IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 19, 2020, accepted May 12, 2020, date of publication June 2, 2020, date of current version June 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2999341

Active Object Detection in Sonar Images

LONGYU JIANG[1,2,3], **(Member, IEEE), TAO CAI**[1], **QIXIANG MA**[1], **FANJIN XU**[1], **AND SHIJIE WANG**[1]

[1]Laboratory of Image Science and Technology, Southeast University, Nanjing 210096, China
[2]Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing 210096, China
[3]Acoustic Science and Technology Laboratory, Harbin Engineering University, Harbin 150001, China

Corresponding author: Longyu Jiang (JLY@seu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61871124 and Grant 61876037, in part by the National Defense Pre-Research Foundation of China, in part by the fund of Science and Technology on Sonar Laboratory under Grant 6142109KF201806, in part by the Stable Supporting Fund of Acoustic Science and Technology Laboratory under Grant JCKYS2019604SSJSSO12, and in part by the Fundamental Research Funds for the Central Universities under Grant 2242019K40090 and Grant 2242020K40042.

ABSTRACT Object detection in sonar images has always been a challenge due to the low resolution of sonar images and the strong noise existing in them. Although convolutional neural networks (CNNs) have been applied to detect objects in sonar images, successful detection is impeded by the lack of large annotated sonar images. Manual annotation is not only tedious and time consuming but also demands specialty-oriented knowledge and skills, which are not easily accessible. To dramatically reduce the heavy annotation cost, this paper proposes three simple but effective active-learning-based algorithms for object detection, which can reduce the annotation cost by seeking the most informative images from unlabeled data and then continuously retraining a model by merging newly annotated samples in each iteration into an already labeled dataset to enhance the CNN's performance. The results of the experiments illustrate that the proposed active framework with approximately 35% data can achieve competitive results compared to the CNN's performance using all data.

INDEX TERMS Sonar image, active learning, object detection, deep learning, convolutional neural network.

I. INTRODUCTION

Object detection in sonar images focuses on analyzing and processing the sonar image after acoustic echo imaging. It aims to recognize possible objects and locate them. Object detection technology based on sonar images is widely used in military and civil applications, such as in detecting sea mines [1] or in identifying and tracking fish [2]. However, correct and robust detection is a challenge due to the resolution determination with changes in operation and environmental conditions and the presence of reverberation, ambient noise, and self-noise.

Because it provides accurate and general results, deep learning has been widely applied to image classification [3], object detection [4]–[7] and image segmentation [8]. At the same time, recent advances in deep learning have also achieved promising results in image classification or object detection in sonar images. For example, Valdenegro-Toro [9] used convolutional neural networks (CNNs) for feature extraction and cross-entropy loss for network classification. The author compared the performance of CNNs with those of the commonly used template matching methods for object detection and showed that CNNs can provide better performance in terms of accuracy and can be well generalized to detect unseen data. Kim *et al.* [10] designed a classifier model to select positive images having higher target existence probabilities and then applied the YOLO [11] object-detection algorithm to them. Valdenegro-Toro [12] presented a CNN approach for objectness estimation and detection proposal generation for forward-looking sonar images, and it works well, especially for detecting objects that are not present in the training set, which is a desirable property for any object detector. Kim *et al.* [13] found that sliding window recognition based on CNNs has higher recognition accuracy than general template-matching algorithms. However, acquiring annotation data in sonar image classification or detection is difficult due to the low image quality and difficulty in accessing the specialty-oriented knowledge. Therefore, how to effectively reduce the cost of data annotation is an important issue that first needs to be solved when considering using the deep learning technique.

Active learning [14], which allows the learning model to select training data, provides a way to solve the above problem. The key assumption is that if the learning algorithm

The associate editor coordinating the review of this manuscript and approving it for publication was Emrecan Demirors.

is allowed to choose the data from which it learns according to the informativeness, it will perform well with less data training. The active learning system attempts to overcome the labeling bottleneck by selecting unlabeled instances and then sending them to human annotators for annotation. Through the use of active learning, we can select only the data that contain the most informative instances from unlabeled data for labeling, which can significantly reduce the annotation cost and can overcome the problem of an unbalanced data distribution.

Active learning has been studied for image classification [15], [16] but has not been much explored for object detection. Recently, Vijayanarasimhan *et al.* [17] introduced a novel part-based detector amenable to linear classifiers and showed how to identify its most uncertain instances in sub-linear time with a hashing-based solution. However, this method is based on a traditional grid-based variant of the *jumping window* method and is not suitable for current popular detection frameworks such as Faster-RCNN [7]. Roy *et al.* [18] proposed a version-space-based active learning method for object detection in a weakly supervised setting. However, it is a kind of weakly supervised setting and still relies on the selective search method [19] to generate approximately 2000 candidate windows for each image. Rhee *et al.* [20] combined active learning and semi-supervised learning to leverage the strong points of both learning methods to achieve better performance in object detection. However, this method is proposed for datasets with imperfect training samples, and the implementation of the algorithm is complicated.

Thus, we propose three active learning algorithms and develop an active object detection framework for sonar images in this paper. In particular, we extend the existing active learning algorithm for image classification to the case of object detection through the use of a proper number of candidate boxes acquired by the detector model to calculate the informativeness of the instance. In addition, we exploit the location information provided by the object detection model as a new way to select data. Our proposed methods integrate active learning into object detection tasks in a continuous fashion to make CNNs more amenable to sonar image analysis to dramatically reduce the annotation cost.

The remainder of this paper is organized as follows. We describe the proposed methods in detail in Section II. The comparative experimental results and a brief discussion of them are provided in Section III. In Section IV, we present the conclusions.

II. THE PROPOSED METHODS

A. OVERVIEW

We first describe the proposed active learning algorithms in detail in section B. We then provide a brief introduction to the Faster-RCNN object detection framework. This is followed by an overview of the architecture of the proposed active learning framework for object detection in section D.

B. THE PROPOSED ACTIVE LEARNING ALGORITHMS

1) UNCERTAINTY SELECTION

For classification problems, uncertainty sampling is the simplest and most commonly used query framework. In this framework, an active learner queries the most uncertain instances for the probabilistic learning models. For simplicity, we use $P_\theta(y/x)$ to denote the probability that sample x belongs to category y under model θ. The general uncertainty sampling strategy usually uses entropy [21] as a measure of the uncertainty, which is computed by

$$x_H^* = \arg\max_x - \sum_{i=1}^{L} P_\theta(y_i/x) \log P_\theta(y_i/x), \tag{1}$$

where y_i $i = 1, \cdots, L$ ranges over all the possible category labels, and x_H^* refers to the most informative instance according to the uncertainty selection from all training data. However, this strategy cannot be applied directly to solve object detection tasks because an image may contain multiple objects to be detected. To overcome the difficulty caused by the uncertain number of objects in each image, in this paper, we develop an approximate method comprising two steps. First, we count the number of objects in each image; however, in actual active learning object detection problems, images are not annotated, and the number of objects in each images is unknown. Based on the current detection model, we assume that detected object windows with higher scores are more reliable than ones with lower scores. Thus, for active learning object detection, we assume that the object windows with confidences higher than a threshold gt are ground-truth boxes in each unlabeled image. The threshold g_t, which is used to approximately select the ground-truth boxes. As in actual active-learning-based object detection problems, the images are not annotated, and we have to first estimate the ground-truth boxes based on the current model. The selection process is similar to determining the correct detections in the output. At the output end, each output box is associated with a category label and a softmax score in [0,1]. Thus, the detected object boxes with confidences higher than a threshold g_t are used as the ground-truth boxes of the objects in each unlabeled image. In this paper, we follow the most popular architectures, namely, two-stage architectures (eg. Faster RCNN [7]) and one-stage architectures (eg. SSD [6]) A softmax score threshold of 0.6 is used to select the output box in [7] and [6], which indicates that this threshold can effectively select the output box that performs well, therefore we use 0.6 as the empirical threshold. g_t is an empirical value, and we will not set it too high because in the first few iterations, the detection model cannot generalize well, and a high threshold will filter out many real objects. Then, we can obtain the average number of objects in each image N_{avg} by dividing the total number of objects by the number of images. Based on the estimated object number in each image, we obtain N_{most}, which is the number of objects exceeding that in the majority of the images. More specifically, in our experiments, 80% of the images in our dataset contain at most N_{most} objects, but the percentage should be determined based on the dataset. We recommend that the choice of this value not be less than 80%, as the selection of N is a rough estimate, to ensure that each instance can calculate the informativeness using a number of candidate boxes that is no less than the number of its real objects, but this inevitably produces errors in the informativeness calculation of instances where the number of real objects is less than N).

