IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 28, 2019, accepted July 3, 2019, date of publication July 16, 2019, date of current version August 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2929080

Precise Ship Location With CNN Filter Selection From Optical Aerial Images

SAMER ALASHHAB[1], ANTONIO-JAVIER GALLEGO[1,2], ANTONIO PERTUSA[1,2], AND PABLO GIL[1,3], (Senior Member, IEEE)

[1]Computer Science Research Institute, University of Alicante, 03690 Alicante, Spain
[2]Department of Software and Computing Systems, University of Alicante, 03690 Alicante, Spain
[3]Department of Physics, Systems Engineering and Signal Theory, University of Alicante, 03690 Alicante, Spain

Corresponding author: Antonio-Javier Gallego (jgallego@dlsi.ua.es)

This work was supported in part by the Spanish Government's Ministry of Economy, Industry, and Competitiveness under Project RTC-2014-1863-8, and in part by the Babcock MCS Spain under Project INAER4-14Y (IDI-20141234).

ABSTRACT This paper presents a method that can be used for the efficient detection of small maritime objects. The proposed method employs aerial images in the visible spectrum as inputs to train a categorical convolutional neural network for the classification of ships. A subset of those filters that make the greatest contribution to the classification of the target class is selected from the inner layers of the CNN. The gradients with respect to the input image are then calculated on these filters, which are subsequently normalized and combined. Thresholding and a morphological operation are then applied in order to eventually obtain the localization. One of the advantages of the proposed approach with regard to previous object detection methods is that it is only required to label a few images with bounding boxes of the targets to be trained for localization. The method was evaluated with an extended version of the MASATI (MAritime SATellite Imagery) dataset. This new dataset has more than 7 000 images, 4 157 of which contain ships. Using only 14 training images, the proposed approach achieves better results for small targets than other well-known object detection methods, which also require many more training images.

INDEX TERMS Artificial neural networks, learning systems, object detection, remote sensing.

I. INTRODUCTION

Systems for automatic ship detection are very important for maritime surveillance operations. They can be used to monitor marine traffic [1], illegal fishing, and sea border activities, and also during search and rescue operations such as the detection of bodies lying in the sea [2]. These types of algorithms are usually based on information gathered from satellite or aerial images, either by means of visible spectrum imagery or through the use of SAR-type sensors [3]–[6], and each one has different advantages and disadvantages.

The detection of small objects in large swaths of imagery is one of the primary problems in aerial imagery analytics [7], and is a particularly challenging task in satellite imagery. The objects of interest in this type of images are often very small and densely clustered, while in other types of lateral or general images the targets are much larger and more prominent, as occurs in the ImageNet dataset [8]. Moreover, objects viewed from overhead can have any orientation (e.g. ships can have any heading angle, whereas the traffic lights or trees in ImageNet are reliably vertical).

Object detection can be addressed using different strategies. The most evident technique is the use of a sliding window on the input image, which yields a prediction for each frame until the entire image has been processed. In this case, the accuracy of the detection varies according to the size of the window and the overlap used. However, this approach is very slow and computationally expensive. Most recent works overcome these limitations by performing classification and localization simultaneously.

The automatic detection of ships has been an active research field for decades, and continues to attract increasing interest. The first techniques used for ship detection were based on hand-crafted descriptors. For example, Lure *et al.* [9] and Weiss *et al.* [10] proposed a detection system for the tracking of ships using High Resolution Radiometer imagery. In this work, image features were first extracted and subsequently classified using similarity

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

measures obtained from features. More recent examples include the use of Boosted Local Structured HOG-LBP for object localization [11], a multi-fold multiple instance learning procedure [12], implicit cues from image tags [13], or image pixel intensity probabilities combined with LBP descriptors [14]. A complete review of ship detection methods can be found in [15].

Selecting hand-crafted features that can be employed to detect targets in images is a challenging task, particularly when objects have a different appearance and size. Recent image classification techniques have attempted to deal with this problem by making use of Deep Learning techniques [16] and, in particular, Convolutional Neural Networks (CNN), to perform classification without having to apply either hand-crafted feature extraction or pre-processing techniques. The performance of these networks has proven to be close to the human level, or even better for some types of tasks. Widely known CNN topologies include Xception [17], Inceptionv3 [18], ResNet [19], and VGG [20], among others.

