Underwater object detection using Invert Multi-Class Adaboost with deep learning

Long Chen
School of Informatics
University of Leicester
Leicester, United Kingdom
lc408@leicester.ac.uk

Zhihua Liu
School of Informatics
University of Leicester
Leicester, United Kingdom
zl208@leicester.ac.uk

Lei Tong
School of Informatics
University of Leicester
Leicester, United Kingdom
lt228@leicester.ac.uk

Zheheng Jiang
School of Informatics
University of Leicester
Leicester, United Kingdom
zj53@leicester.ac.uk

Shengke Wang
Department of Computer Science
Ocean University of China
Qingdao, China
neverme@ouc.edu.cn

Junyu Dong
Department of Computer Science
Ocean University of China
Qingdao, China
dongjunyu@ouc.edu.cn

Huiyu Zhou
School of Informatics
University of Leicester
Leicester, United Kingdom
hz143@leicester.ac.uk

Abstract—In recent years, deep learning based methods have achieved promising performance in standard object detection. However, these methods lack sufficient capabilities to handle underwater object detection due to these challenges: (1) Objects in real applications are usually small and their images are blurry, and (2) images in the underwater datasets and real applications accompany heterogeneous noise. To address these two problems, we first propose a novel neural network architecture, namely Sample-WeIghted hyPEr Network (SWIPENet), for small object detection. SWIPENet consists of high resolution and semantic-rich Hyper Feature Maps which can significantly improve small object detection accuracy. In addition, we propose a novel sample-weighted loss function which can model sample weights for SWIPENet, which uses a novel sample re-weighting algorithm, namely Invert Multi-Class Adaboost (IMA), to reduce the influence of noise on the proposed SWIPENet. Experiments on two underwater robot picking contest datasets URPC2017 and URPC2018 show that the proposed SWIPENet+IMA framework achieves better performance in detection accuracy against several state-of-the-art object detection approaches.

Index Terms—underwater object detection, Multi-Class Adaboost, sample re-weighting, noisy data

I. INTRODUCTION

Underwater object detection aims to localise and recognise objects in underwater scenes. This research has attracted continuous attention because of its widespread applications in the fields such as oceanography [1], underwater navigation [2] and fish farming [3]. However, it is still a challenging task due to complicated underwater environments and lighting conditions.

Deep learning based object detection systems have demonstrated promising performance in various applications but still felt short in 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 in Fig. 1). Secondly, the images in the existing underwater datasets and real applications are cluttered. It has been known that in the underwater scenes, wavelength-dependent absorption and scattering [4] significantly degrade the quality of underwater images. This causes many problems such as visibility loss, weak contrast and color change, which pose numerous challenges to the detection task.

To address these problems, we here propose a deep neural network named Sample-WeIghted hyPEr Network (SWIPENet), which fully takes advantage of multiple Hyper Feature Maps to improve small object detection. Furthermore, we introduce a sample-weighted loss function that cooperates with an Invert Multi-Class Adaboost (IMA) algorithm to reduce the influence of noise on the feature learning of SWIPENet.

The rest of the paper is organised as follows. Section II gives a brief introduction about the related works. Section III describes the structure of SWIPENeT and the sample-weighted loss. Section IV introduces the Invert Multi-Class Adaboost algorithm. Section V reports the results of the proposed method on two underwater datasets URPC2017 and URPC2018.

II. RELATED WORK

A. Underwater object detection

Underwater object detection techniques have been employed in marine ecology studies for many years. Strachan et al. [5] used color and shape descriptors to recognise fish transported on a conveyor belt, monitored by a digital camera. Spampinato et al. [6] presented a vision system for detecting, tracking and counting fish in real-time videos, which consist of video texture analysis, object detection and tracking process. However, the above-mentioned methods heavily rely on hand-crafted features, which have limited the representation ability. Choi [7] applied a foreground detection method to extracting candidate fish windows and used Convolution Neural Networks (CNNs) to classify fish species in the field of view. Ravanbakhsh et al.
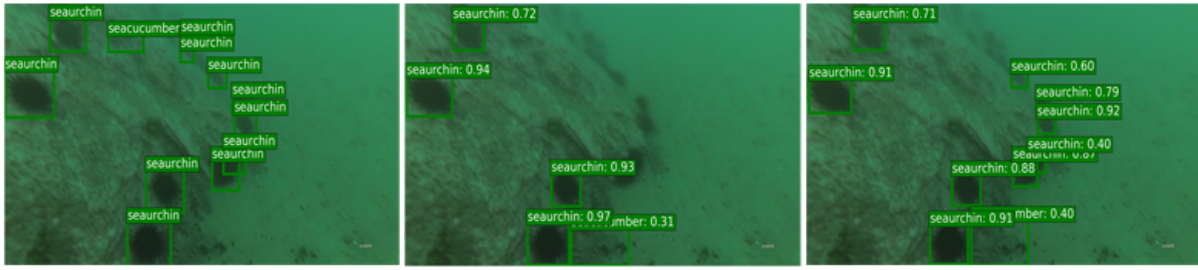
Fig. 1. Exemplar images with ground truth annotations (left), results of Single Shot MultiBox Detector (SSD) (mid) [25] and our method (right). SSD cannot detect all the small objects while our method outperforms SSD in this case.

