This WACV workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

An Automated Method for the Creation of Oriented Bounding Boxes in Remote Sensing Ship Detection Datasets

Giorgos Savathrakis, Antonis Argyros
Institute of Computer Science, FORTH
and
Computer Science Department, University of Crete
gsav@ics.forth.gr, argyros@ics.forth.gr

Abstract

In a variety of maritime applications, the task of accurately detecting ships from remote sensing images is of significant importance. Various object detection algorithms localize objects by identifying either their Horizontal Bounding Boxes (HBBs) or their Oriented Bounding Boxes (OBBs). OBBs provide a far more accurate/tighter localization of object regions as well as their orientation. Several ship detection datasets provide annotations that include both HBBs and OBBs. However, many of them do not include OBB annotations. In this work, we propose a method which takes the ships' HBB annotations as input, and automatically calculates the corresponding OBBs. The proposed method consists of three main parts, (a) object segmentation that is built upon the Segment-Anything Model (SAM) to calculate object masks based on the information provided by the HBBs, (b) morphological filtering which eliminates possible artifacts stemming from the segmentation process, and (c) contour detection applied to the post-processed masks that are used to compute the optimal OBBs of the target objects. By automating the process of OBB annotation, the proposed method permits the exploitation of existing HBB-annotated datasets to train ship detectors of improved performance. We support this finding by reporting the results of several experiments that involve standard datasets, as well as state of the art object detectors.

1. Introduction

The detection of ships in remote sensing satellite images is a task that is both challenging and important for different types of applications. Existing datasets that focus on ship detection, provide annotations that include Horizontal Bounding Boxes (HBBs), which describe the horizontal rectangular region where objects lie, and Oriented Bounding Boxes (OBBs), which tightly enclose the object in the image and provide information about its center, width, height, and orientation. Because of their characteristics, OBBs are a better choice for the task of ship detection and localization. Moreover, in cases where ships are moored close to each other and their orientations is neither horizontal or vertical, HBB object detectors like YOLO [19], Faster-RCNN [20] or SSD [13], fail to correctly detect all ships within the image [15, 25].

1.1. Need for more training data

OBB detectors need to be trained on datasets where oriented bounding boxes ground truth is available. However, not all datasets provide OBB annotations (e.g., [11], [2]). To enrich the training potential of existing datasets and to come up with OBB ship detectors of improved performance, in this work, we propose a method which can automatically create OBBs for ship detection datasets, using only the provided HBB annotations as input. The proposed method is based on the Segment-Anything Model (SAM) [9], which is used to segment objects, given input prompts that can be obtained from the HBB ground truth. These segmented objects are in the form of masks that determine the image pixels where the objects are located. However, a large number of the generated masks exhibit artifacts such as object fragmentation in several connected components. This is resolved with morphological filtering, that connects possibly disconnected object regions, leading to more compact and accurate object masks [1]. In order to obtain OBBs that correspond to the segmented objects, we capitalize on the fact that the target OBB is the bounding box of minimum area. Therefore, we compute the boundaries of the post-processed object masks which are fed into a Minimum Area Rectangle calculation method which returns the target OBB.

The aforementioned process creates reliable OBBs for ship detection datasets, making exclusive use of the ob-

jects' HBB annotations. This is validated through extensive experiments on two benchmark datasets for ship detection, HRSC2016 [16] and ShipRSImageNet [26], where we make use of the HBBs of their training sets to calculate OBBs with the described method. Then, using these OBBs we train a wide range of OBB object detectors (i.e., [4, 6, 7, 12, 22, 24]), and compare their performance on the test set to that which would have been obtained if the ground truth OBBs had been used.

1.2. Need for removing dataset biases

In addition to creating more training data automatically, the proposed automated OBB annotation can lead to the reduction of spatial, scale or orientation imbalance in an existing dataset. Specifically, orientation or aspect ratio bias among the objects in the dataset, may lead to less accurate detections if they are used for training. Data augmentation methods have shown that they can lead to increased performance both in image classification [10, 17, 21] and object detection [27, 30] tasks. When it comes to object detection, geometric transformations (i.e. rotation, scaling) are the most effective ways to resolve these imbalances, since if they are applied correctly to both the images and the annotations, more orientations and sizes will be visible to an object detector during training. This can be done by rotating existing samples in order to cover the orientation space, and thus balance the number of samples per orientation. In our work, we demonstrate that data augmentation performed based on the proposed method leads to improved performance of existing OBB object detectors.

In summary, the main contributions of this paper are the following:

- We provide an automated method which creates OBBs in ship detection datasets, built upon the SAM model and morphological filtering operations, using only the HBB ground truth as prior information.
- We develop a data augmentation technique, using the generated OBB annotations, to reduce orientation imbalance. We prove that by using augmentation we can increase the baseline performance of existing object detectors in ship detection datasets.

