M. Lam, "Integrating qdwd with pattern distinctness and local contrast for underwater saliency detection," *Journal of visual communication and image representation*, vol. 53, pp. 31–41, 2018.

[47] Z. Zhang, H. Zhang, S. O. Arik, H. Lee, and T. Pfister, "Distilling effective supervision from severe label noise," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9294–9303.

[48] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, "Learning from noisy labels with distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1910–1918.

[49] K.-H. Lee, X. He, L. Zhang, and L. Yang, "Cleannet: Transfer learning for scalable image classifier training with label noise," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5447–5456.

[50] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1944–1952.

[51] Y. Wang, W. Liu, X. Ma, J. Bailey, H. Zha, L. Song, and S.-T. Xia, "Iterative learning with open-set noisy labels," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8688–8696.

[52] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4334–4343.

[53] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 761–769.

[54] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[55] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models." in *NIPS*, vol. 1, 2010, p. 2.

[56] K. Ghasedi, X. Wang, C. Deng, and H. Huang, "Balanced self-paced learning for generative adversarial clustering network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4391–4400.

[57] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[58] G. T. Altmann, "Learning and development in neural networks–the importance of prior experience," *Cognition*, vol. 85, no. 2, pp. B43–B50, 2002.

[59] D. L. Rohde and D. C. Plaut, "Language acquisition in the absence of explicit negative evidence: How important is starting small?" *Cognition*, vol. 72, no. 1, pp. 67–109, 1999.

[60] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[61] J. Peng, X. Bu, M. Sun, Z. Zhang, T. Tan, and J. Yan, "Large-scale object detection in the wild from imbalanced multi-labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9709–9718.

[62] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance problems in object detection: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[63] M. Hayat, S. Khan, S. W. Zamir, J. Shen, and L. Shao, "Gaussian affinity for max-margin class imbalanced learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6469–6479.

[64] Y.-X. Wang, D. Ramanan, and M. Hebert, "Learning to model the tail," in *Advances in Neural Information Processing Systems*, 2017, pp. 7029–7039.

[65] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[66] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[67] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5375–5384.

[68] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9268–9277.

[69] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. Van Der Maaten, "Exploring the limits of weakly supervised pretraining," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 181–196.

[70] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[71] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, "Range loss for deep face recognition with long-tailed training data," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5409–5418.

[72] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2537–2546.

[73] Y. Li, T. Wang, B. Kang, S. Tang, C. Wang, J. Li, and J. Feng, "Overcoming classifier imbalance for long-tail object detection with balanced group softmax," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 991–11 000.

[74] A. Gupta, P. Dollar, and R. Girshick, "Lvis: A dataset for large vocabulary instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5356–5364.

[75] C. Ancuti, C. O. Ancuti, T. Haber, and P. Bekaert, "Enhancing underwater images and videos by fusion," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 81–88.

[76] X. Fu, P. Zhuang, Y. Huang, Y. Liao, X.-P. Zhang, and X. Ding, "A retinex-based enhancing approach for single underwater image," in *2014 IEEE international conference on image processing (ICIP)*. IEEE, 2014, pp. 4572–4576.

[77] X. Fu, Z. Fan, M. Ling, Y. Huang, and X. Ding, "Two-step approach for single underwater image enhancement," in *2017 international symposium on intelligent signal processing and communication systems (ISPACS)*. IEEE, 2017, pp. 789–794.

[78] K. Iqbal, M. Odetayo, A. James, R. A. Salam, and A. Z. H. Talib, "Enhancing the low quality images using unsupervised colour correction method," in *2010 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2010, pp. 1703–1709.

[79] A. S. A. Ghani and N. A. M. Isa, "Underwater image quality enhancement through integrated color model with rayleigh distribution," *Applied soft computing*, vol. 27, pp. 219–230, 2015.

[80] C. O. Ancuti, C. Ancuti, C. De Vleeschouwer, L. Neumann, and R. Garcia, "Color transfer for underwater dehazing and depth estimation," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 695–699.

[81] D. J. Jobson, Z.-u. Rahman, and G. A. Woodell, "A multiscale retinex for bridging the gap between color images and the human observation of scenes," *IEEE Transactions on Image processing*, vol. 6, no. 7, pp. 965–976, 1997.