[8] compared a deep learning method against the Histogram of Oriented Gridients (HOG)+Support Vector Machine (SVM) method in detecting coral reef fishes, and the experimental results show the superiority of the deep learning methods in underwater object detection. Li et al. [9] exploited Fast RCNN [10] to detect and recognise fish species. Li et al. [11] accelerated fish detection using Faster RCNN [12]. However, these Fast RCNN methods use features from the last convolution layer of the neural network, which is coarse and cannot be used to effectively detect small objects. In addition, the underwater object detection datasets are extremely scarce that hinders the development of underwater object detection techniques. Recently, Jian et al. [13], [14] proposed a underwater dataset for underwater saliency detection, which provides object-level annotations that can be used to evaluate underwater object detection algorithms.

B. Sample re-weighting

Sample re-weighting can be used to address noisy data problems [15]. It usually assigns a weight to each sample and then optimises the sample-weighted training loss. For training loss based approaches, we may have two research directions. For example, focal loss [16] and hard example mining [17] emphasise samples with higher training loss while self-placed learning [18] and curriculum learning [19] encourage learning samples with low loss. The two possible solutions take different assumptions over the training data. The first solution assumes that high loss samples are those to be learned whilst the second one assumes that high loss samples are prone to be disturbing or noisy data. Different from the training loss based methods, Multi-Class Adaboost [20] re-weights samples according to the classification results. This method focuses on learning misclassified samples through increasing their weights during the iteration. In Section IV, we propose a novel detection based sample re-weighting algorithm, namely Invert Multi-Class Adaboost (IMA), to reduce the influence of noise by re-weighting.

III. SAMPLE-WEIGHTED HYPER NETWORK (SWIPENET)

A. The architecture of our proposed SWIPENeT

Evidence shows that the down-sampling excises of Convolutional Neural Network result in strong semantics that lead to the success of many classification tasks. However, this is not
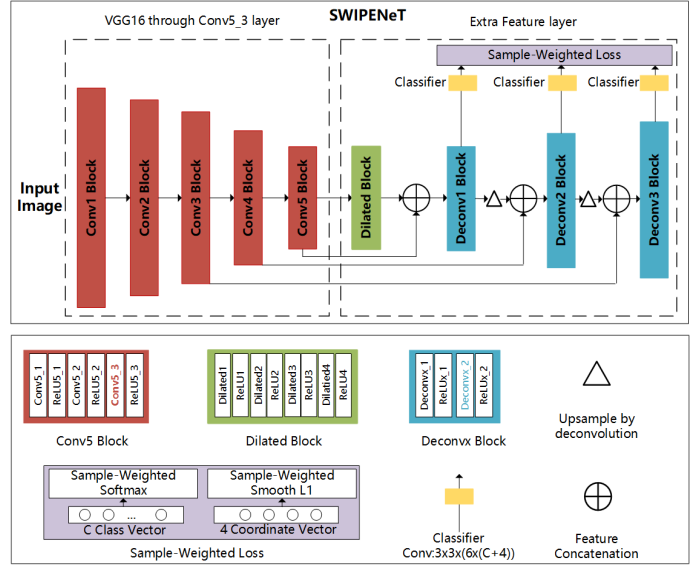

Fig. 2. The overview of our proposed SWIPENeT.

enough for the object detection task which not only needs to recognise the category of the object 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 paper, we propose the SWIPENet architecture that includes several high resolution and semantic-rich Hyper Feature Maps inspired by Deconvolutional Single Shot Detector (DSSD) [21]. DSSD augments a fast down-sampling detection framework SSD [25] with multiple up-sampling deconvolution layers to increase the resolutions of feature maps. In the DSSD architecture, firstly, multiple down-sampling convolution layers are constructed to extract high semantic feature maps that benefit object classification. 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 feature maps. However, the detailed information lost by the down-sampling operations cannot be fully recovered even though the resolutions have been recovered. To improve DSSD, we use dilated convolution layers [22], [23] to obtain strong

semantics without losing detailed information that support object localization. Fig. 2 illustrates the overview of our proposed SWIPENet, which consists of multiple convolution blocks (red), dilated convolution blocks (green), deconvolution blocks (blue) and a novel sample-weigthed loss (gray). The front layers of the SWIPENet are based on the architecture of the standard VGG16 model [24] (truncated at Conv5_3 layer). Different from DSSD, we add four dilated convolution layers with ReLU activations to the network, which can obtain large receptive fields without sacrificing the resolutions of the feature maps (large receptive fields lead to strong semantics). We further up-sample feature maps using deconvolution and then use skip connection to pass fine details of low layers to high layers. Finally, we 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. 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 $C + 1$ class scores (C indicates the number of the object classes and 1 indicates the background class) and 4 coordinate offsets relative to the original default box shape.