2. Related Work

2.1. Ship Detection

Ship detection in satellite images has been trending in the research community for a long time. The first works in that field mainly focused on separating ships from the background by implementing shape and texture analysis. For instance, Zhu *et al*. [29] propose a method which separates possible candidate regions from the sea using edge filters, and then removing false detections using spatial object features like the length-to-width ratio which is considered to be different for ships compared to other types of objects. Yang *et al*. [23] focus instead on a sea surface analysis, which makes use of image intensity frequencies to segment-out abnormalities in sea regions. By using a ratio of background pixels to random noise pixels, the regions with candidate ships are detected. Proia *et al*. [18] assume a Gaussian distribution that describes the sea regions, and use a Bayesian decision method to identify the regions corresponding to ships.

All these methods have a common limitation which is the need for the manual selection of thresholds, which is also highly dataset-dependent. Also, these methods can be used mostly in datasets where the ships are in open sea where the texture differences between ships and background, including clouds, is more apparent.

Modern ship detection methods make use of Deep Learning algorithms that are based on Convolutional Neural Networks (CNNs), to extract image features. One of the properties that discriminates object detectors is their prediction outputs, which can be either HBBs or OBBs. Another difference is the number of stages used for the object detection process. Two-stage object detectors firstly generate candidate object locations, and then the model evaluates whether these locations contain an object and if they do it determines the class in which it belongs. One-stage object detectors on the other hand, do the entire process in one go, using anchor boxes that slide over the entire image, and for each location a class probability and bounding box estimations are calculated. Table 1 provides a list of popular object detectors, specifying their architecture and the type of the bounding box they deliver as an output.

Method	Number of stages	Box type
YOLO [19]	one-stage	HBB
SSD [13]	one-stage	HBB
Fast-RCNN [5]	two-stage	HBB
Faster-RCNN [20]	two-stage	HBB
R^3Det [24]	one-stage	OBB
Rotated-RetinaNet [12]	one-stage	OBB
ReDet [7]	two-stage	OBB
Oriented-RCNN [22]	two-stage	OBB
RoI-Transformer [4]	two-stage	OBB
S^2A-Net [6]	two-stage	OBB

Table 1. List of popular object detectors, with their number of stages and bounding box type.

Liu *et al*. [14], built on the basis of the Fast-RCNN object detector [5] and proposed a Rotated Region CNN (RRCNN) consisting of a Rotated Region of Interest pooling layer (RRoI) and a Rotated Bounding Box regression model (OBB), to allow the detector to extract features from rotated

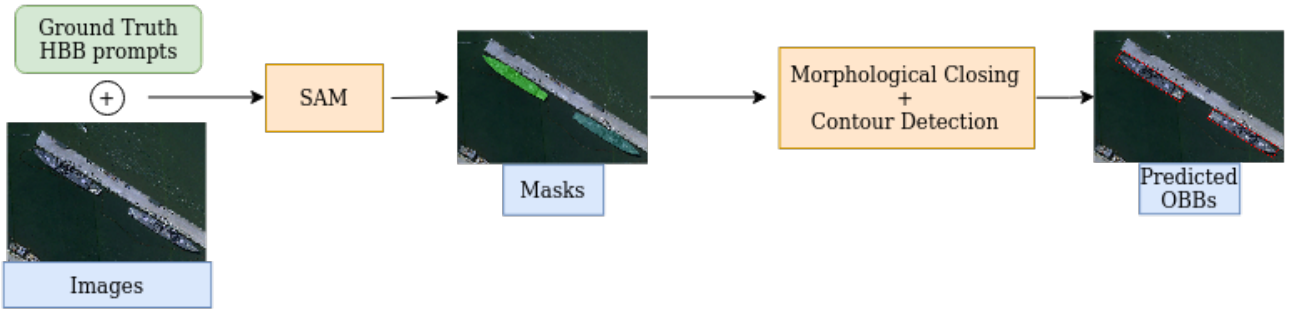


Figure 1. Overview of our proposed method. The images, together with their HBB annotations, are fed as prompts to the SAM model, and the calculated object masks pass through a morphological closing operation and a contour detection function that finally provide the corresponding OBBs.

regions and predict the OBBs more accurately. Zhang *et al*. [25], builds upon the Faster-RCNN object detector [20] and proposed a Rotated Region Proposal Network (R^2PN), which expands the degrees of freedom of the anchor generation mechanism in order to include not only different sizes and aspect ratios but also different orientations, thus enabling it to generate arbitrary-oriented region proposals. Work that was done on improving single-stage oriented object detectors by Yang *et al*. [24], focused on fixing the feature misalignment problem by proposing R^3Det, which implements a feature refinement module to match refined coordinates with the respective feature points. Other oriented object detectors like ReDet [7], RoI Transformer [4] and Oriented-RCNN [22], are two-stage and focus on improving the alignment of the RRoIs, mostly through extraction of rotation invariant features, and on the generation of more accurate proposals with better proposal representations.