[82] C. Neumeyer, "On spectral sensitivity in the goldfish: evidence for neural interactions between different "cone mechanisms"," *Vision research*, vol. 24, no. 10, pp. 1223–1231, 1984.

[83] R. H. Douglas and C. W. Hawryshyn, "Behavioural studies of fish vision: an analysis of visual capabilities," in *The visual system of fish*. Springer, 1990, pp. 373–418.

[84] R. Douglas and M. Djamgoz, *The visual system of fish*. Springer Science & Business Media, 2012.

[85] S.-B. Gao, M. Zhang, Q. Zhao, X.-S. Zhang, and Y.-J. Li, "Underwater image enhancement using adaptive retinal mechanisms," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5580–5595, 2019.

[86] S. Zhang, T. Wang, J. Dong, and H. Yu, "Underwater image enhancement via extended multi-scale retinex," *Neurocomputing*, vol. 245, pp. 1–9, 2017.

[87] P. Drews, E. Nascimento, F. Moraes, S. Botelho, and M. Campos, "Transmission estimation in underwater single images," in *Proceedings of the IEEE international conference on computer vision workshops*, 2013, pp. 825–830.

[88] Y.-T. Peng, K. Cao, and P. C. Cosman, "Generalization of the dark channel prior for single image restoration," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2856–2868, 2018.

[89] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 12, pp. 2341–2353, 2010.

[90] A. Galdran, D. Pardo, A. Picón, and A. Alvarez-Gila, "Automatic red-channel underwater image restoration," *Journal of Visual Communication and Image Representation*, vol. 26, pp. 132–145, 2015.

[91] D. Berman, T. Treibitz, and S. Avidan, "Diving into haze-lines: Color restoration of underwater images," in *Proc. British Machine Vision Conference (BMVC)*, vol. 1, no. 2, 2017.

[92] H. Liu and L.-P. Chau, "Underwater image restoration based on contrast enhancement," in *2016 IEEE International Conference on Digital Signal Processing (DSP)*. IEEE, 2016, pp. 584–588.

[93] Y. Wang, H. Liu, and L.-P. Chau, "Single underwater image restoration using adaptive attenuation-curve prior," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 3, pp. 992–1002, 2017.

[94] C. Li, J. Guo, C. Guo, R. Cong, and J. Gong, "A hybrid method for underwater image correction," *Pattern Recognition Letters*, vol. 94, pp. 62–67, 2017.

[95] Y.-T. Peng and P. C. Cosman, "Underwater image restoration based on image blurriness and light absorption," *IEEE transactions on image processing*, vol. 26, no. 4, pp. 1579–1594, 2017.

[96] C. Li, J. Guo, S. Chen, Y. Tang, Y. Pang, and J. Wang, "Underwater image restoration based on minimum information loss principle and optical properties of underwater imaging," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1993–1997.

[97] M. J. Islam, Y. Xia, and J. Sattar, "Fast underwater image enhancement for improved visual perception," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3227–3234, 2020.

[98] J. Li, K. A. Skinner, R. M. Eustice, and M. Johnson-Roberson, "Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images," *IEEE Robotics and Automation letters*, vol. 3, no. 1, pp. 387–394, 2017.

[99] C. Li, C. Guo, W. Ren, R. Cong, J. Hou, S. Kwong, and D. Tao, "An underwater image enhancement benchmark dataset and beyond," *IEEE Transactions on Image Processing*, vol. 29, pp. 4376–4389, 2019.

[100] X. Ye, H. Xu, X. Ji, and R. Xu, "Underwater image enhancement using stacked generative adversarial networks," in *Pacific Rim Conference on Multimedia*. Springer, 2018, pp. 514–524.

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

[102] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[103] C. Li, J. Guo, and C. Guo, "Emerging from water: Underwater image color correction based on weakly supervised color transfer," *IEEE Signal processing letters*, vol. 25, no. 3, pp. 323–327, 2018.

[104] J. S. Jaffe, "Computer modeling and the design of optimal underwater imaging systems," *IEEE Journal of Oceanic Engineering*, vol. 15, no. 2, pp. 101–111, 1990.

[105] T. Łuczynski, Y. Petillot, and S. Wang, "Seeing through water: From underwater image synthesis to generative adversarial networks based underwater single image enhancement."