B. Sample-weighted loss

We propose a novel sample-weighted loss function which can model sample weights in SWIPENeT. The sample-weighted loss enables SWIPENet to focus on learning high weight samples whilst ignoring low weight samples. It co-operates with a novel sample re-weighting algorithm, namely Invert Multi-Class Invert Adaboost, to reduce the influence of possible noise on the SWIPENet by decreasing their weights. Technically speaking, our sample-weighted 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 (the derivation of the original Softmax loss and Smooth L1 loss can be found in [25]):

$$L = \frac{1}{Num}(\alpha_1 L_{cls}(pre\_cls, gt\_cls) + \alpha_2 L_{reg}(pre\_loc, gt\_loc)) \quad (1)$$

where

$$L_{cls} = -\sum_{i \in Pos}\sum_{c=1}^{C+1} f(\bar{w}_i^m)gt\_cls_i^c log(pre\_cls_i^c)$$
$$-\sum_{i \in Neg}\sum_{c=1}^{C+1} gt\_cls_i^c log(pre\_cls_i^c) \quad (2)$$

$$L_{reg} = \sum_{i \in Pos}\sum_{l \in Loc} f(\bar{w}_i^m)SmoothL1(pre\_loc_i^l - gt\_loc_j^l) \quad (3)$$

Following [25], SWIPENet trains an object detector using default boxes on several layers. If the Intersection over Union (IoU) between a default box and its most overlapped object is larger than a pre-defined threshold, then the default box is a match to this ground truth object and added to the positive

sample set Pos. If a default box doesn't match any ground truth object, it will be regarded as negative sample and added to the negative sample set Neg. Num is the number of the positive default boxes, α_1 and α_2 denote the weight terms of classification loss and regression loss respectively. $C + 1$ denotes C object classes plus one background class. $pre\_cls_i^c$ and $gt\_cls_i^c$ denote the c-th element of the predicted class score and the ground truth class for the i-th default box. $pre\_loc_i^l$ and $gt\_loc_i^l$ denote the l-th element of the predicted coordinate and the ground truth coordinate for the i-th positive default box. $Loc = (cx, cy, w, h)$ denotes the object coordinate information that includes the coordinates of center (cx, cy) with width w and height h. In our unified detection framework, we train SWIPENet whilst re-weighting each positive sample using IMA. Denote $\bar{w}_i^m$ as the weight of the i-th positive sample learned in IMA in the m-th iteration. $f(.)$ is a mapping function that maps weight $\bar{w}_i^m$ to $f(\bar{w}_i^m)$ which indicates the weight of the i-th positive sample used in the sample-weighted loss. We describe the mapping function $f(.)$ in Section IV in more details. The sample-weighted loss enables SWIPENet to focus on learning high weight samples and ignore low weight samples.

Sample weights influence the feature learning of SWIPENet through adapting the gradient of the parameters used in back-propagation. Let w_{cnn} be the parameters of SWIPENet, and the gradient of w_{cnn} can be denoted as $\frac{\partial L}{\partial w_{cnn}}$:

$$\frac{\partial L}{\partial w_{cnn}} = \frac{\alpha_1}{Num}\sum_{i \in Pos} f(\bar{w}_i^m) \bigtriangledown_{w_{cnn}}^{L_{cls\_pos}^i} + \frac{\alpha_1}{Num}\sum_{i \in Neg} \bigtriangledown_{w_{cnn}}^{L_{cls\_neg}^i}$$
$$+ \frac{\alpha_2}{Num}\sum_{i \in Pos} f(\bar{w}_i^m) \bigtriangledown_{w_{cnn}}^{L_{loc\_pos}^i}$$
$$(4)$$

Here, $\bigtriangledown_{w_{cnn}}^{L_{cls\_pos}^i}$ and $\bigtriangledown_{w_{cnn}}^{L_{loc\_pos}^i}$ indicate the influence of the classification and localisation losses of the i-th positive sample on the gradient of the parameters. $\bigtriangledown_{w_{cnn}}^{L_{cls\_neg}^i}$ indicates the influence of the i-th negative sample's localisation loss on the gradient of the parameters. The derivation procedure is shown in the Supplementary. It can be seen from (4) that the gradient of the parameters is influenced by the i-th positive sample's weight $f(\bar{w}_i^m)$. Specially, $f(\bar{w}_i^m)$ influences the first term and the third term on the right hand side of (4). The smaller the weight is, the smaller gradient in back-propagation is used for the i-th sample. For example, if we assign a weight of 1000 and 1 to the same positive sample respectively, then the magnitude of the gradient from the former will be much bigger than that of the gradient from the later. The feature learning of SWIPENeT is dominated by high-weight samples while the feature learning of low-weight samples is ignored.

IV. INVERT MULTI-CLASS ADABOOST (IMA)

A. The overview of IMA

SWIPENet possibly misses or incorrectly detects some objects in the training set, which may be treated as noisy data [26]–[29]. This is because the noisy data are extremely blurry and similar to the background, making them easy to be ignored
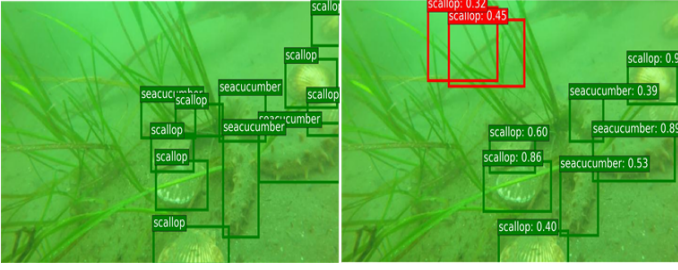
Fig. 3. SWIPENeT treats backgrounds as objects on URPC2018. Left is the ground truth annotations and right includes detection result by SWIPENet.

or detected as the background. If we train the SWIPENet using these noisy data, the performance may be affected. SWIPENet cannot distinguish the background from the objects mainly due to the noise. Fig. 3 shows exemplar testing images and their incorrect detections by SWIPENet. To handle this problem, we here propose the IMA algorithm inspired by [30] to reduce the weights of the uncertain objects in order to improve the detection accuracy of SWIPENet.