2.2. Data augmentation

It has been shown that the strategy of tweaking the data in given datasets in order to widen the range of the current data properties (e.g. illumination, scale, rotation), can lead to performance improvements in several computer vision tasks. Krizhevsky *et al*. [10] and Simonyan *et al*. [21] show that performing data augmentation using translation and intensity changes leads to reduced image classification errors. Zoph *et al*. [30], show that in object detection problems, a combination of color and geometric transformations which, in turn, also affects the Bounding Boxes, leads to improvements of the detection performance of several object detectors in different benchmark datasets. A different approach for data augmentation is proposed by Zhong *et al*. [27], who implement random removal of patches within either the entire image or within object regions. The result of this strategy was an increase of accuracy in both image classification and object detection tasks.

3. Method

The proposed method (see also Figure 1) can be summarized as follows. The input images, as well as their annotations, are given as input to the SAM predictor. The output is in the form of masks that specify the segmented object regions and are used to calculate the OBBs of each object, via morphological filtering and contour detection. The final masks are then passed through the ground truth overlap calculator, which calculates the IoU between the HBBs of the predicted masks and those of the respective ground truth objects. We manually define a threshold for the IoU, above which we consider the detection as valid. In this case, the OBBs of the accepted objects are then represented as the 2D coordinates of their four corners and constitute the final output of the model.

3.1. SAM

The Segment-Anything Model (SAM) [9], is a novel image segmentation method which enables a zero-shot generalization, namely it can segment objects even in images that are in a significantly different domain than the one on which the model was trained. The implementation is based on an encoder-decoder architecture. Two different encoders are implemented, each taking a different type of input. The first takes the image as input and maps it into an embedding space. The second takes prompts as input, which can be dense (mask) or sparse (points, boxes). The purpose of these prompts is to determine the areas where the segmentation has to focus on. The outputs from both encoders are then passed through a mask decoder which returns different sets of segmentation masks that define the segmented regions of an image, each with a confidence score.

The most important factor that determines the quality of the detection, is the accuracy with which we determine the region corresponding to an object. To achieve this, we make use of the SAM predictor and we apply it to all images and all objects. SAM can segment objects in a specified region

Figure 2. Segmentation results for an image with two objects. Green and red stars correspond to foreground and background points respectively. The green boxes that surround the objects, are their HBBs. In the top figure, foreground points lie on the diagonal directed from bottom left to top right, and vice versa in the bottom figure. Each setting yields a different score for the segmentation of each object.

given a set of m 2D points which are specified to belong to the image foreground or background.

The main assumption upon which our implementation is based, is that the objects at hand are elongated, meaning that their length is considerably larger than their width. Since the objects we are interested in are ships, we expect that this is a valid assumption. An important feature that elongated objects present however, is that their orientation will lie upon one of the two diagonals of the surrounding HBB. Due to the fact that we don't have any prior knowledge regarding which diagonal that is, we adopt the following strategy. We take the center of the object's HBB as a foreground point because we are sure that it belongs to the object. The remaining $m-1$ points are equally distributed across the two diagonals so that each diagonal has $(m-1)/2$ points, symmetrically placed with accordance to the center point. Then, we run the segmentation for two cases as shown in Figure 2. In the first case, the input points of one diagonal are treated as foreground points and the rest as background, and in the second case the opposite. Both segmentations yield a score that quantifies SAM's confidence about the accuracy of the segmentation and we assume that the one with the highest score has correctly segmented the object at hand. The output of the segmentation for one image, is a set of binary masks, each corresponding to one object. Each mask has the same size as the input image, with 1s in the pixels that belong to the object and 0s otherwise.

3.2. Morphological filtering

The segmentation masks specify which pixels belong to a certain object. However, a problem that arises is that the pixels belonging to the mask do not necessarily form a single connected component. In several cases there are gaps in the interior and the border regions of the mask. This is resolved through a morphological closing operation:

