DIAMONDNET: SHIP DETECTION IN REMOTE SENSING IMAGES BY EXTRACTING AND CLUSTERING KEYPOINTS IN A DIAMOND

Zicong Zhu[1,2,3], Wenhui Diao[1,2,*], Kaiqiang Chen[1,2], Liangjin Zhao[1,2], Zhiyuan Yan[1,2], Wenkai Zhang[1,2], Guangluan Xu[1,2], Xian Sun[1,2,3]

[1]Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China
[2]Key Laboratory of Network Information System Technology, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China - (whdiao, gluanxu, sunxian)@mail.ie.ac.cn, , (zhaolj004896, zhangwk)@aircas.ac.cn,
[3]University of Chinese Academy of Sciences, Beijing, China - (zhuzicong18, chenkaiqiang14, yanzhiyuan16)@mails.ucas.edu.cn

KEY WORDS: Ship Detection, Keypoints, Clustering, Rotated Bounding Box, Convolutional Neural Networks(CNN)

ABSTRACT:

Ship detection plays an important role in military and civil fields. Despite it has been studied for decays, ship detection in remote sensing images is still a challenging topic. In this work, we come up with a novel ship detection framework based on the keypoint extraction technique. We use a convolutional neural network to detect ship keypoints and then cluster the keypoints into groups, where each group is composed of keypoints belonging to the same ship. The choice of the keypoints is specifically considered to derive an effective ship representation. One keypoint is located at the center of the ship and the rest four keypoints are located at the head, the tail, the midpoint of the left side and the midpoint of the right side, respectively. Since these keypoints are distributed in a diamond, we name our network *DiamondNet*. In addition, a corresponding clustering algorithm based on the geometric characteristics of the ships is proposed to cluster keypoints into groups. We demonstrate that our method provides a more flexible and effective way to represent ships than the popular anchor-based methods, since either the rectangular bounding box or the rotated bounding box of each ship instance can be easily derived from the ship keypoints. Experiments on two datasets reveal that our *DiamondNet* reaches the state-of-the-art results.

1. INTRODUCTION

Ship detection in remote sensing images has played an important role in military and civil fields. It can not only help to supervise the fishery, but also facilitate the protection of maritime traffic (Zhu et al., 2010; Corbane et al., 2010). The research on ship detection has been extensively studied for several decades (Zhu et al., 2010; Yang et al., 2013), but it is still a challenging topic. Thanks to the rapid development of convolutional neural networks (CNNs) (Krizhevsky et al., 2012; He et al., 2016), it has been possible to achieve the accurate localization of ships in remote sensing images (Yang et al., 2018; Feng et al., 2019; Zhang et al., 2016; Fu et al., 2020; Wang et al., 2019) In this paper, we achieve the ship detection based on the keypoint extraction technique (Newell et al., 2016; Cao et al., 2017). Thereby, the detected keypoints and the derived bounding boxes are shown in Figure 1.

The mainstream of the ship detection methods (Jiang et al., 2017; Feng et al., 2019; Yang et al., 2018) or the common object detection methods (Girshick, 2015; Ren et al., 2015; Liu et al., 2016) are based on the anchor mechanism. These methods highly depend on a series of anchor boxes as the reference and require a set of complicated and heuristic rules to match the anchor boxes and the corresponding objects (Ren et al., 2015; Lin et al., 2017a). The design of the anchor boxes has to be carefully considered and tuned through experiments. To derive a better detection result, the two-stage detectors (Ren et al., 2015; Lin et al., 2017a) firstly use a Region Proposal Network (RPN) (Ren et al., 2015) to compute a series of class-agnostic potential bounding boxes. Afterwards, a category prediction and a finer location regression process has to be further made (Ren et al., 2015). Instead, the one-stage detectors (Zhang et al., 2016)

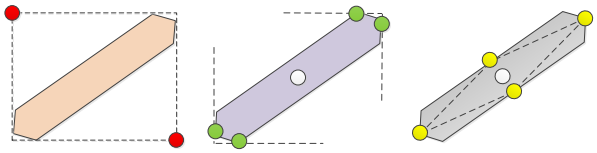
(a) (b) (c)

Figure 1. An example of the five keypoints of a ship and the derived ship detection results. (a) The definition of the ship keypoints: one point located at the center of the ship and four other points located at the head, the tail, the midpoint of the left side and the midpoint of the right side, respectively. (b) The regular rectangular bounding box derived from the keypoints. (c) The rotated bounding box derived from the keypoints.

are proposed to combine the two mentioned stages. However, the performance is not competitive to the two-stage detectors. In addition, these anchor-based methods (Ren et al., 2015; Liu et al., 2016; Lin et al., 2017a) usually derive the regular rectangular bounding boxes as presented in Figure 1(b), while a regular bounding box cannot represent a ship accurately as it usually contains a large portion of background pixels. Despite that some other anchor-based methods are proposed (Yang et al., 2018; Feng et al., 2019) to derive the rotated bounding boxes (as shown in Figure 1(c)), they require extra complicated operators. Therefore, the complexity and the inherent drawbacks of the anchor-based methods cannot be solved simultaneously.