Once we have determined the value of the parameter N, we select the N boxes with the largest probability from all the candidate boxes as the samples for uncertainty measurement according to the confidence scores of all the candidate boxes. For an instance x, we use Entropy to measure these boxes: entropy is an information-theoretic measure that represents the amount of information needed to encode a distribution. Therefore, we use entropy to calculate the uncertainty of each bounding box. Assuming that for each image i, we have B bounding boxes($b_i, \cdots, B$) and C classes, then for each bounding box, Faster RCNN predicts its softmax probabilities for each object class and background. $p(b_{ic})$ denotes the probability that bounding box b_i belongs to class c. Finally, we can use entropy as a measure of the uncertainty of the bounding box b_i:

$$E_{b_i} = -\sum_{c=0}^{C} p(b_{ic}) \log(p(b_{ic})) \tag{2}$$

where $c = 0$ represents the background.
where $p(b_{ic})$ is the softmax probabilities with the corresponding class. (In the object detection framework Faster RCNN, every bounding box is predicted by softmax probabilities for each object class and background.) It is the possibility that the candidate box contains objects with the corresponding category based on the existing detection model. N is the number of candidate boxes considered for calculation.

As mentioned above, we first roughly estimate the value of N, so in the actual calculation, we further screen the probability information to achieve more accurate informativeness calculations by setting a threshold denoted by *thresh*. If the probability of a candidate box is less than the threshold *thresh*, we do not take it into consideration but use g to record this As in the previous step, we choose N_{max} as the estimated number of objects in all the figures, which is a relaxation estimation. We further select partial bounding boxes by setting the probability threshold *thresh* for removing the hard examples contained in part of the figures for a more accurate informativeness calculation. In [5], [7] they used a threshold of 0.1 to filter out easy negative examples and sample object proposals that have a maximum IoU with the ground truth in the interval [0.1, 0.5) as background examples, in this paper, we set *thresh* to 0.1 by default. We add this operation based on the truth that N is a rough estimate, and there must be some instances for which the number of objects is less than N. Thus, if we use N candidate boxes to calculate the informativeness, we inevitably take mis-detected boxes into consideration. For example, if N is set to 5 and an image only has 2 objects, then we find that only two of all the candidate boxes have a probability that is greater than 90%, and the probability of the other mis-detected boxes is less than 15% or even 5%; these low-probability mis-detected boxes have an adverse effect on uncertainty selection, so we ignore them. During the experiment, the selection of the threshold should be set according to the specific situation. It is an empirical value, and it should be guaranteed that in the worst case, the threshold can filter out as many useless candidate boxes as possible. However, there is a special case that needs attention. If the value of g is equal to N, then $E_x = 0$, and the image will be directly filtered out. However, the existing model has poor detection in this image and is in great need of such an image to improve the model, so the index of this picture is therefore directly added to the final candidate queue. The overall uncertainty selection algorithm is presented in Algorithm 1.

Algorithm 1: Uncertainty Selection

```
Input:
Unlabeled dataset $D_u$, pre-trained CNN $M_0$
Batch size $b$
output:
Labeled candidates $D_l$
Retrained CNN model at iteration $t$: $M_t$
Initialize:
Set $D_l = \emptyset$, randomly select $b$ images from $D_u$, then label
  them, add them to $D_l$, remove them from $D_u$, and set
$t = 1$
Method:
Step 1:
    Train an initial detection model $M_1$ using $D_l$
Repeat
Step 2:
    for each image $C_i \in D_u$ do
        Obtain the detection information $W_i$ from $M_t$ and
        set $Obj_i = 0$ (the number of objects in $C_i$)
        for each detected object window $w_k \in W_i$ do
            if $p(b_{kc} > gt$ then
                $Obj_i = Obj_i + 1$
            end
        end
    end
Step 3:
    Determine $N_{max}$ based on $Obj_i$
        $N = N_{max}$
Step 4:
    for each $C_i$ in $D_u$ do
        Set $E_{C_i} = 0$, $g = 0$ and select the top $N$ scoring
        detected object windows.
        for each $w_j$ in them do
            if $p(b_{jc}) > thresh$ then
                $E_{C_i} = E_{C_i} - \log(p(b_{ic})$
            else
                $g = g + 1$
            end
        end
        end
        Sort $D_u$ according to the $E_{C_i}$, add those examples
        whose $g = N$ to $D_l$, and count their number $b_1$
        Query labels for the top $b - b_1$ candidates, generate
$Q$
    Step 5:
        $D_l = D_l \bigcup Q$; $D_u = D_u \setminus Q$; $t = t + 1$
        Train detection model $M_t$ based on $D_l$
    Until the object detection performance is satisfactory
        or the model tends to be stable
```

2) UNCERTAINTY + DIVERSITY SELECTION

Although the use of uncertainty selection can achieve some considerable results, it may result in the additional selection of noise or redundant samples. Thus, the uncertain-labeled samples must be filtered by a diversity criterion to produce diversity-labeled samples with the minimum redundancy, which are highly representative samples. Specifically, diversity selection is achieved through the K-medoids clustering algorithm, which is an improvement of the K-means algorithm [22]. Unlike the K-means algorithm, the K-medoids algorithm can be used with any distance measure in place of the generally used Euclidean distance that is consistent with the mean computation. Statistically, the K-medoids algorithm is more robust to the outliers and strong noise in the images. The algorithm is summarized as follows.

K-medoids

```
    Input:
    $n$ samples and number of clusters $k$
    output:
    $k$ clusters with corresponding instances
    Initialize:
    randomly select $k$ of the $n$ data points as the medoids
    Method:
    Repeat
    Step 1:
        Associate each data point with the closest medoid;
        ("closest" here is defined using any valid
        distance metric, most commonly the Euclidean distance,
        Manhattan distance or Minkowski distance)
    Step 2:
        For each medoid $m$ do
            For each non-medoid data point $o$ do
                Swap $m$ and $o$, and
                compute the total cost of the current state
            end
        end
    Step 3:
        Select the state with the lowest cost
    Until there is no change in the medoid
```

The total number of object categories in the dataset is directly used as the parameter K in the clustering, and the output of the last convolutional layer in the network is also directly used as the high level features of each image. However, this output is three dimensional, and the computational complexity when directly using it is too high. Thus, the feature map is first converted into a feature vector by using the channel-wise mean [23] method to reduce the computational complexity and facilitate the use of the clustering method. After obtaining the clustering results, we select the same number of D images in each category as the final sampling result. If $\min\{Num^k : k = 1 \ldots m\} > D$, where Num^k denotes the number of instances in the k^{th} category, we randomly select D images from each category as the final result. If this value is less than D, we select all images of categories with Num^k less than D, and the remaining images are evenly selected from those categories with Num^k greater than D. We repeat the above selection until the number of images that are set reaches a total of b images per iteration. The uncertainty + diversity selection algorithm is presented in Algorithm 2.