For example, Wu *et al.* [21] classified ships using a CNN and then unified iterative bounding-box regression and ship classification in a multi-task network. In Yang *et al.* [22], in addition to the bounding boxes, the orientations of the ships were provided by using a model consisting of five parts: a Dense Feature Pyramid Network, an adaptive region of interest alignment, a rotational bounding box regression, a prow direction prediction, and a rotational non-maximum suppression. Yu *et al.* [23] used Haar-like features to obtain the approximate positions of ships, and then applied a PCNet architecture to the candidate windows.

Many deep learning methods are dedicated to the detection of objects in general. A review of those methods can be found in [24]–[26], while an evaluation of small object detection can be found in [27], which analyzes the results of known methods such as YOLO (You Only Look Once) [28], SSD (Single Shot MultiBox Detector) [29], and Faster R-CNN [30].

However, these types of techniques also have a number of disadvantages, principally the fact that, since they are supervised methods, they need a large amount of labeled data in order to be trained, which is a very expensive task in terms of time, resources and effort. In addition, methods using weakly-supervised techniques usually have a very low accuracy as regards detecting small objects. Moreover, object detection networks usually require adaptations when targets are very small [31], [32], which makes it impossible to apply this type of methods in a general manner.

In this paper, we propose a weakly-supervised deep learning method for efficient object detection. The method is particularly focused on the detection of small ships in satellite images and requires only a few training data labeled with the location (bounding boxes) of the ships to obtain their precise position. The proposed approach addresses the object detection task on the basis of a network trained for classification. The low precision of the weakly-supervised algorithms is improved through the use of a filter selection process. In this process, the filters learned by the categorical network are analyzed in order to select only those that will allow targets to be detected with greater precision. The method calculates the gradient obtained between the activations of each of these filters and the input image. It then normalizes and combines these gradients, in addition to applying a threshold and a morphological operation, in order to eventually obtain the location of the targets.

This approach was evaluated with an extended version of the MASATI (Maritime SATellite Imagery) dataset [1], to which more than one thousand images of ships were added, in addition to the labeling of their locations. The new dataset consists of a total of 7,389 aerial images, of which ships represent only 0.03 % of the pixels.

We also performed a comparison with current state-of-the-art approaches based on deep learning, and specifically with RetinaNet [33], Faster R-CNN [30], YOLO v2 [34], YOLO v3 [35], YOLT [7], and class-activation maps using backpropagation with VGG-16 and VGG-19 [36]. The results of this comparison are very competitive as regards small objects, particularly when the background is relatively uniform, as occurs with the ship detection task, thus demonstrating that the approach can generalize and learn with very few images.

The remainder of the paper is organized as follows: The following section provides a review of the state of the art of object detection methods; the proposed weakly-supervised object detection method is described in Section III; the new version of the MASATI dataset used for evaluation is described in Section IV; the series of experiments carried out is detailed in Section V, and finally, the main conclusions of this work are summarized in Section VI.

II. STATE-OF-THE-ART

In this section, we review the state of the art of object detection methods, which are, in the scope of this work, divided into supervised and weakly-supervised object detection methods.

A. SUPERVISED OBJECT DETECTION METHODS

Object detection methods can be roughly classified [24] as one-stage detectors (including methods such as YOLO [28], [34], [35], RetinaNet [33], or SSD [29]), two-stage detectors (Faster R-CNN [30] or YOLT [7]), cascade detectors (Bai & Ghanem [37]), and part-based models (Dai *et al.* [38]).

One of the first two-stage object detectors was Faster R-CNN [30], a method consisting of class-agnostic proposals and class-specific detections. In this work, the authors present an efficient fully convolutional approach denominated as Region Proposal Network (RPN) that can be used to propose regions. The detector further classifies and refines bounding boxes around those proposals.