In this work, we propose a simple yet effective ship detection framework based on the keypoint extraction technique (Newell et al., 2016; Cao et al., 2017), which simultaneously reduces the complexity and overcomes the inherent drawbacks of the anchor-based methods. On the one hand, compared with the

*Corresponding author



(a) (b) (c)

Figure 2. Three different keypoint definition strategies. (a) Two keypoints are located at the top-left corner and the bottom-right corner of the bounding box of an object. (b) Four extreme points are selected as the keypoints. (c) We define five keypoints: the center point and four other points located at the head, the tail, the midpoint of the left side and the midpint of the right side, respectively.

anchor-based methods, the convolutional neural network used for the keypoint extraction can be quite simple. Both the complicated match process (Ren et al., 2015; Lin et al., 2017a) and the RPN (Ren et al., 2015) are not necessary anymore. On the other hand, the keypoints provide a more flexible way to represent ships since the keypoints can further derive the regular rectangular bounding boxes (as shown in Figure 1(b)) and the rotated bounding boxes (as shown in Figure 1(c)). In this regard, the keypoints can effectively and accurately represent ships and our method based on keypoint extraction inherently overcome the drawbacks of the anchor-based methods.

We note that we are not the first to apply keypoint extraction to the object detection task, but to the best of our knowledge, we are the first to apply the idea to ship detection. Furthermore, our method is specifically designed for the ship detection task and different from the previous works (Law, Deng; Zhou et al., 2019; Duan et al., 2019) in three aspects: (1) the keypoint definition, (2) the keypoint learning and (3) the clustering algorithm.

The keypoint definition rules are different among the current methods (Law, Deng; Duan et al., 2019; Zhou et al., 2019). CornerNet (Law, Deng) chooses the top-left corner and the bottom-right corner of an object as the two keypoints. With these two keypoints, the regular rectangular bounding box can be easily induced (as can be seen in Figure 2(a)). However, these two keypoints are usually located outside the object, which may fail to represent the appearance features of the object and increase the difficulty of the extraction of keypoints. To address this, a novel *corner pooling* (Law, Deng) operator is proposed. Alternatively, ExtremeNet (Zhou et al., 2019) chooses four extreme points as the keypoints (as can be seen in Figure 2(b)). However, both these two kinds of selected keypoints are aligned with the image sides while the spatial positions of the keypoints relative to the objects are not fixed. Instead, we use five keypoints whose positions are fixed relative to the ships and we consider this way a better feature representaion for the ships. Specifically, we define the five keypoints as one center point of the ships and four other points located at the head, the tail, the midpoint of the left side and the midpoint of the right side, respectively (as can be seen in Figure 2(c)).

The definition of the keypoints determines the way how convolutional neural networks learn keypoints. An intuitive idea is to assign one kind of keypoints to a heatmap (Law, Deng; Duan et al., 2019; Zhou et al., 2019). For example, CornerNet (Law, Deng) predicts the top-left corners of all instances of one category in one heatmap and predicts the bottom-right corners in another heatmap. In this work, we consider the symmetrical geometry structure of ships and it would be difficult to distinguish between the midpoints of the left side and the right side as presented in Figure 2(c). Therefore, we use one heatmap for the prediction of the center point of the ship, one heatmap for the two keypoints located at the ship head and at the tail, and one heatmap for the two keypoints located at the midpoints of the left side and the right side.

With the keypoints extracted from the network, the last step towards to the ship instances is clustering the extracted keypoints into groups, where the keypoints in one group are supposed to belong to the same instance. CornerNet (Law, Deng) and CenterNet (Duan et al., 2019) predict a *associative embedding vector* (Newell et al., 2017) for each keypoint and then cluster keypoints based on the distance of the vectors. These methods require the network to output the new association embedding maps, which results in more learnable weights. ExtremeNet (Zhou et al., 2019) uses a brute force manner to enumerate all the possible cases. Instead, we propose a simple ship-customized clustering algorithm to derive the ship instances from the learned keypoints just based on the geometrical characteristics of the ships.

After briefly describing the related work in Section 2, we introduce our methodology for the ship detection in detail in Section 3. To demonstrate the performance of our method, a set of experiments and corresponding analysis on two datasets are prensented in Section 4. Finally, we provide the conclusions in Section 5.

2. RELATED WORK

At present, the mainstream object detection methods can be categorized into the two-stage methods and the one-stage methods. The two-stage methods first generate proposed regions, which are assumed to contain objects, and then make the object detection task through a classification and regression process from the proposed regions. Many reseaches focus on the first stage. In the early days, the independent region proposal algorithms are applied to generate regions (Girshick et al., 2014; Girshick, 2015). Later, the Region Proposal Network (RPN) (Ren et al., 2015) takes the place of the independent region proposal algorithms, which accelerates the speed through the feature sharing between the region proposal task and the object detection task and improves the accuracy via the altenative fine-tuning for the two tasks. In order to have a good performance for the multi-scale objects, the Feature Pyramid Network (FPN) (Lin et al., 2017a) fuses the intermedia features, and makes the detection based on the features of multiple scales instead of just based on the last features. Instead, the one-stage methods combine the region proposal task and the object detection task into one single stage (Liu et al., 2016), resulting in a higher inference speed but at the cost of reducing the accuracy.