IMA is based on Multi-Class Adaboost [20] which firstly trains multiple base classifiers sequentially and assign a weight value α_m according to its error rate E_m. Then, 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 weights. Our IMA also trains M times SWIPENet and then ensemble them into a unified model. Differently, in each training iteration, IMA decreases the weight of the missed objects to reduce the influence of these 'disturbing' samples. The overview of the proposed IMA algorithm can be found in Algorithm 1. I_{train} indicates the training images with the ground truth objects $B = \{b_1, b_2, ..., b_N\}$, 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 sample's weight is initialised to $\frac{1}{N}$ in the first iteration, i.e. $w_j^1 = \frac{1}{N}, j = 1, ..., N$.

In the m-th iteration, we firstly compute the weights of the positive samples. If the i-th positive sample matches the j-th object during the training, we assign the j-th object's weight w_j^m as the i-th positive sample's weight $\bar{w}_i^m$, i.e. $\bar{w}_i^m = w_j^m$. The mapping function $f(.)$ maps the weight $\bar{w}_i^m$ to the weight $f(\bar{w}_i^m)$ used in the sample-weighted loss by (10). Secondly, we use $f(\bar{w}_i^m)$ to train the m-th SWIPENet G_m. Thirdly, we run the m-th SWIPENet on the training set and get the detection set $D_m = \{d_1, d_2, ..., d_i\}$ while $d_i = (cls, score, cx, xy, w, h)$ is the result of the i-th detected box, including class (cls), score ($score$) and coordinates (cx, cy, w, h). We compute the m-th SWIPENet's error rate E_m based on the percentage of the undetected objects.

$$E_m = \sum_{j=1}^{N} w_j^m I(b_j) / \sum_{j=1}^{N} w_j^m \quad (5)$$

where

$$I(b_j) = \begin{cases} 0, & \exists\, d \in D_m, s.t. b_j.cls = d.cls \wedge IoU(b_j, d) \geq \theta \\ 1, & otherwise \end{cases}$$
$$(6)$$

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 θ (0.5 here), we set $I(b_j) = 0$ which indicates the j-th object has been detected, and $I(b_j) = 1$ indicates the undetected. Fourthly, we compute the m-th SWIPENeT's weight α_m in the final ensemble model. C is the number of the object classes.

$$\alpha_m = log\frac{1 - E_m}{E_m} + log(C - 1) \quad (7)$$

Finally, we update each object's weight w_j^m. Different from Multi-Class Adaboost, we decrease the weights of the undetected objects by (8). z_m is a normalization constant. The iteration repeats again till all M SWIPENeT have been trained.

$$w_j^m \leftarrow \frac{w_j^m}{z_m} exp(\alpha_m(1 - I(b_j))) \quad (8)$$

In the testing stage, we first run all M SWIPENeT on the testing set I_{test} and get M detection set $D_m = G_m(I_{test}), m = 1, 2, ..., M$. Afterwards, we re-score each detection d in D_m according to α_m, i.e. $d.score = \alpha_m d.score$.

Finally, we combine all the detections and apply Non-Maximum Suppression [31] to removing the overlapped detections and generating the final detections by (9).

$$D = NonMaximumSuppression(\bigcup_{m=1}^{M} D_m) \quad (9)$$

Algorithm 1 SWIPENeT with Invert Multi-Class Adaboost

Input: Training images I_{train} with ground truth objects $B = \{b_1, ..., b_N\}$, testing images I_{test}.
Output: Detection results D.
1: Initialize the object weights $w_j^1 = \frac{1}{N}, j = 1, ..., N$.
2: **for** $m = 1$ to M **do**
3:    • Compute the weights of positive samples using (10).
4:    • Train the m-th SWIPENeT G_m using (1)-(3).
5:    • Compute the m-th SWIPENeT's error rate E_m using (5)-(6).
6:    • Compute the m-th SWIPENeT's weight α_m in the ensemble model using (7).
7:    • Decrease the weights of undetected objects and increase the weights of detected objects using (8).
8: **end for**
9: Get the final detections D using (9).
10: **return** Detections results D

B. The mapping function f(.)

We define two weights for the i-th positive sample, i.e. IMA weight $\bar{w}_i^m$ which is learned in IMA and $f(\bar{w}_i^m)$ which indicates the weight used in the sample-weighted loss. In IMA, the initial IMA weight of each positive sample is $\frac{1}{N}$ (N is

the number of the objects in the training data) and the initial weight of each positive sample used in the sample-weighted loss is 1. Hence, in the sample-weighted loss function, the positive samples' weights are N times their IMA weights. Intuitively, we can define a linear mapping function $f(.)$ to map the IMA weight to the weight used in the sample-weighted loss. Here, we first assign the weight of the j-th object as the weight of the i-th default box if they match, we denote the weights of the j-th object and i-th default box as w_j^m and $\bar{w}_i^m$, respectively. We have $\bar{w}_i^m = w_j^m$. Then, we map IMA weight $\bar{w}_i^m$ to $f(\bar{w}_i^m)$ where the sample-weighted loss can be used by a linear mapping function stated in (10).