$$A \bullet B = (A \oplus B) \ominus B. \tag{1}$$

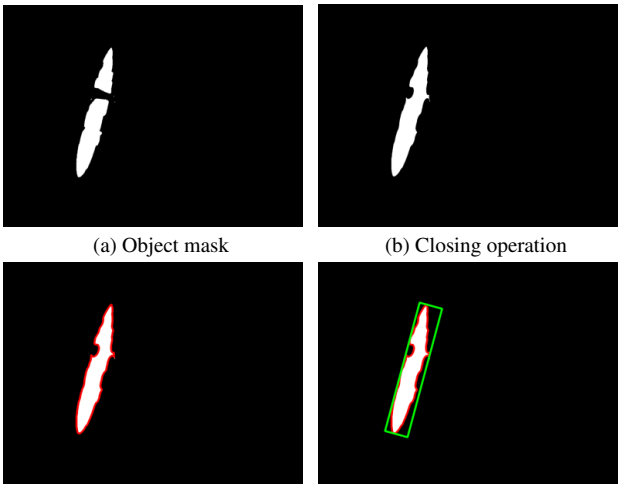
In Equation (1), A is the binary mask resulting from SAM and B is the structuring element used for morphological filtering. Symbols $\oplus$ and $\ominus$ denote dilation and erosion operations, respectively.

Essentially, the closing operation firstly expands the mask according to the structuring element, which leads to the filling of any gaps within the mask, and then reduces it by the same element, which removes the expanded regions at the edges of the object area. We select an ellipse-shaped structuring element B whose size is adaptable to the length of the object at hand. The ellipse however, due to its eccentricity, has to be directed in the same angle as the object. To achieve that we find the diagonal of the HBB upon which the object lies, via the SAM segmentation score, and then we calculate the inverse tangent of the angle formed between the object's direction and the horizontal x-axis. This gives the orientation the ellipse should have, and we also specify the minor and major axes.

3.3. Contour detection

After the closing operation, the resulting masks are used for the calculation of the object region boundaries. This is done by implementing a contour detection method which initially detects changes in the intensity of the binary image, and then after implementing a connected component analysis, it returns one or more sets of points, that enclose the regions containing the object pixels. The reason why there may be more than one contours, is that even after the closing operation, it is possible that some isolated blobs, signified as object regions, are not included within the main object region. To resolve this, we assume that the contour corresponding to the object region, is the one encompassing the largest area in terms of pixel numbers.

After computing the contour surrounding the object mask we obtain the optimal OBB surrounding it, by calculating the minimum bounding rectangle that encompasses



(a) Object mask (b) Closing operation

(c) Contour calculation (d) OBB calculation

Figure 3. Process of obtaining an object's OBB. (a) The initial object's mask. The gap between its two components is apparent. (b) Shows the mask after the closing operation which unites the two components. (c) Calculation of the contour surrounding the mask (red). (d) Calculation of the OBB (green).

the contour. This is done via a Principal Component Analysis (PCA) that is run on the points belonging to the object's contour. The resulting eigenvectors that correspond to the two largest eigenvalues are the ones with the largest variance and in our case they define the length and width of the object mask. The center of the OBB is obtained from the centroid of the contour points, and the angle of the object's orientation is obtained from the orientation of the eigenvector which corresponds to the axis with the largest variance, namely the one related to the object's length. The center, width, height and orientation of the object are all the parameters needed to define an OBB. An example of the mentioned steps is shown in Figure 3.

A way to prove that the segmentation has been done correctly, is to compare the ground truth HBB of each object with the HBB that would be computed, given the segmentation mask of the respective object. The predicted HBB of an object is computed in the following way. The edge points of an object's segmentation mask are considered to be the same as the ones that would define the HBB surrounding it. We then calculate the IoU between the ground truth and predicted HBB of each object, and define a hyperparameter that corresponds to the IoU threshold above which we consider the segmentation as correct. The masks of the correctly segmented objects, will subsequently be used for the calculation of their respective OBBs.

3.4. Dataset Augmentation

Ship datasets contain images of ships in various orientations. However, there might be significant imbalance in the number of ship samples per orientation. Given the proposed method, we can correct such imbalances. Specifically, by having access to the OBBs of the ships, we can augment the given imbalanced dataset so as to cover the space of ship orientations more uniformly. This can be done in several ways; in this work we explore two such techniques.

Same Size Object-wise (SSO) augmentation: We create an augmented dataset, with the same number of samples as the original dataset but with images rotated in such a way so that ships are as equally distributed among orientations as possible. To do that, we firstly create an empty histogram, hereby referred to as SSO-Histogram, with as many bins as the number of quantized ship orientations. Then, we sort the images according to the number of included objects, in descending order. Starting with a random image from those with the highest number of objects, we calculate their orientation histogram and add it to the SSO-Histogram. The remainder of the process is the following. We iteratively select random images, calculate the orientation histogram of the included objects, and then rotate the image in such a way so that the newly updated SSO-Histogram has the minimum variance. Each time an image is selected, it gets removed from the selection pool, so that the random selection begins with images with the most objects and gradually with images with fewer ones. The iterations terminate when the total number of objects in the SSO-Histogram reaches the total number of objects in the dataset.