The keypoint extraction is widely applied in human pose estimation and object component recognition (Zhu et al., 2017; Newell et al., 2016). These tasks estimate the pose of the object by detecting the key parts like the joints of human bodies or the corners of chairs. There are two main ways to implement these tasks: the top-down methods and the bottom-up methods. The top-down methods divide the task into two parts: object detection and keypoint extraction (Guler et al., 2018). On the contrary, the bottom-up method first extracts the keypoints and then clusters the keypoints into groups so as to find the keypoints belonging to a single object (Cao et al., 2017). It can be

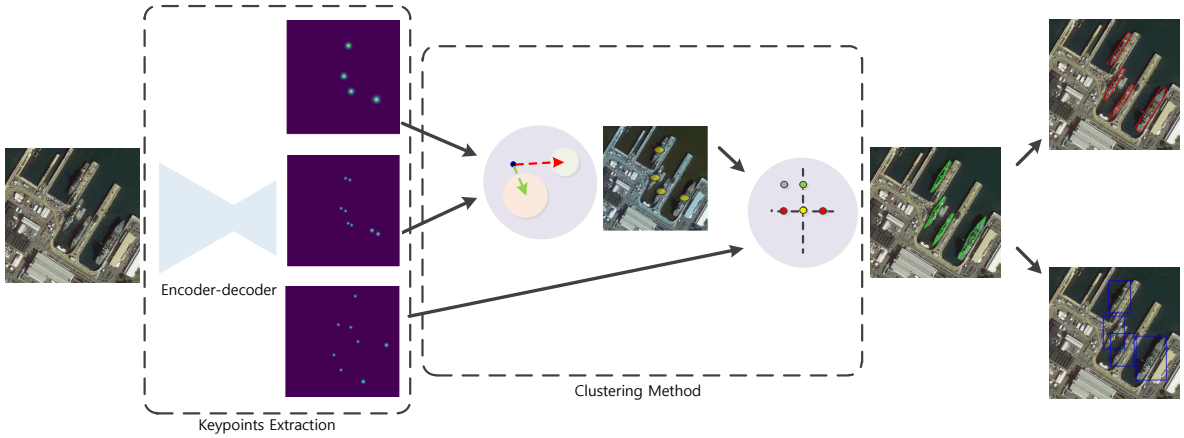


Figure 3. The complete flow of our method. The input image is first fed into an encoder-decoder network to extract keypoints. Then the keypoints are clustered into groups, where each group represent a single ship instance. Finally, both the regular rectangular bounding box and the rotated bounding box can be obtained by a simple transform method.

seen that keypoint extraction and object detection are inextricably linked.

In recent years, the keypoint extraction has been transferred to the field of object detection (Law, Deng; Zhou et al., 2019). The object detection task is divided into two steps: keypoint extraction and keypoint clustering. The keypoints, which can represent the position of an object, are extracted through networks. CornerNet (Law, Deng) localizes the object by detecting the top-left corner and the bottom-right corner and then clusters the keypoints based on the associative embedding vector dereived from the network. Instead, ExtremeNet (Zhou et al., 2019) extends the keypoints to the extreme points of the top side, the bottom side, the left side and the right side of an object respectively to get a more accurate object representation. The corresponding clustering algorithm is based on the geometric relationship of the relative position of the points.

Currently, ship detection methods are still dominated by the anchor mechanism (Yang et al., 2018; Zhang et al., 2016). To fit the high aspect ratio of ships, a series of specifically designed anchor boxes are preset (Yang et al., 2018). Furthermore, a Dense Feature Pyramid Network (DFPN) is put forward to build the high-level semantic feature maps across all scales and a rotated anchor box strategy is designed to achieve the rotated bounding boxes (Yang et al., 2018). Besides, some researches focus on the localization of the ship proposals by the saliency detection methods (Zhang et al., 2016), which are dedicated to matching the unique characteristics of ship. Afterwards, these proposals are fed into a trained CNN for a robust and efficient detection. Some other improvements are made in the classification stage by adding the inclined box regression in the classifier branch, and using the inclined Non-Maximum Suppression (NMS) algorithm to get the detection results in the form of rotated bounding boxes (Jiang et al., 2017). Instead of detecting ships based on anchors, we achieve the ship detection task through extracting and clustering the keypoints of ships in a diamond.

3. METHODS

In this section, the keypoint selection method is introduced in Section 3.1. Then we present how to extract the keypoints from the convolutional neural networks in Section 3.2 and the correpoinding cluster algorithm is presented in Section 3.3. As it is quite intuitive and easy to generate bounding boxes from the keypoints, the method to derive the regular rectangular bounding box and the rotated bounding box are not introduced in this work. The complete flow of our method is shown in Figure 3.

3.1 Keypoint Selection

The selection of keypoints is essential for the ship detection task. We define the ship keypoints out of the considerations in two aspects. Firstly, the keypoints must best represent a ship. In this regard, the keypoints should lie in a ship and be irrelevant to the background. Therefore, the corners as described in Figure 2(a) do not meet with this condition. Secondly, the position of the keypoints relative to the ships must be fixed and be independent of the position and the pose of the ships in an image. In this regard, the extreme points presented in Figure 2(b) do not meet with the requirement. We define the keypoints as presented in Figure 2(c). There are five keypoints in our definition, including one center point and four other points located at the head, the tail, the midpoint of the left side and the midpiont of the right side, respectively. All these keypoints lie in the ships and the positions are fixed relative to the ships.

3.2 Keypoint Extraction

To achieve the keypoint extraction, we simply regard it as a pixel-wise classification problem and categorize all the pixels into four types. One type represents the center point, one type represents the midpoints of the left side and the right side, one type represents the head and the tail, and the rest type represents the non-keypoint pixels. The midpoints of the left side and the right side are classified as the same type since in many cases the ships have a symmetrical structure and thus it is difficult to distinguish between the left side and the right side. The head and the tail are classified as the same type, despite they are distinguishable in most cases. The qualitative experiments in Section 4.3 reveal that this strategy is effective and derives a better keypoint extraction result.