$$f(\bar{w}_i^m) = N * \bar{w}_i^m, 0 < \bar{w}_i^m < 1 \tag{10}$$

V. EXPERIMENTS ON URPC2017 AND URPC2018

We evaluate our approach on two underwater datasets URPC2017 and URPC2018 from the Underwater Robot Picking Contest. The URPC2017 dataset has 3 object categories, including seacucumber, seaurchin and scallop. There are 18,982 training images and 983 testing images. The URPC2018 dataset has 4 object categories, including seacucumber, seaurchin, scallop and starfish. There are 2,897 images in the training set, but the testing set is not publicly available. We randomly split the training set of URPC2018 into a training set of 1,999 images and a testing set of 898 images. Both two datasets provide underwater images and box level annotations (detailed descriptions are provided in the Supplementary). In this section, we firstly analyse our method using ablation studies in Subsection V-A. Then, we compare our method against the other state of the art detection frameworks, including SSD [25], YOLOv3 [32] and Faster RCNN [12] shown in Subsection V-B.

A. Ablation studies

We study the role of several components in our SWIPENet in this section, including dilated convolution layers, skip connection and IMA.

Implementation details. To investigate the influence of dilated convolution and skip connection on our SWIPENet, we design two networks for the comparisons. All the networks in the ablation studies are trained with the Adam optimisation algorithm on a single NVIDIA Tesla P100 GPU with a 16 GB memory. We use an image scale of 512x512 as the input for both training and testing. The source code is developed upon Keras and will be published[1]. For URPC2017, the batch-size is 16, the learning rate is 0.0001, our models often diverge when we use a high learning rate due to unstable gradients, and all the networks achieve the best performance after running 120 epochs. For URPC2018, the batch-size is 16, our models converges quickly when we use a high learning rate 0.001. All the networks achieve the best performance after running 80 epochs.

Ablation studies on skip connection and dilated convolution layer. To investigate the influence of skip connection,

[1] https://github.com/LongChenCV/SWIPENet

TABLE I
ABLATION STUDIES ON URPC2017 AND URPC2018. SKIP INDICATES SKIP CONNECTION, AND DILATION INDICATES DILATED CONVOLUTION LAYER. 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

TABLE II
THE PERFORMANCE OF SWIPENeT (MAP(%)) IN EACH ITERATION OF IMA.

Dataset	IMA iteration	Single	Ensemble
URPC2017	1	42.1	-
	2	44.2	45.0
	3	45.3	46.3
	4	40.5	45.3
	5	37.2	44.2
URPC2018	1	62.2	-
	2	63.3	64.5
	3	62.4	64.0
	4	61.2	62.8
	5	59.3	62.1

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 four dilated convolution layers in UWNet1 with convolution layers to learn the influence of dilated convolution. Table I shows the performance comparison of different networks on URPC2017 and URPC2018. SWIPENet performs 1.7% and 1.0% better than UWNet1 on the two datasets respectively. 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 2.1% and 3.1% improvement because the dilated convolution in UWNet1 brings much semantic information to the high layers which enhances the classification ability.

Ablation studies on IMA. Table II shows the performance of the single model and the ensemble model after each iteration. The ensemble model has better performance on the two datasets. By reducing the influence of noise, IMA gives SWIPENet 4.2% and 2.3% improvement on the two datasets respectively. The single model and the ensemble model both perform best in the 3rd iteration on URPC2017 and in the 2nd iteration on URPC2018. However, the performance of the two models goes down as most of the detected objects are continuously up-weighted with the increasing of the iteration number where SWIPENet over-fits with the high-weight objects. According to the experimental results in Table II, we set the number of iterations as 3 on URPC2017 and 2 on URPC2018.

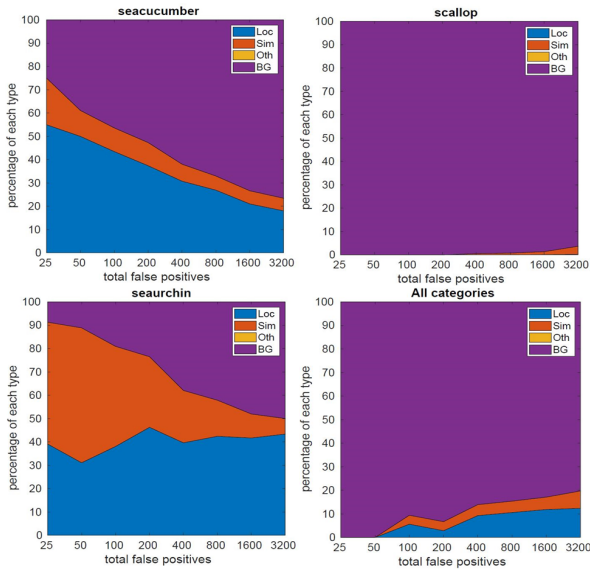We use the detection analysis tool of [33] to analyse the

Fig. 4. The distribution of top-ranked false positive types for each category and all categories on URPC2017. The false positive types include localisation error (Loc), confusion with similar categories (Sim), with others (Oth), or with background (BG).

Fig. 5. Precision/Recall curves of different methods on URPC2017.

TABLE III
COMPARISON WITH THE STATE-OF-ARTS ON URPC2017.