Increased Size Object-wise (ISO) augmentation: The second way for performing augmentation is to increase the number of objects in under-represented orientations so that each orientation has twice as many samples as the most prevalent orientation in the pre-augmented dataset. To do this we use the dataset's orientation distribution with the same quantization as before, and we assign it to an ISO-Histogram. We select the orientation with the most objects and define the upper bound of all the orientation bins as the double of that number of objects. If for example, the dataset has 80 objects pointing at 70 degrees, the upper bound for all bins is set to 160. The next step is to constantly select random images, and rotate them in such a way that at least one of the objects is brought to the most prevalent orientation, while simultaneously adding the orientation distribution of the rotated image, to the ISO-Histogram. This is done until the number of objects in that orientation, reaches the upper limit. Then, we iteratively select random images from the dataset, we calculate their objects' orientation distribution, and rotate the image by the angle at which the updated ISO-Histogram will have the minimum variance. This is done while constantly checking if at any orientation the number of objects exceeds the upper bound, and if it does we select another image. The iterations terminate when the total number of objects in the ISO-Histogram reaches the

upper bound per orientation times the number of orientation bins.

The calculation of the rotated OBB coordinates is done by applying the rotation matrix on the OBB center, and because the image orientation increases the image size, the center points are adjusted to the new image size, as shown in Equations 2, 3,

$$c'_x = cos\theta \cdot \left(c_x - \frac{w}{2}\right) - sin\theta \cdot \left(\frac{h}{2} - c_y\right) + \frac{w'}{2} \quad (2)$$

$$c'_y = -\left(sin\theta \cdot \left(c_x - \frac{w}{2}\right) + cos\theta \cdot \left(\frac{h}{2} - c_y\right) - \frac{h'}{2}\right), \quad (3)$$

where θ is the difference between the object's orientation and the orientation to which we want to bring it, w and h are the width and height of the original image, and w' and h' are the width and height of the rotated image, respectively. The height and width of the OBB remain the same since the number of pixels corresponding to each of the object's sides, is rotation invariant and also invariant with respect to the image size.

4. Experiments

4.1. Datasets

In order to validate the effectiveness of the developed method, we run experiments on two benchmark datasets for ship detection, HRSC2016 [16] and ShipRSImageNet [26]. Both are remote sensing ship detection datasets, that consist of high resolution images containing ships of several types. They provide extensive annotation files that include both HBBs and OBBs for the image objects, and implement a 4-level hierarchical classification scheme regarding the object type. The first level simply describes whether an object is a ship or not, and the remaining levels classify the ships in ever increasing category detail.

HRSC2016 is split into training, validation and test sets, that consist of 436, 181 and 453 images, respectively. The HBBs are in the form of (x_{min}, y_{min}, x_{max}, y_{max}) and the OBBs in the form of (c_x, c_y, w, h, θ). The object orientation θ ranges between $-\pi/2$ to $\pi/2$. It is important to note that for this dataset, annotation files are provided for all sets including the test set.

ShipRSImageNet is also split accordingly but there are 2198 images in the training set, 550 in the validation set, and 687 in the test set. The HBBs are again in the form of (x_{min}, y_{min}, x_{max}, y_{max}), but the OBBs are provided in two forms. The first is (c_x, c_y, w, h, θ) where θ is the angle between the object's orientation and the horizontal axis. The second is (x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4), which contains the coordinates of the bounding box edge points, and is the form we use for the experiments with this dataset. The bounding box annotations in this case, are provided only for the training and validation sets, and the first level object classification includes the "ship" and "dock" classes.

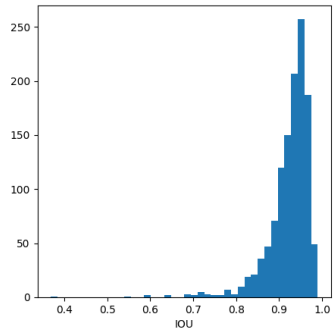


Figure 4. Histogram of objects' IoU, between predicted and ground truth HBBs, in the HRSC2016 dataset [16]

4.2. Implementation details and experimental setup

In order to ensure the fairness of the provided results, we discard the OBB ground truth from the training sets of both datasets, keeping only the HBBs as input to the model. The expected form of the HBBs is (x_{min}, y_{min}, x_{max}, y_{max}).

During the segmentation process (see section 3.1), we use $m = 5$ points. One of them is the center of the HBB, and the others are placed halfway between the center of the box and the respective box edge. For instance, the points belonging to the 1st diagonal are the center, the one in the middle between the top left edge and the center, and the one in the middle between the bottom right edge and the center. Similarly, for the 2nd diagonal, the points are the center, the middle between the top right edge and the center, and the middle between the bottom left edge and the center. Additionally, the respective objects' HBBs are also used as input prompts to SAM.