Algorithm 2: Uncertainty + Diversity Selection

```
    Input:
    Unlabeled dataset $D_u$, pre-trained CNN $M_0$
    Batch size $b$
    output:
    Labeled candidates $D_l$
    Retrained CNN model at iteration $t$: $M_t$
    Initialize:
    Set $D_l = \emptyset$, randomly select $b$ images from $D_u$, label
      them, add them to $D_l$, remove them from $D_u$, and set
$t = 1$
    Method:
    Step 1:
        Train an initial detection model $M_1$ using $D_l$
    Repeat
    Step 2:
        Use the uncertainty method to first choose $2b$ candidates
        $D_f$ from $D_u$
        for each $C_i$ in $D_f$ do
            The output of the last convolution layer can be
            viewed as high level features $C_i^f$ of $C_i$;
            Then calculate the channel-wise mean of $C_i^f$ to
            generate condensed features $C_i^c$ as the final image
            descriptor.
        end
    Step 3:
        Run $K-medoids$ on the $D_f$ using features $C_i^c$ to obtain
        the clustering results. Select the set of diversity samples $D_d$
        by taking a total of $b$ samples from the clusters in
        uncertainty sample set $D_f$.
        A human annotates the images in $D_d$
    Step 4:
        $D_l = D_l \bigcup D_d$; $D_u = D_u \setminus D_d$; $t = t + 1$
        Train detection model $M_t$ based on $D_l$
    Until the object detection performance is satisfactory
          or the model tends to be stable
```

3) LOCATION INFORMATION SELECTION

Unlike in image classification problems (in which each image represents one object), multiple object windows may exist in one image in object detection problems such that it is impossible to directly determine the uncertain number of objects in each image. In addition, for object detection, the output of the model not only provides the probability of the presence of an object at each spatial position but also provides the position information of the candidate box, which can be used to select the most representative images.

The maximum mean discrepancy method was first introduced by Gretton *et al.* [24] to analyze and compare two different distributions, which they used to construct statistical tests to determine if two samples are drawn from different distributions. Thus, in this work, we can use the MMD to select the most representative images from a dataset by narrowing the difference between the object location distribution of selected images and that of all training images. To determine which images should be selected from the unlabeled dataset and which object windows should be used to calculate the difference between two distributions [25], we first define $\mathbf{M}$ as a matrix of size $N_i \times N_w$ comprising zeros and ones to measure the distribution of the image over its object windows. If the j^{th} object window belongs to the i^{th} image, let $\mathbf{M}(i, j) = 1$; otherwise, set it equal to 0.

By utilizing $\mathbf{M}$, the most representative images can be selected by minimizing the cost function as below:

$$\arg\min_{\alpha} \left|\left| \frac{\sum_{i=1}^{N_i} \alpha_i * \sum_{j=1}^{N_w} M_{i,j} * \phi(x_j) + \sum_{i=1}^{N_l} \phi(x_i)}{N_s + N_l} - \frac{\sum_{i=1}^{N_f} \phi(x_i)}{N_f} \right|\right|^2$$
$$s.t.\ \alpha \in \{0, 1\}, \quad \alpha^T 1_I = N_{si}, \qquad (3)$$

where $\boldsymbol{\alpha}$ is a vector composed of zeros and ones, which represents whether the image is selected or not. ϕ is a function in the unit ball of the reproducing kernel Hilbert space. N_s is the number of object windows in the selected images. N_{si} is the number of selected images. N_i is the number of unlabeled images. N_w is the number of object windows in the unlabeled images. N_l is the number of object windows in the labeled images. N_f is the number of object windows in all the images. $\mathbf{1}_I$ is an $N_i \times 1$ vector, the elements of which are all equal to 1.

For ease of calculation, the problem can be transformed into a quadratic programming problem by simultaneously relaxing $\boldsymbol{\alpha}$ from having only two values, 0 or 1, to having a continuous value in [0,1].

$$\min\ 0.5\alpha^T H\alpha + f^T\alpha$$
$$H = \frac{MK_{UU}M^T}{(N_s + N_l)^2}$$
$$f = [\ \frac{1_L^T K_{LU} M^T}{(N_s + N_l)} - \frac{1_F^T K_{FU} M^T}{N_f * (N_s + N_l)}\ ]^T + const,$$
$$s.t.\ \alpha \in [0, 1], \quad \alpha^T 1_I = N_{si}, \qquad (4)$$

where $\mathbf{U}$ is the set of unlabeled object windows, $\mathbf{L}$ is the set of labeled object windows, $\mathbf{F}$ is the set of object windows in all the images, and all of the object windows are stored as upper-left and lower-right normalized coordinates. K_{UU}, K_{LU}, and K_{FU} are three kernel Gram matrices between the object window datasets $\mathbf{U}$ and $\mathbf{U}$, $\mathbf{L}$ and $\mathbf{U}$, and $\mathbf{F}$ and $\mathbf{U}$. The Gaussian kernel is used in this work. $\mathbf{1_F}$ is an $N_f \times 1$ vector, and $\mathbf{1_L}$ is an $N_l \times 1$ vector, the elements of which are all equal to 1.

In (3), only N_s is unknown. Ideally, N_s represents the total number of objects in the selected image. However, since the number of objects in each image is unknown, it is difficult to determine the value of N_s. In this paper, we estimate it as $N_s = N_{si} \times N$, where N denotes the average number of objects in each image. The determination of N follows the same principle as choosing the value of N in the uncertainty selection subsection.

However, the above algorithm is based on the assumption that the position information of all the images is known; however, in most cases, the information is unknown. Therefore, in the actual calculation process, we will determine the category probability and the location distribution information of all the unlabeled data according to the existing model. We set a threshold to determine how many objects exist in each image based on the presence probability of each object's windows instead of using the top K scoring detections as reliably detected windows. Although using the top K scoring detections allows N_s to be easily determined, the assumption that the number of objects in each image is the same does not conform with the actual situation. The overall location information selection algorithm is presented in Algorithm 3.

Algorithm 3: Location Information Selection

```
Input:
Unlabeled dataset $D_u$, pre-trained CNN $M_0$
Batch size $b$
output:
Labeled candidates $D_l$
Retrained CNN model at iteration $t$: $M_t$
Initialize:
Set $D_l = \emptyset$, randomly select $b$ images from $D_u$, label
    them, add them to $D_l$, remove them from $D_u$, and set
$t = 1$
Method:
Step 1:
    Train an initial detection model $M_1$ using $D_l$
Repeat
```

```
  Step 2:
      Obtain the detection information $W_i$ from $M_t$, set $j = 0$
      Define $\mathbf{M}$ as a matrix of size $N_i \times N_w$, set all the elements
      in $\mathbf{M}$ equal to 0
      for each detected object window $w_k \in W_i$ do
          if $Score_k > gt$ then
              $M(i, j) = 1; j = j + 1$
          end
      end
      Add the corresponding information in $D_l$ to $M$
  Step 3:
      Solve Eq.4 using quadratic programming solvers, for example,
      CVX
      Obtain $b$ candidates for set $D_p$ from $D_u$
      A human annotates the images in $D_p$
  Step 4:
      $D_l = D_l \bigcup D_p; D_u = D_u \setminus D_p; t = t + 1$
      Train detection model $M_t$ based on $D_l$
  Until the object detection performance is satisfactory
        or the model tends to be stable
```

C. OVERVIEW OF THE FASTER-RCNN FRAMEWORK

We use the Faster-RCNN framework for object detection [7], which is shown in Fig. 1. The Faster-RCNN framework not only takes convolutional feature maps used by region-based detectors to generate region proposals but also shares convolutional features with the detection network, enabling nearly cost-free region proposals, which is also an effective way of improving object detection accuracy. It comprises the following two parts: (i) a region proposal network (RPN) generating a set of rectangular object proposals with the objectness scores and (ii) a fast-RCNN with an RoI-pooling layer and a few fully connected layers that output the class probabilities and the bounding boxes. The RPN takes the last shared convolutional feature map as input and runs a small network over it to modify the anchors. It has two sibling layers (for cls and reg). The region proposals generated by the RPN are then sent to the fast-RCNN for more accurate classification and regression. The fast-RCNN take an image and numerous region proposals from the RPN as inputs and then uses a region of interest (RoI) pooling layer to generate a fixed size feature vector for the final two output layers: (i) softmax probabilities for each object class including the background class and (ii) refined bounding box coordinates. In the active object detection framework, we update the Faster-RCNN model at each iteration and use the updated model to detect unlabeled images, storing the test results for subsequent use of the active learning algorithms. However, the RPN and the fast-RCNN are two separate networks and need to be trained independently. Thus, in our experiment, we adopt an approximate joint training. That is, we combine the RPN and fast-RCNN losses as one total loss, and the backward propagation takes place as usual. It is faster to implement and perform end-to-end learning while maintaining good accuracy.