To generate the heatmaps for keypoints, our *DiamondNet* uses a fully convolutional encoder-decorder structure as the most keypoint extraction researches (Zhou et al., 2019; Newell et al.,

2016). We use ResNet-34 (He et al., 2016) as the encoder to extract the features of input images, whose output stride is 32. In the decoder, we choose three groups of transposed convolution with a stride of 2, and each transposed convolution is followed by a deformable convolution (Dai et al., 2017) to have a more flexible feature recovery. After that, the feature maps derived from the encoder are increased by 8 times, resulting in the final feature maps with an output stride of 4.

Suppose that the input image $I \in N^{W \times H \times 3}$, and the output keypoint heatmaps are categoried into three groups $\hat{P}_c$, $\hat{P}_w$, and $\hat{P}_l$, where $\hat{P}_c$ refers to the heatmap predicting the center of a ship, $\hat{P}_w$ refers to the heatmap predicting the midpoint of the left and right side of a ship, and $\hat{P}_l$ refers to the heatmap predicting the head and tail of a ship. Each heatmap has the shape of $\frac{H}{S} \times \frac{W}{S}$ pixels, where S refers to the output stride ($S = 4$ in our network). Each pixel value in heatmaps represents the confidence of being a keypoint in that position.

For each ground truth map, we first resize it to the shape of $\frac{H}{S} \times \frac{W}{S}$ pixels for an accurate match with the network output. Then we partition the keypoints into three groups and compute the blurred heatmaps $P_{xy} \in [0, 1]$ with a Gaussion filter $K(x, y) = exp(-\frac{x^2+y^2}{2\sigma^2})$, where σ is the standard deviation adapted according to the object size (Law, Deng). The training objective of the keypoint extraction is the sum of three adapted focal loss (Lin et al., 2017b):

$$L_{kp} = \gamma_1 L_w + \gamma_2 L_l + L_c, \tag{1}$$

where L_c, L_w and L_l represent the keypoint extraction objectives of $\hat{P}_c$, $\hat{P}_w$, and $\hat{P}_l$, respectively. We set $\gamma_1 = \gamma_2 = 2$ to ensure the equivalent learning effect of these five keypoints. The definitions of L_c, L_w and L_l are the same as presented as follows,

$$\begin{aligned} L = -\frac{1}{N} \sum_{x,y} [&P_{xy}(1 - \hat{P}_{xy})^{\alpha} \log \hat{P}_{xy} \\ &+ (1 - P_{xy})^{\beta} (\hat{P}_{xy})^{\alpha} \log(1 - \hat{P}_{xy})], \end{aligned} \tag{2}$$

where α and β are hyper-parameters of the focal loss (we set $\alpha = 2$ and $\beta = 4$), and N is the number of ships per image.

3.3 Clustering Algorithm

Based on the derived heatmaps from the network, we first choose the keypoints whose confidences are greater than or equal to the 3×3 neighbors. Then we pick up the top 100 keypoints for each heatmap with the highest confidences from the selected keypoints. We use three sets S_c, S_w and S_l to represent the collection of the selected keypoints in the three heatmaps $\hat{P}_c$, $\hat{P}_w$, and $\hat{P}_l$, respectively. Each keypoint in the sets is represented by a triplet (x, y, f), where (x, y) refers to the keypoint coordinate in the image and f refers to the confidence. The coordinate (x, y) can be mapped to the input image through multiplying x and y by the output stride S. For the set S_c, we remove the keypoints whose confidences are under a specific threshold T ($T = 0.56$ in this paper, which is tuned through experiments). The rest keypoints in S_c are regarded as the center points of the ship candidates. Then we design Algorithm 1 *Side Keypoint Clustering Algorithm* to cluster the points in S_c and S_w into groups. In our experiments, we use $T = 0.3$ and $\delta = 8$ in the algorithms.

The triplets representing the center keypoint, the left-side keypoint and the right-side keypoint of a ship can be derived from our *Side Keypoint Clustering Algorithm*. The following step is the determination of the head keypoints and the tail keypoints from the set S_l. Correspondingly, we come up with Algorithm 2 *End Keypoint Clustering Algorithm* to cluster the keypoints in S_l into the ship instance candidates. In this algorithm, we replace the Euclidean distance with a new distance (Line 6 in the algorithm). The first part ensures the distance between the candidate point and the two keypoints on the sides as small as possible. The second part is used to maximally guarantee the candidate points located at the vertical bisector of the two points on the two sides. Considering the different contribution of these two parts, we introduce two weighted factors λ_1 and λ_2 to balance the contribution of these two parts ($\lambda_1 + \lambda_2 = 1$ for normalization). The determination of the values of λ_1 and λ_2 will be discussed through experiments in Section 4.3.

Algorithm 1 Side Keypoint Clustering Algorithm.

Require: the selected keypoint sets S_c and S_w
Require: candidates = []
Require: distance deviation δ
Require: confidence threshold T
1: remove the points in S_w whose confidences are under T
2: **for** each point $p_i^c \in S_c$ **do**
3: **for** each point $p_j^w \in S_w$ **do**
4: compute the distance $d_{ij} = \| p_j^w - p_i^c \|_2$
5: **end for**
6: rank the elements in S_w according to the distance d_{ij} in an increasing order
7: define the jth element in S_w as p_j^w
8: **for** $j = 2$ to $count(S_w)$ **do**
9: **if** $\| p_1^w + p_j^w - 2p_i^c \|_2 < \delta$ **then**
10: add a triplet (p_i^c, p_1^w, p_j^w) to candidates
11: break
12: **end if**
13: **end for**
14: **end for**

Algorithm 2 End Keypoint Clustering Algorithm.