Methods	Backbone	seacucumber	seaurchin	scallop	mAP
SSD300	VGG16	28.1	51.3	21.2	33.5
SSD512	VGG16	38.4	52.9	15.7	35.7
YOLOv3	DarkNet53	28.4	50.3	22.4	33.7
Faster RCNN	VGG16	27.2	45.0	31.9	34.7
Faster RCNN	ResNet50	31.0	41.4	33.5	35.3
Faster RCNN	ResNet101	26.2	47.7	32.5	35.5
OurFirstSingle	SWIPENeT	43.6	51.3	31.2	42.1
OurBestSingle	SWIPENeT	**45.0**	49.7	41.3	45.3
OurEnsemble	SWIPENeT	44.4	**52.4**	**42.1**	**46.3**

false positives of SWIPENeT without IMA. Fig. 4 shows the distribution of the top-ranked false positive types for each category of the URPC2017 testing set. SWIPENeT cannot well distinguish objects with complex background and localise objects accurately due to the noise in the data.

B. Comparison with state-of-the-art detection frameworks

In this section, we compare our method with other state-of-the-art detection frameworks, including SSD [16], YOLOv3 [19] and Faster RCNN [22].

Implementation details. For SSD, we use VGG16 [19] as the backbone network and conduct experiments on two SSD

TABLE IV
COMPARISON WITH THE STATE-OF-ARTS ON URPC2018.

Methods	Backbone	seacucumber	seaurchin	scallop	starfish	mAP
SSD300	VGG16	38.5	83.0	30.8	75.1	56.9
SSD512	VGG16	44.2	**84.4**	35.8	78.1	60.6
YOLOv3	DarkNet53	35.7	83.0	34.0	77.9	57.7
Faster RCNN	VGG16	43.3	83.0	32.0	74.5	58.2
Faster RCNN	ResNet50	41.1	83.2	34.5	77.2	59.0
Faster RCNN	ResNet101	44.3	82.5	34.7	77.5	59.8
OurFirstSingle	SWIPENeT	46.4	84.0	40.2	78.2	62.2
OurBestSingle	SWIPENeT	50.3	83.7	39.8	**79.4**	63.3
OurEnsemble	SWIPENeT	**52.8**	84.1	**42.9**	78.0	**64.5**
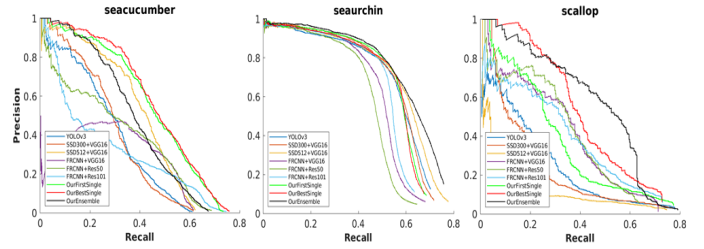
with different input sizes, i.e. SSD300 and SSD512. For Faster RCNN, we use three backbone networks including VGG16, ResNet50 [34] and ResNet101 [34]. For YOLOv3, we use its original DarkNet53 network. When we train SWIPENeT, the parameter setting is the same as that used in the ablation studies.

Tables III and IV show experimental results on URPC2017 and URPC2018. On URPC2017, SSD512 achieves 35.7 mAP, which improves 2.2% over SSD300. The gain comes from the increase of the input size. Faster RCNN with ResNet101 performs better than Faster RCNN with ResNet50 and VGG16, where the ResNet-101 plays a critical role. In addition, SSD512 achieves better performance than Faster RCNN, even though Faster RCNN uses ResNet101 as the backbone network, which has better performance than VGG16. It is because SSD512 detects multi-scale objects on different layers, and performs better than Faster RCNN on small object detection. OurFirstSingle, the SWIPENeT trained in the first iteration of IMA, outperforms all the state-of-the-arts by a large margin (above 6.4%) on URPC2017, demonstrating the superiority of our SWIPENeT in detecting small objects. OurBestSingle, the best performing single SWIPENeT, improves 3.2% over OurFirstSingle. We ensemble all the SWIPENeTs into OurEnsemble, and this further improves the results to 46.3 mAP. The gain comes from the ensemble model. In addition, both OurBestSingle and OurEnsemble surpass the best result in the URPC2017 competition, and the official leaderboard of the URPC2017 competition is shown in the Supplementary. Fig. 5 shows the Precision/Recall curves of different methods on URPC2017. OurEnsemble (black curve) performs best on the seaurchin and scallop categories, and OurBestSingle (red curve) performs better than OurEnsemble on the seacucumber category, which indicates model ensembling may not bring performance improvement for a single object category.