III. EXPERIMENTS

In this section, we examine the performances of the proposed active learning approaches for object detection. The comparison to recent state of art in active learning for object detection (Specifically, Roy *et al.* [18] took inspiration from the paradigm of the querying-by-committee method [26] and

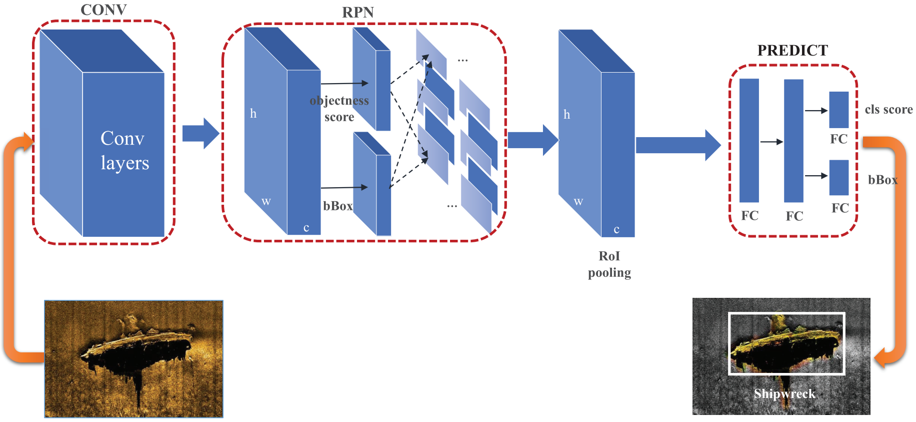


FIGURE 1. Training procedure of the object detection network based on Faster-RCNN. A region proposal network (RPN) takes an image (of any size) as input and outputs a set of rectangular object proposals, each with an objectness score to estimate the existence probability of an object belonging to a category. The input image and multiple object proposals are then input into an RoI (region of interest) pooling layer, and each object proposal is pooled into a fixed-size feature map and then mapped to a feature vector by fully connected layers (FCs). The network has two output vectors per object proposal: softmax probabilities and per-class bounding-box regression offsets.

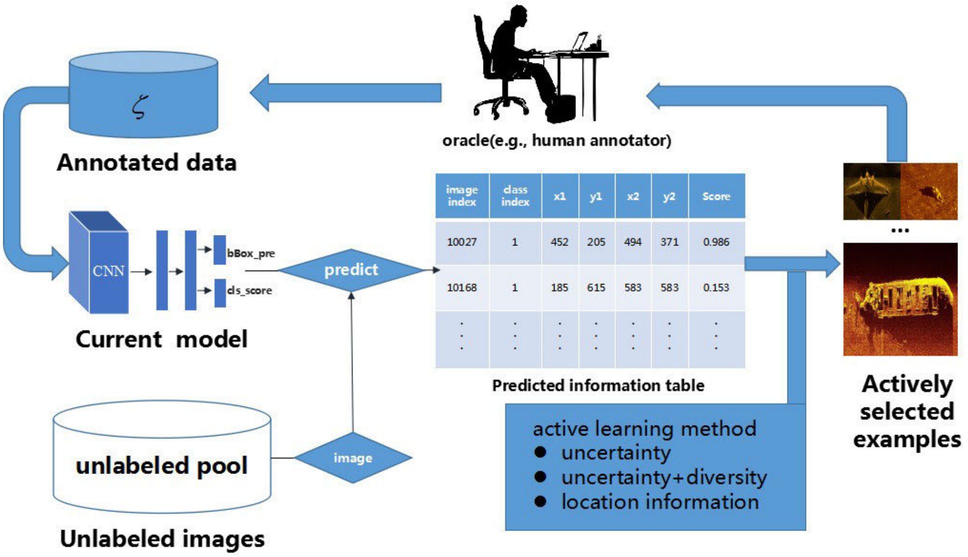


image index	class index	x1	y1	x2	y2	Score
10027	1	452	205	494	371	0.986
10168	1	185	615	583	583	0.153
⋮	⋮	⋮	⋮	⋮	⋮	⋮

FIGURE 2. The proposed active-learning-based framework for object detection in sonar images.

uses the disagreement between the convolution layers in the SSD [6] architecture to query images. Kao *et al* [27] presented two different metrics –the ‘‘localization tightness’’ and the ‘‘localization stability’’ to quantitatively evaluate the localization uncertainty of an object detector and combine them with the classification uncertainty.) are also considered in this section.We considered a detection correct when the area of overlap a_0 between the predicted bounding box B_p and ground-truth bounding box B_{gt} exceeds 50% by the formula: $a_0 = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}$ [28], [29]. We will first introduce our dataset and evaluation metrics. The detection results of the proposed active learning methods and the comparative method are then shown. Finally, a comprehensive analysis is provided.

A. ACTIVE LEARNING PERFORMANCE

1) DATA

We focus on the subject of object detection of sonar images in the context of underwater search and rescue. There is no public dataset and almost no possibility to perform a real experiment to obtain many more images. The purpose of our active learning algorithms is to maintain the performance of detection models with less data training on a group of selected images. To this end and according to the general standard of building a dataset of natural images, we built a sonar image dataset that satisfies the following three requirements:

(a)The dataset contains remarkable variety in terms of object size, illumination, position and noise distribution; (b) it is important that the dataset does not illustrate systematic bias such as a preference for images that contain centered objects with ideal illumination and orientation; (c) the annotations of each image need to be consistent, precise and exhaustive in both procedures of collecting images and choosing classes. Following the above three standards, the specific process is described as follows:

Step 1: We gathered a set of 216 sonar images that were captured by side scan sonar (SSS) and synthetic aperture sonar (SAS) from public photo-sharing websites in the context of underwater search and rescue. None of these images were previously used with the purpose of object detection, which guarantees that these images are not biased. Then, by random cropping, we obtained 25 samples from each original image for the purpose of data augmentation. The cropping size was 80% of the original size on the along-track and across-track directions, respectively.

Step 2: We resized the 5400 samples to the size of the images in 600*600 pixels in image annotation procedures and marked each object in the samples with an annotation that contains two attributes following the guideline of building a PASCAL VOC dataset [28]. (One attribute is a class, which indicates the category to which the object belongs. The other attribute is an axis-aligned bounding box that surrounds the extension of the visible part of the object in the image.

Step 3: Finally, we split the samples into two datasets, one of which is composed of 4300 samples for training, and the other contains the remaining 1100 samples for testing. None of the samples from the same original images are split into two different datasets, which guarantees that there are no intersections between the training set and the testing set.