[106] T. Ueda, K. Yamada, and Y. Tanaka, "Underwater image synthesis from rgb-d images and its application to deep underwater image restoration," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 2115–2119.

[107] N. G. Jerlov, *Marine optics*. Elsevier, 1976.

[108] H. Gupta and K. Mitra, "Unsupervised single image underwater depth estimation," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 624–628.

[109] M. Hou, R. Liu, X. Fan, and Z. Luo, "Joint residual learning for underwater image enhancement," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 4043–4047.

[110] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Multimodal image-to-image translation by enforcing bi-cycle consistency," in *Advances in neural information processing systems*, 2017, pp. 465–476.

[111] H. You, Y. Cheng, T. Cheng, C. Li, and P. Zhou, "Bayesian cyclegan via marginalizing latent sampling," 2018.

[112] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[113] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711.

[114] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," *Advances in neural information processing systems*, vol. 29, pp. 658–666, 2016.

[115] J. Bruna, P. Sprechmann, and Y. LeCun, "Super-resolution with deep convolutional sufficient statistics," *arXiv preprint arXiv:1511.05666*, 2015.

[116] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[117] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *In Workshop at International Conference on Learning Representations*. Citeseer, 2014.

[118] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *arXiv preprint arXiv:1506.06579*, 2015.

[119] J. Bruna, C. Szegedy, I. Sutskever, I. Goodfellow, W. Zaremba, R. Fergus, and D. Erhan, "Intriguing properties of neural networks," 2013.

[120] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.

[121] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 1817–1824.

[122] L. Chen, L. Tong, F. Zhou, Z. Jiang, Z. Li, J. Lv, J. Dong, and H. Zhou, "A benchmark dataset for both underwater image enhancement and underwater object detection," *arXiv preprint arXiv:2006.15789*, 2020.

[123] D. Berman, D. Levy, S. Avidan, and T. Treibitz, "Underwater single image color restoration using haze-lines and a new quantitative dataset," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[124] S. Wang, K. Ma, H. Yeganeh, Z. Wang, and W. Lin, "A patch-structure representation method for quality assessment of contrast changed images," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2387–2390, 2015.

[125] K. Panetta, C. Gao, and S. Agaian, "Human-visual-system-inspired underwater image quality measures," *IEEE Journal of Oceanic Engineering*, vol. 41, no. 3, pp. 541–551, 2015.

[126] M. Yang and A. Sowmya, "An underwater color image quality evaluation metric," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 6062–6071, 2015.

[127] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic gradient descent," in *ICLR: International Conference on Learning Representations*, 2015, pp. 1–15.

[128] D. Hoiem, Y. Chodpathumwan, and Q. Dai, "Diagnosing error in object detectors," in *European conference on computer vision*. Springer, 2012, pp. 340–353.

[129] P. M. Uplavikar, Z. Wu, and Z. Wang, "All-in-one underwater image enhancement using domain-adversarial learning." in *CVPR Workshops*, 2019, pp. 1–8.

[130] C. Ancuti, C. O. Ancuti, C. De Vleeschouwer, R. Garcia, and A. C. Bovik, "Multi-scale underwater descattering," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 4202–4207.

[131] S. Emberton, L. Chittka, and A. Cavallaro, "Underwater image and video dehazing with pure haze region segmentation," *Computer Vision and Image Understanding*, vol. 168, pp. 145–156, 2018.

[132] D. Akkaynak and T. Treibitz, "A revised underwater image formation model," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6723–6732.

[133] G. Cheng, J. Han, P. Zhou, and D. Xu, "Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 265–278, 2018.

[134] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 159, pp. 296–307, 2020.

[135] B. Hanin and D. Rolnick, "How to start training: The effect of initialization and architecture," *arXiv preprint arXiv:1803.01719*, 2018.

[136] D. Mishkin and J. Matas, "All you need is a good init," *arXiv preprint arXiv:1511.06422*, 2015.

[137] I. Derényi, T. Geszti, and G. Györgyi, "Generalization in the programed teaching of a perceptron," *Physical Review E*, vol. 50, no. 4, p. 3192, 1994.

[138] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[139] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[140] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 472–480.