Require: the candidates
Require: the selected keypoint sets S_l
Require: ships = []
Require: distance deviation δ
Require: confidence threshold T
1: remove the points in S_l whose confidences are under T
2: **for** each triplet $(p_i^c, p_i^{w1}, p_i^{w2})$ in the candidates **do**
3: **for** each point $p_j^l \in S_l$ **do**
4: compute the distance $d_{ij}^{w1} = \| p_j^l - p_i^{w1} \|_2$
5: compute the distance $d_{ij}^{w2} = \| p_j^l - p_i^{w2} \|_2$
6: compute the new distance $d_{ij}^{wl} = \lambda_1(d_{ij}^{w1} + d_{ij}^{w2}) + \lambda_2|d_{ij}^{w1} - d_{ij}^{w2}|$
7: **end for**
8: rank the elements in S_l according to the distance d_{ij}^{wl} in an increasing order
9: define the jth element in S_l as p_j^l
10: **for** $j = 2$ to $count(S_l)$ **do**
11: **if** $\| p_1^l + p_j^l - 2p_i^c \|_2 < \delta$ **then**
12: add a quintuplet $(p_i^c, p_i^{w1}, p_i^{w2}, p_1^l, p_j^l)$ to ships
13: break
14: **end if**
15: **end for**
16: **end for**

4. EXPERIMENTS

4.1 Datasets

We evaluate the performance of our method on two ship detection datasets, whose images are collected from the Google Earth. The scenes cover ports and sea surfaces. We name the two datasets as Dataset A and Dataset B for convenience. Dataset A contains 1010 images of 600×600 pixels and 900×900 pixels and 1124 ship instances. Dataset B contains 2938 images of 900×900 pixels and 5430 ship instances. The distributions of the length and the width of the ships on the two datasets are prensented in Figure 4. We note that Dataset A have a higher aspect ratio (the ratio of height and width) and a lower aspect ratio variance than Dataset B in average. We partition the two datasets into a training set and a test set by a ratio of $4:1$, respectively.

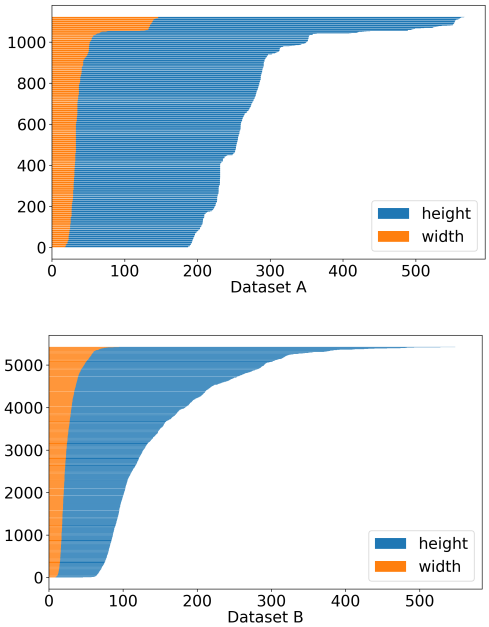


Figure 4. The distributions of the ship length and width on Dataset A and Dataset B, respectively. The horizontal axises represent the length and the height of the ships and the vertical axises represent the number of ship instances.

4.2 Experiments and Results

Our implementations are based on the PyTorch (Paszke et al., 2017) deep learning framework. The networks are trained on a computer equipped with a NVIDIA Geforce 1050Ti GPU.

In the training phase, we resize the images in the training set to 512×512 pixels and therefore the output heatmaps are 128×128 pixels. In order to alleviate overfitting, we add the data argumentation methods including the random horizontal flip and the color jitter. We use Adam (Kingma, Ba) as the optimizer for 70 epoches. The initial learning rate is $1.25e-4$, and decays by a half at the 30th and 45th epoch, respectively. During the test phase, we use the original images from the datasets without resizing. The Average Precision (*AP*) proposed in *Pascal VOC challenge* (Everingham et al., 2010) are adopted as the evaluation metric of our experiments.

We perform our method on the two ship detection datasets and derive the results in the forms of both the regular rectangular bounding boxes and the rotated bounding boxes. The quantitative results are presented in Table 1. The AP metrics in the form of the regular bounding box reach 90.9% on Dataset A and 89.7% on Dataset B respectively, outperforming the other state-of-the-art methods. The AP metrics in the form of the rotated bounding box reach 90.6% on Dataset A and 89.2% on Dataset B respectively. Our method outpforms RSCNN (Jiang et al., 2017) on Dataset A by a margin of 1.4% and performs competitively on Dataset B (a 0.1% gap). We note that our method has the least number of learnable parameters (281.6 MB) and computation cost (27.19 GFLOPs). In addition, it runs at 9.62 frames per second on a NVIDIA GeForce 1050Ti GPU, which is faster than the other methods.

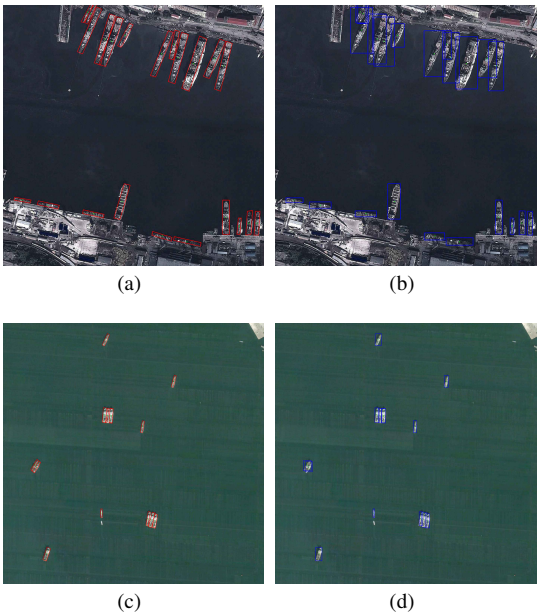


Figure 5. Qualitative ship detection results of our method in the forms of the rotated bounding boxes (a and c) and the regular rectangular bounding boxes (b and d). The scenes include ports (a and b) and sea surfaces (c and d). We note that the small ships are not taken into consideration on the two datasets.