For the closing operation, we use an ellipse shaped structuring element with minor axis equal to $\sqrt{2}$ times 3% that of the HBB diagonal. Regarding the true and predicted HBB overlap calculation, we set the IoU threshold to be 0.7 in order to consider it as a valid segmentation. Objects with invalid segmentation are not considered parts of the newly generated dataset. After the calculation of the OBBs, we create new annotation files that include the HBBs and OBBs of the objects whose segmentation from SAM was deemed valid. For the augmentation part, we set the discretization of the orientations to be equal to an angle range of 10 degrees for all three methods.

The most decisive factor which determines the performance of the proposed method, is its ability to generate OBB annotations that are accurately aligned with the ac-

tual objects in the images. To validate this, we make use of several oriented object detectors by training them using the generated annotations from our method, and testing their performance on the test sets that were left intact during the entire process. However, a fair evaluation should take into consideration that any object detector would itself have a certain performance on the original datasets. Therefore, in order to estimate the relative effectiveness of our proposed method, we also need to train the object detectors that are to be used, on the training sets of the original datasets, and then compare their respective performances on the test sets. The oriented object detectors that we use, are obtained from the OpenMMLab project [28], and they are all implemented with ResNet-50 [8] backbone, pretrained on ImageNet [3]. The optimizer used is SGD with momentum, where the momentum is 0.9 and the weight decay is 1e-4. These object detectors are either single-stage (e.g. R^3Det [24]), or two-stage (e.g. ReDet [7], Oriented-RCNN [22]). The learning rate for the ReDet detector is 0.01, and 0.0025 for all the other detectors used. We used one NVIDIA GeForce GTX 1080 Ti GPU with 11GB RAM.

4.3. Evaluation metrics

The evaluation metric used for the detection performance is the mean average precision (mAP) as it is used in PASCAL 2007, and the training lasts for 36 epochs, since it has been found that for most object detection tasks, it is a reasonable limit before convergence is reached. In the case of the HRSC2016 dataset, we make use of the training set only and not the validation set. Therefore, the results obtained from the training of selected object detectors will not be the same as the ones presented in the original papers (i.e. [4,6,7,12,22,24]), but since the purpose of this work is to provide a comparative assessment between original and generated OBBs, our main focus is not on the performance metric values of the detection, but on the relative differences between the detection performances yielded from training on the different sets.

4.4. Parameter setting and ablation studies

For the implementation of the proposed method, it was important to carefully tune three main parameters. These were, the number of points given as prompts to SAM, the shape and size of the structuring element used for the closing operation, and the IoU threshold between predicted and ground truth HBBs. The prompt points were set to 5, which was the minimum possible, because qualitative assessments of the created segmentation masks, indicated that more points lead to segmentation of the background, which is more uniform, and therefore yielding higher segmentation scores than the object. For the same reason, it is very important to also include the HBB annotations as prompt inputs, since failure to do so, results in even more background segmentations, especially in cases of ships in open seas.

Type	Size	IoU threshold			
		90%	80%	70%	60%
circle	$\sqrt{2}$*3%	76.22	96.52	98.67	99.50
	$\sqrt{2}$*6%	75.97	96.93	99.25	99.83
	$\sqrt{2}$*9%	74.81	96.77	99.34	**99.92**
	$\sqrt{2}$*12%	73.82	96.35	99.34	99.83
	$\sqrt{2}$*15%	73.65	96.27	99.34	99.83
ellipse	$\sqrt{2}$*3%	**78.62**	**97.27**	99.17	99.67
	$\sqrt{2}$*6%	77.30	97.02	**99.42**	99.75
	$\sqrt{2}$*9%	75.14	96.52	99.25	99.59
	$\sqrt{2}$*12%	72.49	95.44	99.01	99.67
	$\sqrt{2}$*15%	69.76	94.12	98.34	99.42

Table 2. Different structuring element settings and percentage of objects from the HRSC2016 dataset that exceed certain IoU thresholds. The types of the element are either circular or elliptical and the sizes correspond to the diameter and minor axis respectively. Their values are in the form of object length percentage.

For identifying the best configuration of the structuring element, we investigated which setting provided the maximum percentage of objects of the original HRSC2016 dataset for each IoU threshold between predicted and ground truth HBBs. The results in Table 2 indicate that an elliptic structuring element with minor axis equal to $\sqrt{2}$ times 3% the length of the object, meets this requirement. The major axis is set equal to twice the minor axis.