One of the best-known single stage object detectors is YOLO (You Only Look Once) [28]. This architecture addresses object detection as a regression problem in order

to obtain spatially separate bounding boxes and associated class probabilities. A single neural network directly predicts bounding boxes and class probabilities from full images in one evaluation. YOLO v2 [34] was an improvement to the first version. In this new version, the image was divided into regions, and bounding boxes and probabilities were predicted for each region. It outperformed previous state-of-the-art methods, such as Faster R-CNN [30] and SSD [29]. YOLO v3 [35] is an improvement to YOLO v2 which, despite being larger, is faster and more accurate.

RetinaNet [33] proposes a *focal loss* that makes it possible to train a high-accuracy one-stage detector. The focal loss was designed to address the one-stage object detection scenario in which there is an extreme imbalance between foreground and background classes during training. The name RetinaNet originates from its dense sampling of object locations in an input image. Its design comprises an efficient in-network feature pyramid and the use of anchor boxes.

YOLT (You Only Look Twice) [7] is one of the specific methods for the detection of ships in satellite imagery. It is a two-stage detector consisting of a fully-convolutional neural network with a passthrough layer (similar to identity mappings in ResNet [19]) that concatenates the final layer onto the last convolutional layer, thus giving the detector access to the finer grained features of this expanded feature map.

A number of existing methods use feature maps for ship detection and have a similar architecture to Faster R-CNN (employing a two-step methodology). For example, Li *et al.* [39] proposed a topology similarly to Faster R-CNN (called HSF-Net) that employs a regional proposal network to generate ship candidates from feature maps. In Huang *et al.* [40], a new neural network architecture denominated as squeeze excitation skip-connection path networks (SESP-Nets) was proposed. The authors added a bottom-up path to a feature pyramid network to improve the feature extraction capability and obtain more accurate and multi-scale proposals.

B. WEAKLY-SUPERVISED OBJECT DETECTION METHODS

The localization of objects can also be estimated by using visualization methods, which have localization capabilities, despite not being explicitly trained to do so. These approaches use standard CNN trained for classification and analyze the feature maps (also called *activation maps*), which are the output activations of each convolutional filter. Some of these methods also consider error gradients in order to highlight those locations that have made the greatest contribution to the prediction of a particular class. Their output (namely *Saliency Maps* or *Class-Activation Maps*) serves to visually analyze what a network has learned and also to localize objects within the image.

One of the first methods for weakly-supervised object localization from CNN was proposed in [36]. This approach performs a single backpropagation (BP) pass to obtain the true gradient, which masks out negative bottom data entries via the forward ReLU [41]. The class-activation map for an input image and a given class is computed as the average of gradients for the filters when the feature map value is positive. In the context of this paper, we shall denominate this method as *BP*. A more recent technique, denominated as Class Activation Mapping (CAM), was proposed in [42]. In this case, the feature maps of the last convolutional layer are spatially pooled using a Global Average Pooling (GAP) [43] operation and are linearly transformed using the weights learned from the final layer to obtain the class-activation map.

The main issue of CAM is that it is necessary to adapt architectures with fully-connected layers in order to use this method, and also that it requires the retraining of multiple linear classifiers (one for each class) after the initial model has been trained. Grad-CAM (Gradient-weighted Class Activation Mapping) [44] was introduced to overcome these limitations and to enable its use with any CNN architecture without having to adapt it.

The proposed method belongs to this group, since it makes use of the feature maps learned by a CNN and the gradient obtained for each of the activation maps with respect to the input layer. However, this method introduces a filter selection process that uses only those filters that detect the target class with greater precision. It also combines the selected filers in order to improve the accuracy of the location and remove possible false positives.

III. METHOD

Previous weakly-supervised localization methods can help show the regions from the image that make the greatest contribution to the classification of a particular class. However, a CNN tends to focus on more elements than the main target to be searched, as some of these elements may contribute to the classification decision. For example, in our case, in addition to the ships, the network can detect whether there is sea or coast.

Some examples of this problem can be seen in Figure 1. The first row shows the original input image, while the second and third rows show the saliency maps after the application of backpropagation [36] and Grad-CAM [44], respectively. Figures 1(a) and (e) clearly show how the attention of the network focuses on other locations rather than the ship targets. In addition, depending on the architecture and the selected layer, the precision of localization may be very poor when the layer activation is high in a wide zone of the input image (see Figures 1 (b), (c) and (d)).