The dataset for this experiment includes the corpse, shipwreck and plane wreckage categories and a total of 5400 images. We split it into two datasets, one dataset comprising 4300 images for training, and the other one containing the residual images for testing. First, $b = 100$ images were randomly selected according to the proportion of the categories in all the data. (This operation is based on some experimental results from [25], [31], in which they find at the first

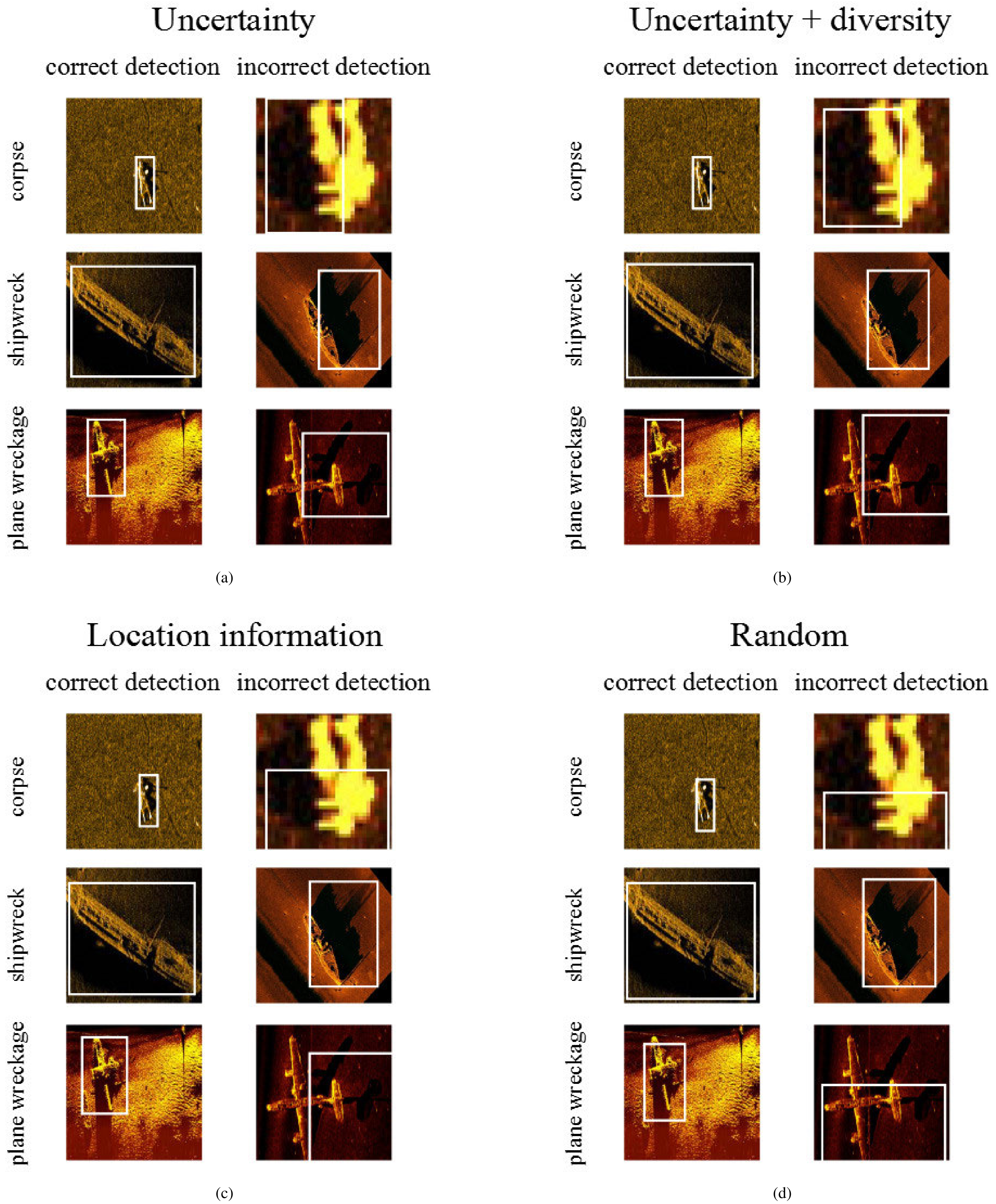


FIGURE 3. Some detection results obtained with the proposed active-learning framework detector trained on 1500 images for the three categories (corpse, shipwreck, and plane wreckage). The active-learning framework provides accurate localization in most of the test images, while the inaccurate positioning is mainly caused by the existence of deep shadows in the images. It can also be found that the intersection-over-union(Iou) of uncertainty and uncertainty + diversity for correct detections yields a bit better performance than that of random selection method.

step that random selection yields the best performance. The uncertainty selection strategy is an informativeness-based approach. Informativeness-based approaches completely rely on labeled data for constructing the initial model to select the query instance, which means that the model is always unstable when too few initial training data points are used in the first steps. In contrast, representative-based methods [32] may achieve a relatively better performance when there are little or no initial labeled data.) Then, a fixed number of data points (100 in our experiments) is continuously selected from the remaining 4200 images according to the principles of the proposed active learning algorithms. Because ground-truth boxes are not given and the number of objects in each images is also unknown, in this experiment, we take object windows whose *Score* is higher than $gt = 0.5$ as a reliable detection. The last thing to note is that the distribution of the entire dataset is unbalanced. Corpse occupies nearly 10%, plane wreckage occupies almost 20% of total data, and the remaining is shipwreck.

2) EVALUATION METRICS

Evaluating the performance of a classifier always occurs through observing how the precision and recall values change when the threshold changes. A better classifier performs as follows: the proportion of the target objects in the identified image is relatively large, and as many target objects as possible are correctly detected before identifying other objects. That is, let recall grow while maintaining precision at a high level, which leads to a higher average precision (AP) from the perspective of computation. Thus, we use the mean average precision (MAP), which is commonly used in most international detection competitions, in this experiment.

3) IMPLEMENTATION DETAILS

For our sonar dataset, we use the pre-trained resnet model [33] adopted in [7] to initialize our network. The parameters of newly added convolutional layers and fully connected layers are initialized with Xavier [34]. The input image is resized for better detection of small objects such that its shorter side has 600 pixels. All experiments are fine-tuned on the pre-trained ImageNet model. The baseline model using all images is trained for 20K iterations with an initial learning rate of 0.001, which is then divided by 10 at 10K iterations for better convergence of the model. The experiments of the random selection method are repeated 5 times, and the average performances are reported. For the data selected by the proposed methods, we train the network multiple times to obtain the best results. We use horizontal image flipping as the only form of data augmentation unless otherwise noted. The entire network is trained with stochastic gradient descent (SGD) with a momentum of 0.9 and a weight decay of 0.0001 on a single NVIDIA GeForce GTX TITAN X GPU with 12 GB memory. Each mini-batch involves only 1 image per GPU and 512 RoIs per image. The classification loss is the softmax loss, and the standard smooth L_1 loss is used for box regression [5]. Some of the test results are shown in Fig.7.

4) UNCERTAINTY SELECTION RESULTS

We set the value of *thresh* to 0.1 and train the initial Faster-RCNN [7] model on the first set of randomly selected 100 images. We then use the trained Faster-RCNN [7] model to test the remaining unlabeled images and set $N = 1$ for the first step of uncertainty selection based on the testing results. The MAP performance is illustrated in Fig.4. It can be found that constantly selecting new data for training through uncertainty selection can make the MAP rise steadily, and using nearly 35% of all the data achieves almost the same result as when using all the images. Uncertainty selection methods quickly surpass random selection after the first fine-tuning, as they select important samples for fine-tuning, making the training process more efficient than just randomly selecting from the remaining training dataset. We can also find that the greatest MAP increase (from 0.37 to 0.664) is obtained by uncertainty selection when the first group of 500 images is selected. The remaining images, which continuously increase from 500 to 3800, only make the MAP increase by 0.127. The steeper red solid curve indicates improvement in the accuracy on the test set using fewer images and verifies the effectiveness of the uncertainty selection algorithm, which can select the most informative samples. A possible explanation is that the first group of selected images based on uncertainty already has the most informative instances for training and that the remaining instances are not as informative as the first selected 500 images. Fig.5 shows the average precision performance for the three categories. In Fig.5(a), a notable point is that the corpse average precision (AP) based on 1500 selected images (0.612) is higher than that based on all the training images (0.543). This result more or less reflects the possibility that the uncertainty algorithm can alleviate the imbalance of the data distribution to some extent and obtain better corpse AP performance. This is because the number of corpses is the lowest, and the model may not be able to fully learn the features of this category and