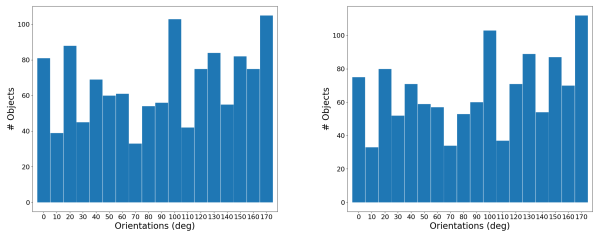
Finally, we set the IoU threshold to 70%. The reason for this comes from information obtained from Figure 4, where we see that the vast majority of the segmented objects have an HBB IoU overlap that exceeds the value of 0.7. Specifically, more than 99% of the total objects in the original training set, satisfy that IoU threshold.

4.5. Experimental results

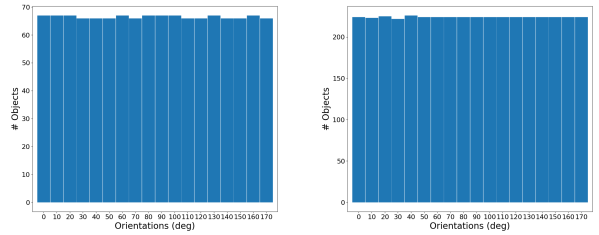
In order to provide a visible comparison between the ground truth data, and the data generated from our method, we train all detectors using the ground truth OBBs, then using the generated OBBs, all from the training sets, and validate the performance on the test set which is the same for both the ground truth and generated data.

4.5.1 Results on the HRSC2016 dataset

In Table 3 we present the performance of our annotation generation and augmentation methods, for a number of object detectors using ground truth and generated data for the HRSC2016 dataset. We can see that the object detectors' performance on the test set, when trained using the OBBs generated from our method, is for the most part slightly lower but close to that obtained by training them using the ground truth annotations. This means that the generated



(a) ground truth data (b) generated data



(c) SSO data (d) ISO data

Figure 5. Orientation distribution of the objects' OBBs, for HRSC2016 in the: (a) ground truth data, (b) generated annotations, (c) SSO augmentation dataset and, (d) ISO augmentation dataset.

Method	GT	Gen	SSO	ISO
	mAP	mAP	mAP	mAP
R^3Det [24]	83.06	79.49	79.18	**88.33**
ReDet [7]	79.73	78.96	77.78	**87.71**
Oriented-RCNN [22]	**89.93**	81.13	88.91	89.77
RoI-Transformer [4]	87.07	86.92	76.38	**87.86**
Rotated-RetinaNet [12]	62.26	63.01	60.42	**78.43**
S^2A-Net [6]	**89.61**	80.59	86.68	89.23

Table 3. mAP scores for object detectors, trained on HRSC2016, using ground truth annotations (GT), generated annotations (Gen), and augmented data with the proposed methods (SSO, ISO).

OBBs are, at least in their vast majority, of similar quality to the ones from the ground truth. Figure 5 shows the distribution of the objects' orientations in the ground truth (5a) and in the generated (5b) samples. It can be verified that the distribution is almost identical. This means that the proposed method, not only creates data that can lead to similar detection performances as those from the original dataset, but also the newly formed dataset maintains the same spatial properties as the original. This is important because if the method had managed to capture a subset of the original dataset where the orientations of the included objects were different than those of the original, then the new dataset would not be consistent with the original.

Considering the promising results of the annotation method, we use the generated annotations for the creation of augmented datasets. For both augmentation methods used (SSO, ISO), the resulting orientation histograms are completely uniform within the specified orientation quantization, which means that the resulting datasets have nearly zero orientation bias, as can be verified from figures 5c and 5d. The performance of the same object detectors using the augmented datasets, indicates that the most effective augmentation method is the ISO. SSO yields results that are similar to the ones obtained from ground truth training, but it does not exceed them for any of the object detectors used. Apart from the Oriented-RCNN [22] and S^2A-Net [6] object detectors, ISO yields improved detection performances ranging from 5 to 16% increases in the mAP metric.

Method	GT	Gen	SSO	ISO
	mAP	mAP	mAP	mAP
R^3Det [24]	42.61	36.89	33.99	**44.37**
ReDet [7]	**59.31**	46.75	41.09	52.79
Oriented-RCNN [22]	57.43	46.80	49.71	**58.93**
RoI-Transformer [4]	**44.91**	36.09	34.18	43.19
Rotated-RetinaNet [12]	37.61	29.85	27.42	**37.82**
S^2A-Net [6]	**55.61**	39.79	36.95	51.25

Table 4. mAP scores for object detectors, trained on ShipRSImageNet, using ground truth annotations (GT), generated annotations (Gen) and augmented data with the proposed methods (SSO, ISO).

4.5.2 Results of the ShipRSImageNet dataset

The detection performances obtained for the ShipRSImageNet dataset, are shown in Table 4. We can see that in this dataset, three out of the six detectors used, yielded better performance when trained with the ISO augmented data compared to the one obtained using ground truth data. Specifically, the improvements range from $\sim$0.2% to $\sim$2%, which are not as significant as those in the HRSC2016 dataset, but still exhibit that even in more complex datasets, our method can lead to improved detection performances.