This occurs because a feature hierarchy is learned in the different convolutional layers of the CNN, from the low-level features (such as edges, corners, etc.) to the last convolutional layers (which are usually those employed to calculate the heatmaps or visual saliency), from which high-level features are obtained. However, in the last layers, filters are usually activated with different elements in the image, and the classification is eventually performed by using a combination of activations. This means that, for classification, some filters are activated that do not necessarily contain the target object,

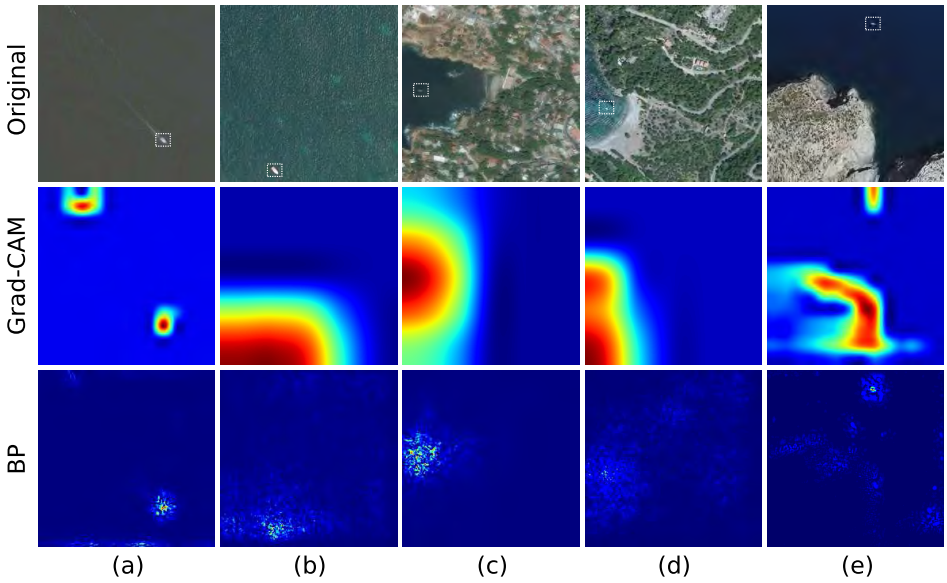


FIGURE 1. Examples of saliency maps from backpropagation [36] and Grad-CAM [44]. The first row shows the original images, in which ships are marked with a bounding box.

but rather other elements in the image that help perform the classification.

Figure 2 shows a subset of feature maps from different filters for a sample image, which contains coast and one ship. As can be seen, most of the filters do not have activations in the target location, and those filters that detect the ship also have activations for other elements from the image which may be helpful for classification.

A. SCHEME OF THE PROPOSED METHOD

The objective of the proposed approach is to use only those filters with high activation values for the target object. Figure 3 shows a scheme of the method. First, a categorical CNN is trained for classification. Once the weights have been learned, a *Filter Selection* process is performed to select the set of filters that maximize the precision as regards the location of the target class. Finally, in the inference stage, a new image is classified using the CNN and, if the predicted class corresponds to the target class, the subset of filters that was selected in the previous stage is then used to calculate its position in the image.

Steps 1 (Train CNN) and 2 (Fit FS) of the scheme in Figure 3 correspond to the training stage of the method, while step 3 corresponds to the inference stage, once the training stage has finished. Details of the steps in this method are provided in the following sections.

B. STEP 1 – TRAIN THE CNN

In this first step, a categorical CNN is trained for classification. In the experimentation, we evaluated two widely-known topologies of CNN for categorical classification, VGG-16 and VGG-19 [20]. These two architectures were selected because they obtained a good result for the classification of this dataset (close to 100%, as will be seen in the evaluation section), and also because they are frequently used as a basis for localization methods such as SSD [29], Faster RCNN [30] and CAM [42], among others.

VGG-16 has 13 convolutional and 3 fully-connected layers, whereas VGG-19 is composed of 16 convolutional and 3 fully-connected layers. Both topologies use dropout [45], max-pooling [46] and ReLU [41] activation functions.