OurFirstSingle achieves 62.2 mAP on URPC2018 and outperforms the other state-of-the-arts. OurBestSingle improves 1.1% over OurFirstSingle, and OurEnsemble achieves the best performance, 64.5 mAP. OurEnsemle outperforms all the other state-of-the-art methods by a large margin (above 3.9%), demonstrating its superiority in detecting small objects and handling noisy data. Fig. 6 shows the Precision/Recall curves of different methods on URPC2018. OurEnsemble performs the best on the seacucumber and scallop categories. All the methods achieves higher accuracy on the seaurchin and starfish
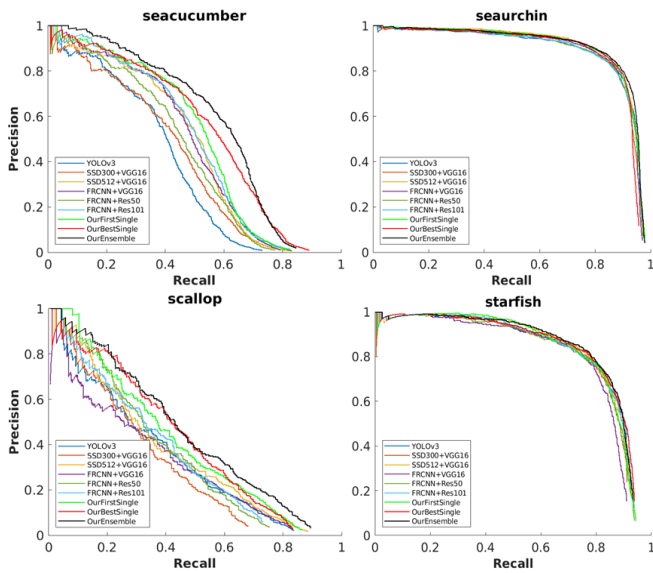
Fig. 6. Precision/Recall curves of different methods on URPC2018.

categories than on the seacucumber and scallop categories.

VI. CONCLUSION

In this paper, we proposed a neural network architecture, called Sample-WeIghted hyPEr Network (SWIPENet), for small underwater object detection. Moreover, a sample re-weighting algorithm named Invert Multi-Class Adaboost (IMA) had been presented to solve the noise issue. Our proposed method achieved state-of-the-art performance on challenging datasets, but its time complexity is M times higher than a single model since it is an ensemble of M deep neural networks. Hence, in future work, reducing the computational complexity of our proposed method is of vital importance. In addition, current deep models introduce attention mechanisms and novel loss to solve the issues of noise and small objects detection, which provide us insightful ideas to develop our SWIPENet.

ACKNOWLEDGMENT

REFERENCES

[1] Michael R Heithaus and Lawrence M Dill. Food availability and tiger shark predation risk influence bottlenose dolphin habitat use. Ecology, 83(2):480–491, 2002.

[2] Florian Shkurti,Wei-Di Chang, Peter Henderson, Md Jahidul Islam, Juan Camilo Gamboa Higuera, Jimmy Li, Travis Manderson, Anqi Xu, Gregory Dudek, and Junaed Sattar. Underwater multi-robot convoying using visual tracking by detection. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4189–4196. IEEE, 2017.

[3] Andrew Rova, Greg Mori, and Lawrence M Dill. One fish, two fish, butterfish, trumpeter: Recognizing fish in underwater video. In MVA, pages 404–407, 2007.

[4] Derya Akkaynak and Tali Treibitz. A revised underwater image formation model. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6723–6732, 2018.

[5] NJC Strachan. Recognition of fish species by colour and shape. Image and vision computing, 11(1):2–10, 1993.

[6] Concetto Spampinato, Yun-Heh Chen-Burger, Gayathri Nadarajan, and Robert B Fisher. Detecting, tracking and counting fish in low quality unconstrained underwater videos. VISAPP (2), 2008(514-519):1, 2008.

[7] Sungbin Choi. Fish identification in underwater video with deep convolutional neural network: Snumedinfo at lifeclef fish task 2015. In CLEF (Working Notes), 2015.

[8] Sébastien Villon, Marc Chaumont, Gérard Subsol, Sébastien Villéger, Thomas Claverie, and David 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, pages 160–171. Springer, 2016.

[9] Xiu Li, Min Shang, Hongwei Qin, and Liansheng Chen. Fast accurate fish detection and recognition of underwater images with fast r-cnn. In OCEANS 2015-MTS/IEEE Washington, pages 1–5. IEEE, 2015.

[10] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.

[11] Xiu Li, Min Shang, Jing Hao, and Zhixiong Yang. Accelerating fish detection and recognition by sharing cnns with objectness learning. In OCEANS 2016-Shanghai, pages 1–5. IEEE, 2016.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.

[13] Muwei Jian, Qiang Qi, et al. "The OUC-Vision Large-Scale Underwater Image Database". Proceedings 2017 IEEE International Conference on Multimedia and Expo (ICME 2017), Hong Kong, July, 2017.

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

[15] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. arXiv preprint arXiv:1803.09050, 2018.

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017.

[17] Tomasz Malisiewicz, Abhinav Gupta, Alexei A Efros, et al. Ensemble of exemplarsvms for object detection and beyond. In Iccv, volume 1, page 6. Citeseer, 2011.

[18] M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In Advances in Neural Information Processing Systems, pages 1189–1197, 2010.

[19] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In Proceedings of the 26th annual international conference on machine learning, pages 41–48. ACM, 2009.

[20] Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. Multi-class adaboost. Statistics and its Interface, 2(3):349–360, 2009.

[21] Fu, C. Y., Liu, W., Ranga, A., Tyagi, A., Berg, A. C. (2017). Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659.

[22] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: ICLR (2015)

[23] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 472–480, 2017.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, in Proc. Int. Conf. Learn. Representat., San Diego, CA, USA, 2015, pp. 1–14.

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016.

[26] A. Drory, S. Avidan, and R. Giryes, On the Resistance of Neural Nets to Label Noise, CoRR, pp. 1–19, 2018.