The qualitative results are presented in Figure 5. It can be seen that the rotated bounding box can better represent a ship than the regular rectangular bounding box as the rotated bounding boxes can bound the ships tighter and contain less background. Specifically, when the ships are lying closely, the bounding boxes of the neighbouring ships are highly overlapped and one bounding box would contain the other ships nearby. More qualitative results in the form of the roated bounding box derived from our method can be seen in Figure 6.

4.3 Ablation Studies

Keypoint learning and heatmap assignment: We define the keypoints as the center of the ship, the tail, the head, the midpoint of the left side and the midpoint of the right side as presented in Figure 1. The tail and the head are classified as the same type of keypoints and therefore one heatmap is assigned to extract the two keypoints. In the same way, the midpoints of the left side and right side are classified as the same type and therefore they are assigned with the same heatmap. The center point of the ship is assigned with a separate heatmap. We make this decision out of the consideration of the symmetrical

Models	Parameters(MB)	FPS	GFLOPs	AP^A	AP^B
Regular Rectangular Bounding Box					
Two-stage detectors					
Faster R-CNN+FPN (Ren et al., 2015; Lin et al., 2017a)	330.2	3.56	34.51	89.8%	87.2%
One-stage detectors					
RetinaNet (Lin et al., 2017b)	290.0	3.88	28.84	81.8%	81.8%
ExtremeNet (Zhou et al., 2019)	794.8	1.33	83.45	78.3%	77.2%
ours	**281.6**	**9.35**	**27.19**	**90.9%**	**89.7%**
Rotated Bounding Box					
R2CNN (Jiang et al., 2017)	330.1	3.60	35.77	89.2%	**89.3%**
ours	**281.6**	**9.62**	**27.19**	**90.6%**	89.2%

Table 1. Numerical ship detection results on Dataset A and Dataset B. We derive the ship instances both in the forms of the regular rectangular bounding boxes and the rotated bounding boxes from the extracted keypoints with our method. We compare our method with some other methods in Average Precision (AP). FPS is short for *Frames Per Second* and the results are tested on a NVIDIA GeForce 1050Ti GPU.

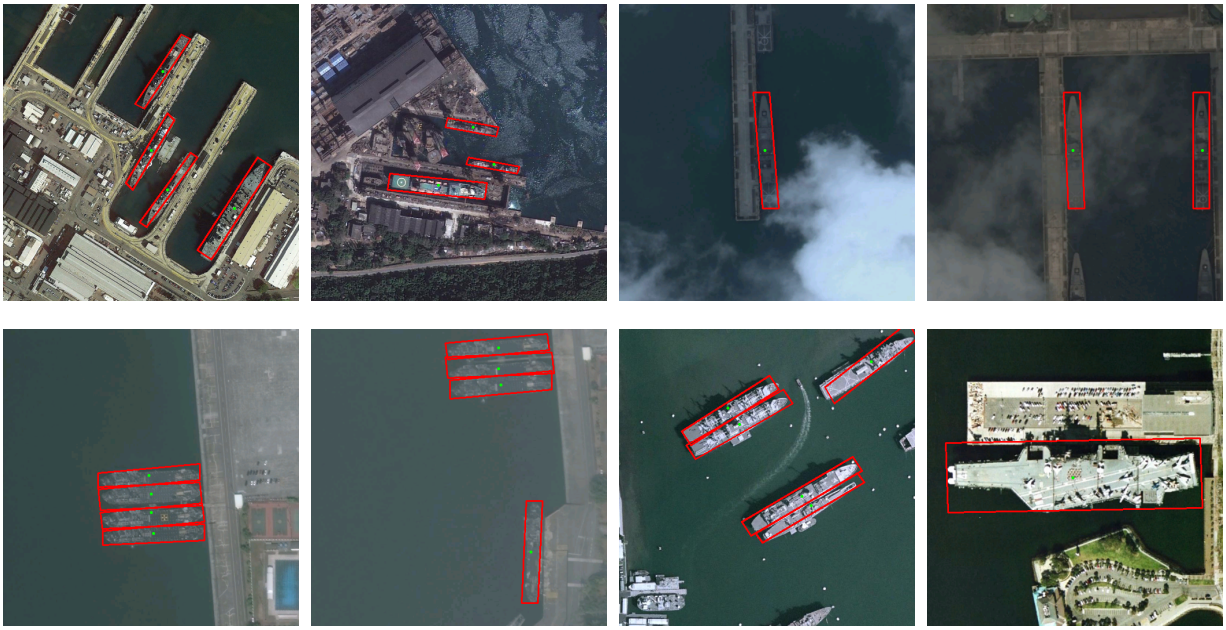


Figure 6. Examples of the ship detection results by our method in the form of the rotated bounding boxes. We note that the small ships are not taken into consideration on our two datasets.

appearance of the most ships, which may result in that it would be difficult to distinguish between the midpoints of the left side and the right side.