Fine-tuning was performed for training, and the networks were initialized with the pre-trained weights from the ILSVRC dataset,[1] and then trained with the classes from our dataset. This process usually speeds up the training and obtains better results when domains are similar [47]. The last fully-connected layer of the pretrained networks was modified to match the number of classes in our dataset, as is usual in transfer learning tasks.

Training was performed by means of standard backpropagation using Stochastic Gradient Descent [48] and considering the adaptive learning rate method proposed in [49]. In the backpropagation algorithm, *categorical crossentropy* was used as the loss function between the CNN output and the expected result. The training stage lasted a maximum of 500 epochs with *early stopping* when the loss did not decrease during 10 epochs. The mini-batch size was set to 32 samples.

C. STEP 2 – FIT FILTER SELECTION

Once the CNN had been trained for classification, we proceeded to fit the filter selection algorithm. This process basically consisted of selecting the most relevant filters from this network for the location of a target class c. This was done

[1] ILSVRC is a 1,000 classes subset from ImageNet [46], a generic purpose database for object classification.

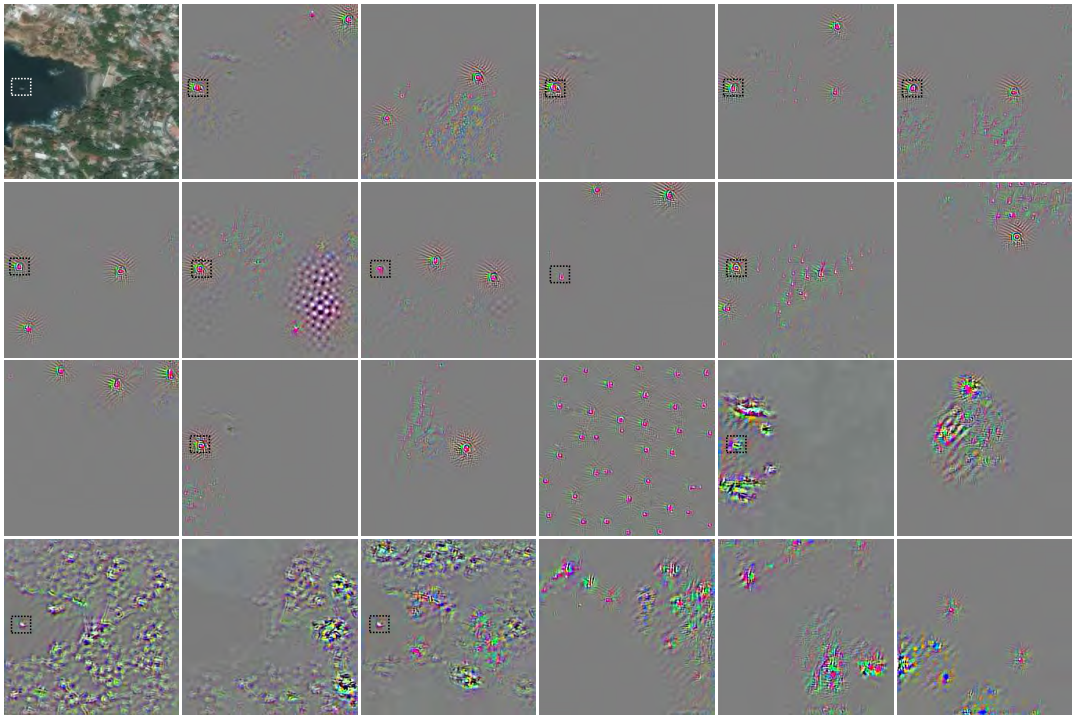


FIGURE 2. Example of the activations (feature maps) obtained when classifying a coast sample containing a ship. The top-left image shows the input sample, and the others display activations of a random subset of filters from the VGG-16 last convolutional layer. Ships are marked with a bounding box only if the activation detected it correctly.