[27] Jonathan Krause, Benjamin Sapp, Andrew Howard, Howard Zhou, Alexander Toshev, Tom Duerig, James Philbin, and Li Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. In European Conference on Computer Vision, pages 301–320. Springer, 2016.

[28] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, Training deep neural networks on noisy labels with bootstrapping, in Proc. Int. Conf. Learn. Represent. Workshop, 2015, pp. 1–11.

[29] David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. arXiv preprint arXiv:1705.10694, 2017, unpublished.

[30] Tong, L., Zhang, Q., Sadka, A., Li, L., Zhou, H. (2019). Inverse boosting pruning trees for depression detection on Twitter. arXiv preprint arXiv:1906.00398.

[31] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In 18th International Conference on Pattern Recognition (ICPR'06), volume 3, pages 850–855. IEEE, 2006.

[32] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[33] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In European conference on computer vision, pages 340–353. Springer, 2012.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

VII. Supplementary

A. The derivation of the sample-weighted loss

Our sample-weighted loss L consists of a sample-weighted Softmax loss for the bounding box classification (L_{cls}) and a sample-weighted Smooth L1 loss for the bounding box regression (L_{reg}):

$$L = \frac{1}{Num}(\alpha_1 L_{cls}(pre\_cls, gt\_cls) + \alpha_2 L_{reg}(pre\_loc, gt\_loc)) \quad (11)$$

where

$$L_{cls} = -\sum_{i \in Pos}\sum_{c=1}^{C+1} f(\bar{w}_i^m) gt\_cls_i^c log(pre\_cls_i^c) \\ - \sum_{i \in Neg}\sum_{c=1}^{C+1} gt\_cls_i^c log(pre\_cls_i^c) \quad (12)$$

$$L_{reg} = \sum_{i \in Pos}\sum_{l \in Loc} f(\bar{w}_i^m) SmoothL1(pre\_loc_i^l - gt\_loc_j^l) \quad (13)$$

Then we obtain the partial derivative of the network parameters w.r.t. sample-weighted loss.

$$\frac{\partial L}{\partial w_{cnn}} = \frac{\alpha_1}{Num}\frac{\partial L_{cls}(pre\_cls, gt\_cls)}{\partial w_{cnn}} + \frac{\alpha_2}{Num}\frac{\partial L_{reg}(pre\_loc, gt\_loc)}{\partial w_{cnn}} \\ = \frac{\alpha_1}{Num}\sum_{i \in Pos} f(\bar{w}_i^m)\sum_{c=1}^{C+1}\frac{\partial gt\_cls_i^c log\frac{1}{pre\_cls_i^c}}{\partial w_{cnn}} \\ + \frac{\alpha_1}{Num}\sum_{i \in Neg}\sum_{c=1}^{C+1}\frac{\partial gt\_cls_i^c log\frac{1}{pre\_cls_i^c}}{\partial w_{cnn}} \\ + \frac{\alpha_2}{Num}\sum_{i \in Pos} f(\bar{w}_i^m)\frac{\partial \sum_{l \in Loc} SmoothL1(pre\_loc_i^l - gt\_loc_i^l)}{\partial w_{cnn}} \quad (14)$$

Let

$$\nabla_{w_{cnn}}^{L_{cls\_pos}^i} = \sum_{c=1}^{C+1}\frac{\partial gt\_cls_i^c log\frac{1}{pre\_cls_i^c}}{\partial w_{cnn}} \quad (15)$$

$$\nabla_{w_{cnn}}^{L_{cls\_neg}^i} = \sum_{c=1}^{C+1}\frac{\partial gt\_cls_i^c log\frac{1}{pre\_cls_i^c}}{\partial w_{cnn}} \quad (16)$$

$$\nabla_{w_{cnn}}^{L_{reg\_pos}^i} = \frac{\partial \sum_{l \in Loc} SmoothL1(pre\_loc_i^l - gt\_loc_i^l)}{\partial w_{cnn}} \quad (17)$$

Here, $\nabla_{w_{cnn}}^{L_{cls\_pos}^i}$ and $\nabla_{w_{cnn}}^{L_{reg\_pos}^i}$ denote the partial derivatives of the i-th positive sample on classification loss and regression loss respectively. $\nabla_{w_{cnn}}^{L_{cls\_neg}^i}$ is the partial derivative of the i-th negative sample on the classification loss. Then, we have

$$\frac{\partial L}{\partial w_{cnn}} = \frac{\alpha_1}{Num}\sum_{i \in Pos} f(\bar{w}_i^m) \nabla_{w_{cnn}}^{L_{cls\_pos}^i} + \frac{\alpha_1}{Num}\sum_{i \in Neg} \nabla_{w_{cnn}}^{L_{cls\_neg}^i} \\ + \frac{\alpha_2}{Num}\sum_{i \in Pos} f(\bar{w}_i^m)\nabla_{w_{cnn}}^{L_{reg\_pos}^i} \quad (18)$$

B. Description of 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 and URPC2018 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

C. The official leaderboard of the URPC2017 competition

Table V shows the official leaderboard of the URPC2017 competition, which is an anonymous leaderboard with mean Average Precision (mAP).

TABLE V
THE OFFICIAL LEADERBOARD OF URPC2017 COMPETITION.

Method	1	2	3	4	5	6	7	8	9
mAP(%)	**45.1**	35.7	33.4	32.0	30.2	29.6	28.8	28.4	26.6