5. Conclusions

We proposed a new method to automatically annotate objects in ship detection datasets with OBBs, and demonstrated its effectiveness by showing that the performance of object detectors when trained on the data with the generated annotations is similar to that obtained when using the ground truth annotations for training. We also proposed two data augmentation techniques aiming to make the dataset more balanced in terms of orientations which are, object-wise with same size and object-wise with increased size. The second method, managed to improve the performance achieved by benchmark object detectors in the test set, a fact that proves the importance of the proposed methods in ship detection tasks.

References

[1] Shao-Yi Chien, Shyh-Yih Ma, and Liang-Gee Chen. Efficient moving object segmentation algorithm using background registration technique. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(7):577–586, 2002. 1

[2] Aaron Walter Avila Cordova, William Condori Quispe, Remy Jorge Cuba Inca, Wilder Nina Choquehuayta, and Eveling Castro Gutierrez. New approaches and tools for ship detection in optical satellite imagery. *Journal of Physics: Conference Series*, 1642(1):012003, sep 2020. 1

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 7

[4] Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning roi transformer for oriented object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2849–2858, 2019. 2, 3, 7, 8

[5] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[6] Jiaming Han, Jian Ding, Jie Li, and Gui-Song Xia. Align deep features for oriented object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2021. 2, 7, 8

[7] Jiaming Han, Jian Ding, Nan Xue, and Gui-Song Xia. Redet: A rotation-equivariant detector for aerial object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2786–2795, 2021. 2, 3, 7, 8

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 7

[9] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, October 2023. 1, 3

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 2, 3

[11] Darius Lam, Richard Kuzma, Kevin McGee, Samuel Dooley, Michael Laielli, Matthew Klaric, Yaroslav Bulatov, and Brendan McCord. xview: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856*, 2018. 1

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2017. 2, 7, 8

[13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing. 1, 2

[14] Zikun Liu, Jingao Hu, Lubin Weng, and Yiping Yang. Rotated region based cnn for ship detection. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 900–904. IEEE, 2017. 2

[15] Zikun Liu, Hongzhen Wang, Lubin Weng, and Yiping Yang. Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds. *IEEE Geoscience and Remote Sensing Letters*, 13(8):1074–1078, 2016. 1

[16] Zikun Liu, Liu Yuan, Lubin Weng, and Yiping Yang. A high resolution optical satellite image dataset for ship recognition and some new baselines. In *International conference on pattern recognition applications and methods*, volume 2, pages 324–331. SciTePress, 2017. 2, 6

[17] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, pages 117–122, 2018. 2

[18] Nadia Proia and Vincent Pagé. Characterization of a bayesian ship detection method in optical satellite images. *IEEE geoscience and remote sensing letters*, 7(2):226–230, 2009. 2

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society. 1, 2

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 1, 2, 3

[21] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. pages 1–14. Computational and Biological Learning Society, 2015. 2, 3

[22] Xingxing Xie, Gong Cheng, Jiabao Wang, Xiwen Yao, and Junwei Han. Oriented r-cnn for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3520–3529, October 2021. 2, 3, 7, 8

[23] Guang Yang, Bo Li, Shufan Ji, Feng Gao, and Qizhi Xu. Ship detection from optical satellite images based on sea surface analysis. *IEEE Geoscience and Remote Sensing Letters*, 11(3):641–645, 2013. 2

[24] Xue Yang, Junchi Yan, Ziming Feng, and Tao He. R3det: Refined single-stage detector with feature refinement for rotating object. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3163–3171, 2021. 2, 3, 7, 8

[25] Zenghui Zhang, Weiwei Guo, Shengnan Zhu, and Wenxian Yu. Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks. *IEEE Geoscience and Remote Sensing Letters*, 15(11):1745–1749, 2018. 1, 3

[26] Zhengning Zhang, Lin Zhang, Yue Wang, Pengming Feng, and Ran He. Shiprsimagenet: A large-scale fine-grained dataset for ship detection in high-resolution optical remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:8458–8472, 2021. 2, 6

[27] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020. 2, 3

[28] Yue Zhou, Xue Yang, Gefan Zhang, Jiabao Wang, Yanyi Liu, Liping Hou, Xue Jiang, Xingzhao Liu, Junchi Yan, Chengqi Lyu, Wenwei Zhang, and Kai Chen. Mmrotate: A rotated object detection benchmark using pytorch. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 7331–7334, New York, NY, USA, 2022. Association for Computing Machinery. 7

[29] Changren Zhu, Hui Zhou, Runsheng Wang, and Jun Guo. 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, 2010. 2

[30] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 566–583. Springer, 2020. 2, 3