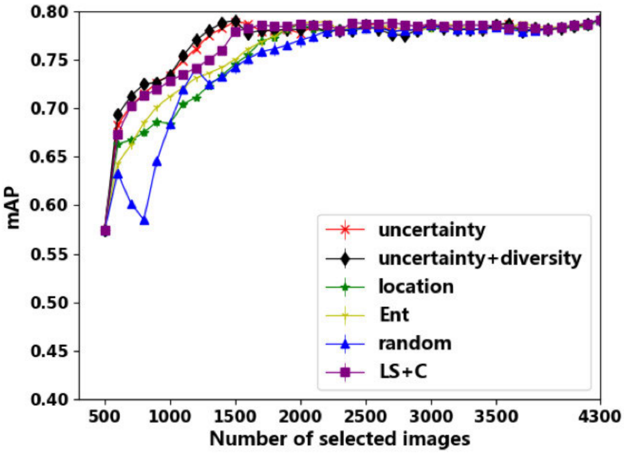
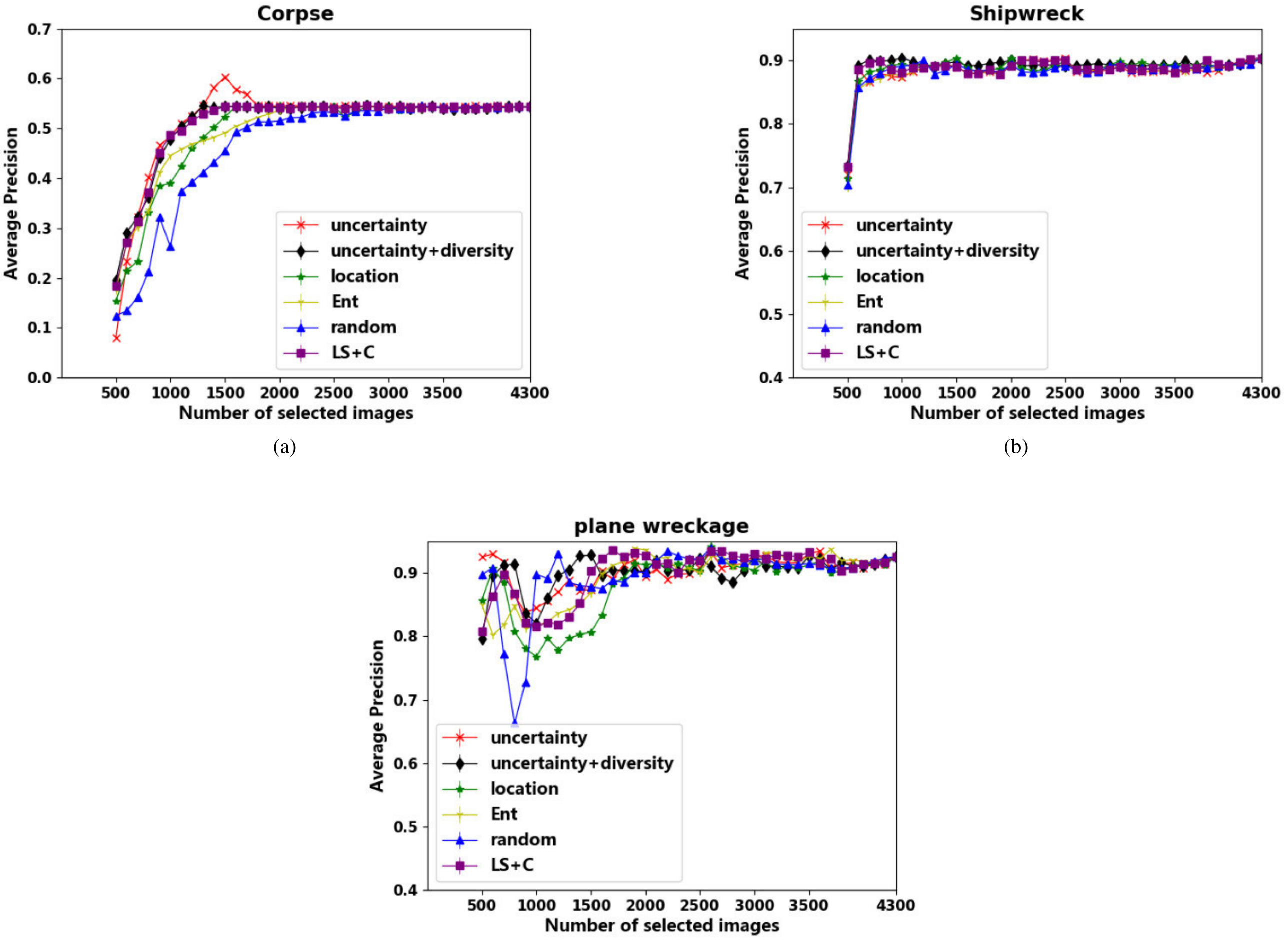


FIGURE 4. Mean average precision curve of different active learning methods (these results are obtained using Faster R-CNN as the detector) on our sonar dataset. The uncertainty, uncertainty + diversity selection and LS+C methods achieve better performance than random selection after the first fine-tuning step, and with only 1500 selected images, they achieve almost the same results as those of random selection using all the images. Although location information selection is not stable in the first few iterations, it still reaches its best performance when querying approximately 2000 samples.



(a)

(b)

(c)

FIGURE 5. Performance comparison for three different categories (these results are obtained using Faster R-CNN as the detector), (a) corpse, (b) shipwreck and (c) plane wreckage. The graphs are plotted as average accuracy (y-axis) vs. number of training images (x-axis). The general performance is consistent with the MAP curves shown in Fig. 4. The AP for detecting shipwrecks is stable for all the methods, as the number of images in this category is dominant in the training set. There are some fluctuations in the results for corpses and plane wreckage for different selection methods.

accurately locate objects. However, the uncertainty algorithm selects more images that contain corpses, causing the model to have better detection performance for corpses. In Fig. 5(b), the MAP of the shipwreck category is almost saturated when 600 images were selected. There are three reasons: (1) the number of pictures of the shipwreck is dominant in the training set; (2) the shipwreck features are are usually similar to each other. and (3) the shapes and fragments around plane wreckage objects are irregular, which results in a large difference in features between objects. In Fig. 5(c), the red solid line continues to fluctuate as the number of selected pictures increases. This is mainly caused by the large-scale variation across object instances in plane wreckage images which inevitably hampers the accuracy of detectors and makes the detection performance unstable.

5) UNCERTAINTY + DIVERSITY SELECTION RESULTS

Diversity selection is performed subsequently after uncertainty selection. First, we choose 200 images based on the above uncertainty selection and then run K-medoids on them to finally choose 100 images. In general, the MAP grows stably when more images are selected, yielding a slightly better performance than uncertainty in the first group of 1500 images selected from the dataset, and can achieve almost the same result as that when using all the images. Different from uncertainty selection, the AP of which has a greater improvement when detecting the corpses with 1500 images than when using all the training images, the AP values for detecting all three categories via uncertainty + diversity are very close to those using all the training images. This is attributed to combining the diversity selection with uncertainty selection because the diversity selection can further efficiently maintain the balance of the numbers of the three categories in the candidates selected from uncertain instances first selected by uncertainty selection. To a certain extent, this approach avoids the extreme situation in which all selected images are from the currently poorest detection category.