We demonstrate the advisability of our decision through experiments as presented in Figure 7 and Table 2. The definition of the keypoints and the assignment of the heatmaps of our method are presented in Figure 7(a). We use different colors to represent the assignments of the heatmaps. For simplicity, the center point is not visualized. The extracted keypoints of our method are presented in Figure 7(d). It can be found that the network can learn the keypoints correctly in this way. However, when the keypoints are assigned with separate heatmaps (as shown in Figure 7(b)), some failure cases occur as presented within the white circle area in Figure 7(e). The numerical results in Table 2 further demonstrate the reasonability of our decision. In Table 2, we use the MS COCO Average Precision (AP) and Average Recall (AR) as the evaluation metrics for the keypoint similarity measurement (Lin et al., 2014). It can be found that both the AP and the AR drop when assigning the keypoints with separate heatmaps.

Methods	*AP*	*AR*
keypoints defined in Figure 7(a)	**33.7%**	**40.8%**
keypoints defined in Figure 7(b)	31.2%	39.1%
keypoints defined in Figure 7(c)	17.7%	22.4%

Table 2. The performance of the keypoint extraction results on Dataset A using different keypoint defintions and keypoint learning methods.

Keypoint definition: For the definition of the keypoints, we argue that the keypoints should be located inside the ships. To demonstrate this point of view, we further conduct the experiments with the keypoints defined as the four corners of the bounding box plus the center point as presented in Figure 7(c). The corresponding keypoint extraction results are presented in Figure 7(f). It can be seen that some failure cases also occur. In addition, the numerical results in Table 4 (the method represented as c) further reveal that the keypoints located inside

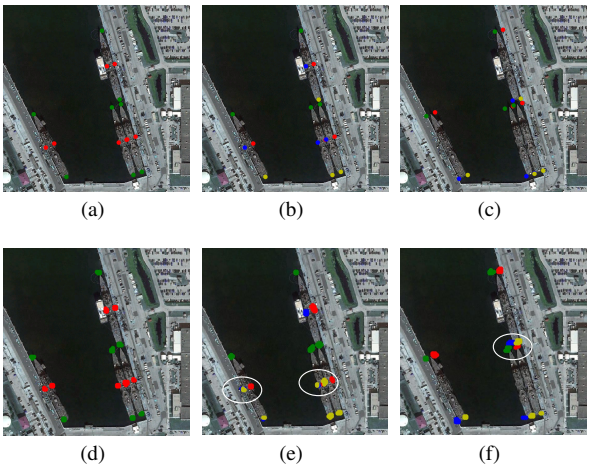


Figure 7. The results of different keypoint definitions and keypoint learning methods. The same color of the keypoints represents that the keypoints are assigned to the same heatmap for learning. For simplicity, the center points are not visualized.

objects can better represent a ship.

Clustering method: We further evaluate our clustering algorithm on the two datasets simultaneously. Based on our encoder-decoder network, we compare the performance of our clustering algorithm with the popular clustering algorithms, including the *Associative Embedding* (*AE*) (Newell et al., 2017) and the *Brute Force Enumeration* (*BFE*) (Zhou et al., 2019). The numerical results preseted in Table 3 verify the effectiveness and superiority of our method in the ship detection task.

Methods	$\mathbf{AP^A}$	$\mathbf{AP^B}$
AE	73.74%	68.28%
BFE	75.43%	70.81%
ours	**90.70%**	**89.72%**

Table 3. The performance of the clustering algorithms. *AE* is short for *Associative Embedding* and *BFE* is short for *Brute Force Enumeration* algorithm.

Hyperparameters λ_1 and λ_2: We define a new distance in the *End keypoint Clustering Algorithm* (Section 3.3). The new distance is divided into two parts, of which the first part is parameterized by λ_1 and the second part is parameterized by λ_2. The first part ensures that the candidate point should have a smallest distance to the two side keypoints (the midpoints of the left side and the right side). The second part ensures that the position of the candidate point should be close to the vertival bisector of the two side keypoints. We evaluate the importance of the two parts through extensive experiments on the two datasets by altering the ratio of λ_1 and λ_2. The experimental results are presented in Table 4.

The numerical results in Table 4 reveal that the performance of the clustering algorithm is not sensitive to the ratio of λ_1 and λ_2, which guarantees the robustness. Comparative results are derived in the range of [3,7] with little performance gap (less than 0.3% on both datasets).

The absence of any part of the distance will result in a poor performace. The absence of the second part ($\lambda_2 = 0$) results in a drop of the AP by about 9% on Dataset B and the absence of the first part ($\lambda_1 = 0$) results in a drop of the AP by about 9% on Dataset A and by about 27% on Dataset B.

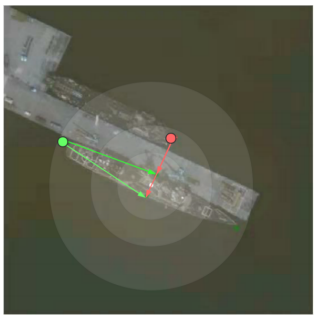


Figure 8. A special case that may fail to detect the ship instance represented by the green keypoints with our clustering algorithm if the ratio of λ_1 and λ_2 equals to 1.

λ_2/λ_1	AP^A	AP^B
0 ($\lambda_2 = 0$)	90.370%	80.686%
1	90.583%	80.642%
3	90.434%	89.714%
5	90.583%	**89.718%**
7	**90.700%**	89.689%
20	90.583%	80.852%
$+\infty$ ($\lambda_1 = 0$)	81.333%	62.209%

Table 4. The evaluation of the importance of the two parts of the new distance in the *End Keypoint Clustering Algorithm* on Dataset A and Dataset B through altering the ratio of λ_2 and λ_1.