[141] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[142] L. Yang, "Classifiers selection for ensemble learning based on accuracy and diversity," *Procedia Engineering*, vol. 15, pp. 4266–4270, 2011.

[143] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial intelligence*, vol. 137, no. 1-2, pp. 239–263, 2002.

[144] Z.-H. Zhou and W. Tang, "Selective ensemble of decision trees," in *International workshop on rough sets, fuzzy sets, data mining, and granular-soft computing*. Springer, 2003, pp. 476–483.

[145] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine learning*, vol. 51, no. 2, pp. 181–207, 2003.

[146] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3. IEEE, 2006, pp. 850–855.

[147] Y. Gong, X. Yu, Y. Ding, X. Peng, J. Zhao, and Z. Han, "Effective fusion factor in fpn for tiny object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1160–1168.

[148] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9627–9636.

[149] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[150] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[151] X. Zhang, F. Wan, C. Liu, X. Ji, and Q. Ye, "Learning to match anchors for visual object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[152] B. Li, Y. Liu, and X. Wang, "Gradient harmonized single-stage detector," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8577–8584.

[153] Y. Song, B. He, and P. Liu, "Real-time object detection for auvs using self-cascaded convolutional neural networks," *IEEE Journal of Oceanic Engineering*, 2019.

[154] H. Zhou, F. Yang, W. Wang, T. Wang, and C. Yan, "Research and application of an underwater detection robot with three level control mode of rov/arv/auv," *DEStech Transactions on Computer Science and Engineering*, no. iciti, 2018.

[155] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," 2016.

[156] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, pp. 8778–8788, 2018.

[157] J. Kim, J. Jeong, and J. Shin, "M2m: Imbalanced classification via major-to-minor translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 896–13 905.

[158] T. Dutta, A. Singh, and S. Biswas, "Adaptive margin diversity regularizer for handling data imbalance in zero-shot sbir," in *European Conference on Computer Vision*. Springer, 2020, pp. 349–364.

[159] K. R. M. Fernando and C. P. Tsokos, "Dynamically weighted balanced loss: Class imbalanced learning and confidence calibration of deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[160] S. Beery, Y. Liu, D. Morris, J. Piavis, A. Kapoor, N. Joshi, M. Meister, and P. Perona, "Synthetic examples improve generalization for rare classes," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 863–873.

[161] Z. Wang, K. Qinami, I. C. Karakozis, K. Genova, P. Nair, K. Hata, and O. Russakovsky, "Towards fairness in visual recognition: Effective strategies for bias mitigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8919–8928.

[162] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on knowledge and data engineering*, vol. 18, no. 1, pp. 63–77, 2005.

[163] C. Drummond, R. C. Holte *et al.*, "C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in *Workshop on learning from imbalanced datasets II*, vol. 11. Citeseer, 2003, pp. 1–8.

[164] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*. Springer, 2005, pp. 878–887.

[165] L. Shen, Z. Lin, and Q. Huang, "Relay backpropagation for effective learning of deep convolutional neural networks," in *European conference on computer vision*. Springer, 2016, pp. 467–482.

[166] J. T. Zhou, L. Zhang, Z. Fang, J. Du, X. Peng, and Y. Xiao, "Attention-driven loss for anomaly detection in video surveillance," *IEEE transactions on circuits and systems for video technology*, vol. 30, no. 12, pp. 4639–4647, 2019.

[167] K. Song, H. Yang, and Z. Yin, "Multi-scale attention deep neural network for fast accurate object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 10, pp. 2972–2985, 2018.

[168] C. Liu, Z. Wang, S. Wang, T. Tang, Y. Tao, C. Yang, H. Li, X. Liu, and X. Fan, "A new dataset, poisson gan and aquanet for underwater object grabbing," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[169] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[170] L. Bottou, "Stochastic gradient descent tricks," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 421–436.

[171] F. R. Hampel, *Robust statistics: the approach based on influence functions*. Wiley-Interscience, 1986, vol. 196.

[172] J. T. Barron, "A general and adaptive robust loss function," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4331–4339.

[173] L. Chen, J. Dong, and H. Zhou, "Class balanced underwater object detection dataset generated by class-wise style augmentation," *arXiv preprint arXiv:2101.07959*, 2021.

[174] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[175] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.