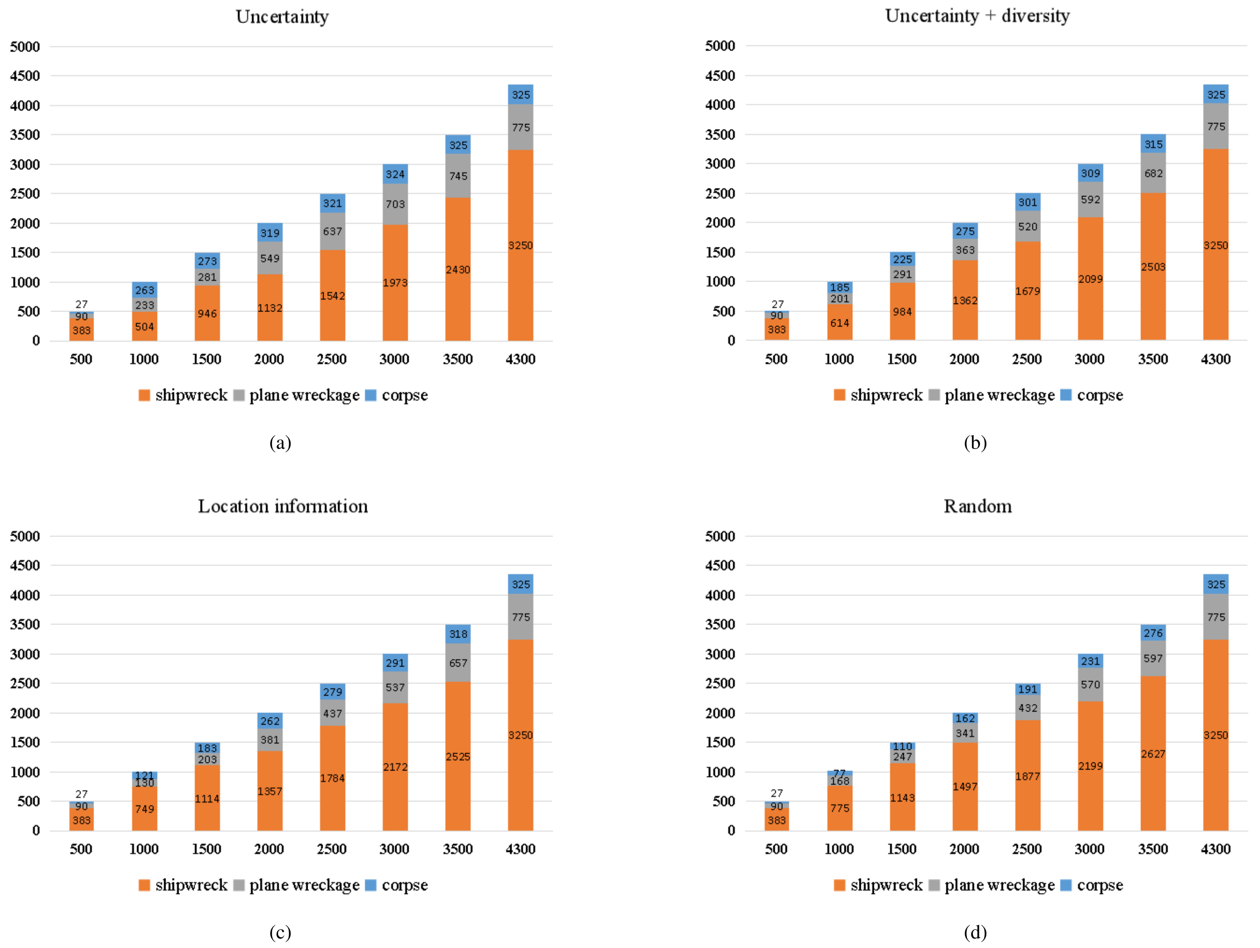


FIGURE 6. The numbers of corpses, shipwrecks and plane wrecks selected by the four methods in each group of samples. The orange bars represent the selected number of shipwrecks, the blue bars denote that of corpses, and the gray bars represent that of plane wrecks. More than 80% of the corpse images are selected from the unlabeled dataset after the first uncertainty selection, making the ratios of the three categories close to approximately 1:2:1, which greatly alleviates the unbalanced data distribution problem and obtains a better corpse average precision (AP) performance.

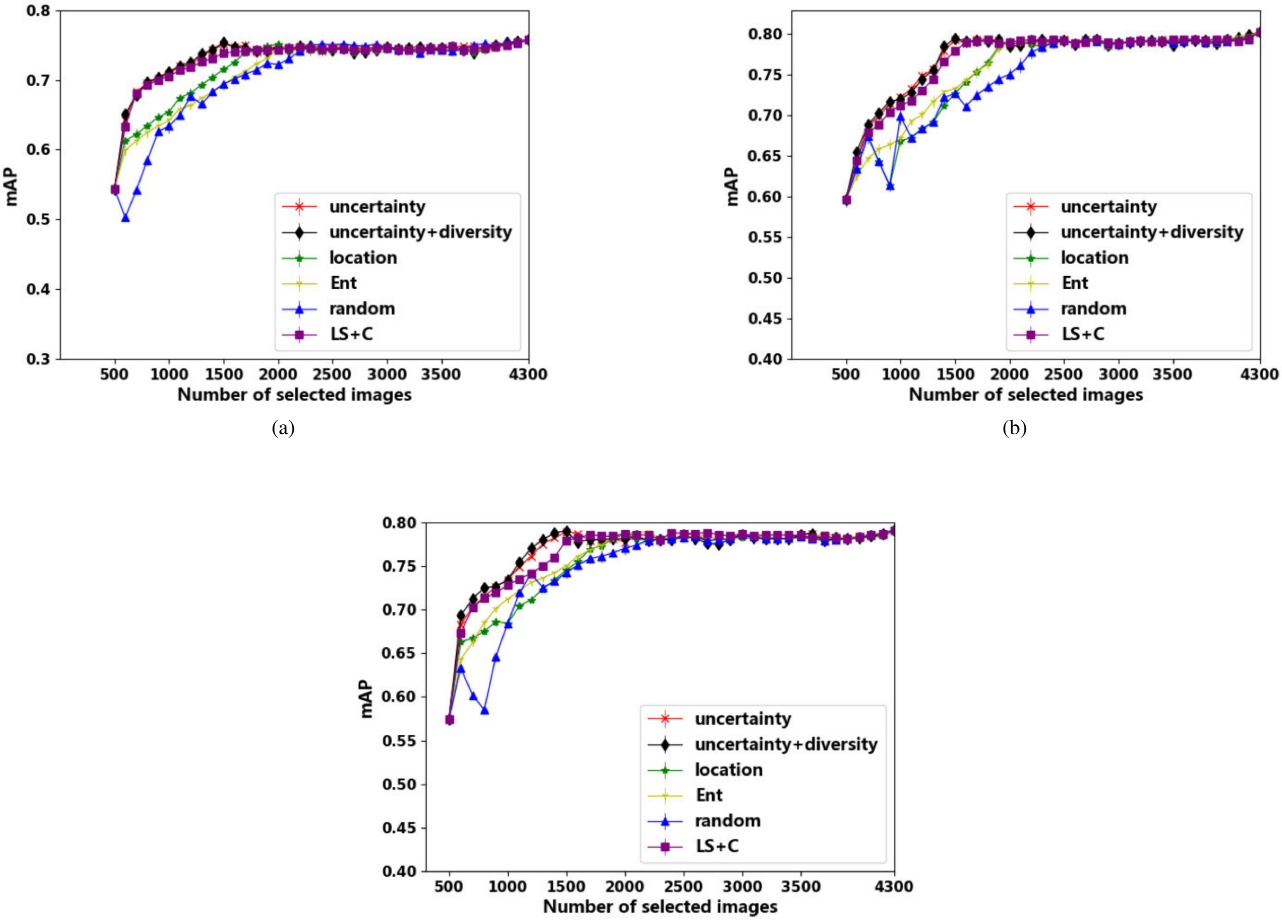
6) LOCATION INFORMATION SELECTION RESULTS

As α is a continuous value in [0,1], we choose the top $b = 100$ indexes as the selected results. Based on the detection model trained on the first group of 100 images, which is randomly selected, we set $N_s = N_{si} * 1$ for the first time of the selection. Generally, the green MAP dash-dot line of location information selection in Fig.4 shows an upward trend when more images are selected, and the use of 2000 images can achieve almost the same result as that obtained by using all the images. In Fig.6(c), we can also find that the location information selection performance is not very stable in the first few steps. This fluctuation is caused by some inaccurate location prediction. In the first few steps, the detection model does not generalize well due to the lack of sufficient data, so there must be some mis-detected object windows or actually existing but undetected objects, which inevitably affect the selection of this algorithm.