It is noteworthy that the best results are not derived when the ratio of λ_2 and λ_1 equals to 1. We blame this on that the first part is much larger than the second part if the candidate keypoint is the tail keypoint or the head keypoint of the ship instance. Therefore, the importance of the second part is not fully strengthened if the ratio of λ_2 and λ_1 equals to 1. A special case is presented in Figure 8 where the distance of the red point is smaller than the green point if using the new distance ($\lambda_2/\lambda_2 = 1$) mentioned in the *End Keypoint Clustering Algorithm*. This can be alleviated through increasing the importance of the second part. Experimental results in Table 4 show that the best results are derived when the ratio of λ_2 and λ_1 is 5 or 7 and therefore it indirectly witness this point of view.

5. CONCLUSION

In this paper, we come up with a ship detection framework through extracting and clustering the ship keypoints in a diamond. An encoder-decoder network is used to extract the keypoints, followed by a clustering algorithm to cluster the keypoint into ship instances. The definition of the keypoints and the assignment of the heatmaps for the keypoint extraction are carefully considered to ease the network learning difficulty. We evaluate our method on two ship detection datasets using the popular Average Precision metric. Experimental results reveal that our method reaches the state-of-the-art results, outperforming other popular object detection algorithms. Further ablation studies are conducted to demonstrate the reasonability of our keypoint defition and the superiority of the clustering algorithm in the ship detection task.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2020-625-2020 | © Authors 2020. CC BY 4.0 License.

References

Cao, Z., Simon, T., Wei, S.-E., Sheikh, Y., 2017. Realtime multi-person 2d pose estimation using part affinity fields. *CVPR*.

Corbane, C., Najman, L., Pecoul, E., Demagistri, L., Petit, M., 2010. A complete processing chain for ship detection using optical satellite imagery. *International Journal of Remote Sensing*, 31(22), 5837–5854.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y., 2017. Deformable Convolutional Networks. http://arxiv.org/abs/1703.06211. cite arxiv:1703.06211.

Duan, K., Bai, S., Xie, L., Honggang, Q., Huang, Q., Tian, Q., 2019. CenterNet: Keypoint Triplets for Object Detection. *CoRR*, abs/1904.08189. http://arxiv.org/abs/1904.08189.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2), 303–338.

Feng, Y., Diao, W., Sun, X., Yan, M., Gao, X., 2019. Towards Automated Ship Detection and Category Recognition from High-Resolution Aerial Images. *Remote Sensing*, 11(16), 1901.

Fu, K., Chang, Z., Zhang, Y., Xu, G., Zhang, K., Sun, X., 2020. Rotation-aware and multi-scale convolutional neural network for object detection in remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 161, 294–308.

Girshick, R., 2015. Fast R-CNN. *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.

Guler, R. A., Neverova, N., Kokkinos, I., 2018. Densepose: Dense human pose estimation in the wild. *Conference on Computer Vision and Pattern Recognition (CVPR) 2018*.

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

Jiang, Y., Zhu, X., Wang, X., Yang, S., Li, W., Wang, H., Fu, P., Luo, Z., 2017. R2cnn: Rotational region cnn for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*.

Kingma, D. P., Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 1097–1105.

Law, H., Deng, J., 2018. Cornernet: Detecting objects as paired keypoints. *Proceedings of the European Conference on Computer Vision (ECCV)*, 734–750.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017a. Feature pyramid networks for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2117–2125.

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

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., 2014. Microsoft coco: Common objects in context. *European conference on computer vision*, Springer, 740–755.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A. C., 2016. SSD: Single shot multibox detector. *European conference on computer vision*, Springer, 21–37.

Newell, A., Huang, Z., Deng, J., 2017. Associative embedding: End-to-end learning for joint detection and grouping. *Advances in Neural Information Processing Systems*, 2277–2287.

Newell, A., Yang, K., Deng, J., 2016. Stacked hourglass networks for human pose estimation. *European Conference on Computer Vision*, Springer, 483–499.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in pytorch.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 91–99.

Wang, P., Sun, X., Diao, W., Fu, K., 2019. FMSSD: Feature-Merged Single-Shot Detection for Multiscale Objects in Large-Scale Remote Sensing Imagery. *IEEE Transactions on Geoscience and Remote Sensing*.

Yang, G., Li, B., Ji, S., Gao, F., Xu, Q., 2013. Ship detection from optical satellite images based on sea surface analysis. *IEEE Geoscience and Remote Sensing Letters*, 11(3), 641–645.

Yang, X., Sun, H., Fu, K., Yang, J., Sun, X., Yan, M., Guo, Z., 2018. Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks. *Remote Sensing*, 10(1), 132.

Zhang, R., Yao, J., Zhang, K., Feng, C., Zhang, J., 2016. S-CNN-Based ship detection from high-resolution remote sensing images. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 41.

Zhou, X., Zhuo, J., Krahenbuhl, P., 2019. Bottom-up object detection by grouping extreme and center points. 850–859.

Zhu, C., Zhou, H., Wang, R., Guo, J., 2010. A novel hierarchical method of ship detection from spaceborne optical image based on shape and texture features. *IEEE Transactions on geoscience and remote sensing*, 48(9), 3446–3456.

Zhu, Y., Zhao, C., Wang, J., Xu, Wu, Y., Lu, H., 2017. Couplenet: Coupling global structure with local parts for object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4126–4134.