7) OVERALL RESULTS

In this section, we compare the three proposed active learning methods with random selection and conduct an overall analysis. The performance of each method becomes saturated after querying 2500 labels. Uncertainty and uncertainty + diversity converge faster among the four methods and yield better overall performance. This is attributed to the informativeness calculation method proposed for uncertainty selection. The uncertainty and uncertainty + diversity selection methods with only 1500 images can achieve the same performance as that of the random selection method with 3000 selected images. Compared to random selection, 50% of the labeling cost could be saved by the uncertainty and uncertainty + diversity methods. When 1500 images are selected by the uncertainty or uncertainty + diversity method, the detection model tends to be stable. The same situation occurs when 2000 images are selected by the location information method and when 2500 images are selected by the random selection method.

Our sonar dataset is unbalanced, and we have already noted the possibility that the uncertainty algorithm can alleviate the data distribution imbalance in the above discussion. Furthermore, we test the ratio of the three categories in the images selected by the proposed methods. We set $b = 500$ for



(a)

(b)

(c)

FIGURE 7. Result comparisons of the three detectors. (a) Results of YOLO-based framework; (b) Results of SSD-based framework;(c) Results of Faster RCNN-based framework.

this experiment, as more data can better reflect the distribution of data. Fig.6 shows the number of selected images in each category by the uncertainty, uncertainty + diversity, location information and random selection methods. As we expected, more than 80% of the corpse images are selected from the unlabeled dataset after the first uncertainty selection, making the ratios of the three categories approximately close to 1:2:1, which greatly alleviates the problem of the unbalanced data distribution and achieves a better average precision (AP) performance for corpse. Uncertainty selection and uncertainty + diversity control the balance of the numbers of selected images in the three categories and cause the model's detection capability to grow rapidly and steadily. For random selection, the ratio is nearly the same as for the entire training dataset. In the first few steps of location information selection, the selected numbers of corpses and plane wreckage are very close. However, there are gaps between these numbers and that of shipwrecks. This is because the number of shipwreck images is dominant in all of the training sets, and the location information selection method takes the entire training dataset's distribution into consideration.

8) EXTENSION TO THE OTHER COMPETING DETECTORS

In this subsection, we also combined the proposed active-learning algorithms with the other competing detectors (YOLO and SSD) and compared these three detectors to improve the quality and usefulness of the results. The results and the corresponding analysis are shown in Fig.7.

Generally, the three active-deep learning frameworks all work while the SSD-based and Faster-RCNN-based frameworks perform better than the YOLO-based one. The uncertainty-type frameworks generally perform better according to the effective selection ability of the more informative images. Thus, the proposed active-deep-learning framework is extensible and has a strong generalization ability.

IV. CONCLUSION

To reduce the annotation cost as much as possible, in this paper, we propose three active learning algorithms. They all start with a randomly selected dataset and incrementally improve the CNN's performance through continuous retraining by actively selecting the most informative and representative images. Their performances are tested in our sonar dataset for object detection, and the results demonstrate that the annotation cost can be reduced by at least 60%. In the future, weakly supervised object detection methods [35], [36] may be utilized to replace the previously trained model for generating detected object windows. Moreover, since the proposed active learning methods retrain the detection model at each iteration, which is time consuming, more efficient updating strategies will be explored in future work.

REFERENCES

[1] G. J. Dobeck, J. C. Hyland, and L. Smedley, "Automated detection and classification of sea mines in sonar imagery," *Proc. SPIE*, vol. 3079, pp. 90–110, Jul. 1997.

[2] M. E. Clarke, N. Tolimieri, and H. Singh, "Using the seabed AUV to assess populations of groundfish in untrawlable areas," in *The Future of Fisheries Science in North America*. Dordrecht, The Netherlands: Springer, 2009, pp. 357–372.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[5] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2980–2988.

[9] M. Valdenegro-Toro, "Object recognition in forward-looking sonar images with convolutional neural networks," in *Proc. OCEANS MTS/IEEE Monterey*, Sep. 2016, pp. 1–6.

[10] J. Kim and S.-C. Yu, "Convolutional neural network-based real-time ROV detection using forward-looking sonar image," in *Proc. IEEE/OES Auton. Underwater Vehicles (AUV)*, Nov. 2016, pp. 396–400.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[12] M. Valdenegro-Toro, "Objectness scoring and detection proposals in forward-looking sonar images with convolutional neural networks," in *Proc. IAPR Workshop Artif. Neural Netw. Pattern Recognit.*, 2016, pp. 209–219.

[13] J. Kim, H. Cho, J. Pyo, B. Kim, and S.-C. Yu, "The convolution neural network based agent vehicle detection using forward-looking sonar image," in *Proc. OCEANS MTS/IEEE Monterey*, Sep. 2016, pp. 1–5.

[14] B. Settles, "Active learning," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 6, no. 1, pp. 1–114, 2012.

[15] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2372–2379.

[16] X. Li and Y. Guo, "Adaptive active learning for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 859–866.

[17] S. Vijayanarasimhan and K. Grauman, "Large-scale live active learning: Training object detectors with crawled data and crowds," *Int. J. Comput. Vis.*, vol. 108, nos. 1–2, pp. 97–114, 2014.

[18] S. Roy, V. P. Namboodiri, and A. Biswas, "Active learning with version spaces for object detection," Nov. 2016, *arXiv:1611.07285*. [Online]. Available: http://arxiv.org/abs/1611.07285

[19] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013.

[20] P. K. Rhee, E. Erdenee, S. D. Kyun, M. U. Ahmed, and S. Jin, "Active and semi-supervised learning for object detection with imperfect data," *Cognit. Syst. Res.*, vol. 45, pp. 109–123, Oct. 2017.

[21] A. Holub, P. Perona, and M. C. Burl, "Entropy-based active learning for object recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2008, pp. 1–8.

[22] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, Oakland, CA, USA, 1967, vol. 1, no. 14, pp. 281–297.

[23] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, "Suggestive annotation: A deep active learning framework for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2017, pp. 399–407.

[24] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, Mar. 2012.

[25] Y. Liu, Y. Wang, and A. Sowmya, "Batch mode active learning for object detection based on maximum mean discrepancy," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Nov. 2015, pp. 1–7.

[26] P. Melville and R. J. Mooney, "Diverse ensembles for active learning," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 74.

[27] C. C. Kao, T. Y. Lee, P. Sen, and M. Y. Liu, "Localization-aware active learning for object detection," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 506–522.

[28] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[30] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[31] Z. Zhou, J. Shin, L. Zhang, S. Gurudu, M. Gotway, and J. Liang, "Fine-tuning convolutional neural networks for biomedical image analysis: Actively and incrementally," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4761–4772.

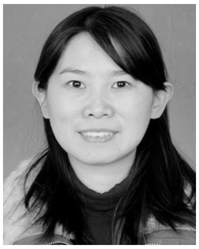
[32] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 705–712, 1996.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[34] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[35] H. Bilen, M. Pedersoli, and T. Tuytelaars, "Weakly supervised object detection with posterior regularization," in *Proc. BMVC*, 2014, pp. 1–12.

[36] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2846–2854.



LONGYU JIANG (Member, IEEE) received the Ph.D. degree in signal processing from Grenoble University, Grenoble, France, in 2013. Since 2013, she has been an Associate Professor with the Department of Computer Science and Engineering, Southeast University, Nanjing, China. Her major research interests include array signal processing, underwater acoustics, and object detection in sonar images.

TAO CAI is currently pursuing the master's degree in computer science and engineering with Southeast University, Nanjing, China. His major research interest is active learning.

QIXIANG MA is currently pursuing the master's degree in computer science and engineering with Southeast University, Nanjing, China. His major research interest is object detection in sonar images.

FANJIN XU is currently pursuing the master's degree in computer science and engineering with Southeast University, Nanjing, China. His major research interest is active learning.

SHIJIE WANG is currently an Associate Professor with the Department of Computer Science and Engineering, Southeast University, Nanjing, China. His major research interest is medical image processing.