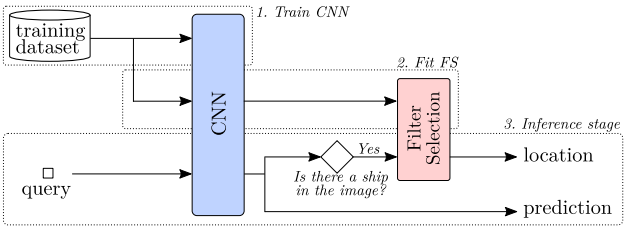


FIGURE 3. Scheme of the proposed method.

by calculating the subset of filters $\mathcal{F}^c \subseteq \mathcal{F}$ from all the possible filters $\mathcal{F}$ in the selected convolutional layer l whose average localization results were over a given threshold α. This subset of filters was subsequently used in the inference stage to obtain the location of targets for unseen images.

In order to obtain the subset $\mathcal{F}^c$, we first calculated the localization results obtained for each filter $f \in \mathcal{F}$. This was done by computing the prediction set $P_{l,f}^{(i)}$ for an input image i and a filter f from layer l, as follows:

$$P_{l,f}^{(i)} = Blobs((\tilde{G}_{l,f}^{(i)} > \beta) \oplus s) \tag{1}$$

where the set $\tilde{G}_{l,f}^{(i)}$ contains the normalized gradients in the range [0, 1] obtained for the filter f in layer l (see Equation 2). Only those values over a threshold β were selected for these gradients, thus allowing us to obtain a binary matrix of the same size $\mathbb{R}^{(w \times h)} \rightarrow [0, 1]^{(w \times h)}$, where w and h are the width and height of the input image, respectively. A dilation morphological operation (denoted by $\oplus$) was then applied with a structuring element s. Since the noise was removed by the thresholding operation, this dilation was intended to close small gaps and increase the size of the detections after thresholding. Finally, the function Blobs calculated the groups of connected pixels (or blobs), returning a list of bounding boxes containing the detected blobs.

The gradients $G_{l,f}^{(i)}$ of the filter f of layer l with respect to an input image i were computed by performing a single backpropagation pass, calculating the partial derivative of the activation map $A_{l,f}^{(i)}$ (also known as the feature map) obtained for the filter f with respect to the input image space I and evaluated at the image $I^{(i)}$. The gradients obtained were then rescaled in the range [0, 1] using the function r, as follows:

$$\tilde{G}_{l,f}^{(i)} = r\left(\frac{\partial A_{l,f}^{(i)}}{\partial I}\Big|_{I^{(i)}} \right) \tag{2}$$

where $A_{l,f}^{(i)}$ represents the activation map obtained by the filter f of layer l when the input image i is processed by the previously trained CNN.

As stated previously, the normalized gradients $\tilde{G}_{l,f}^{(i)}$ were used in Equation 1 to calculate the prediction set $P_{l,f}$ by selecting only the higher activations.

Once the prediction set $P_{l,f}^{(i)}$ had been obtained for all the selected input images I^c of a given class c, it was possible to calculate the subset of filters $\mathcal{F}^c$ that would be used to predict the location of that class in the inference stage. To do this, the Intersection over Union (IoU) between the prediction set

$P_{l,f}^{(i)}$ and the ground-truth was computed for all the images in that class. Only those filters whose average IoU was greater than a threshold α were selected from this result. Formally, the subset of filters $\mathcal{F}^c$ is calculated as follows:

$$\mathcal{F}^c = \left\{ f \in \mathcal{F} \mid \frac{1}{|I^c|} \sum_{i=1}^{|I^c|} IoU(P_{l,f}^{(i)}, B_g^{(i)}) > \alpha \right\} \quad (3)$$

where $B_g^{(i)}$ are the ground-truth localizations for the image i and class c, and $|I^c|$ represents the cardinality of the set I^c with the input images of class c.

In order to calculate the IoU of the predictions obtained for an input image i, each predicted bounding box from the set $P_{l,f}^{(i)}$ was mapped onto the ground truth ($B_g^{(i)}$) bounding box with which it had a maximum IoU overlap. A detection was considered to be positive if the area overlap ratio between the predicted bounding box and the ground truth bounding box exceeded a certain threshold λ according to Equation 4.

$$IoU(P_{l,f}^{(i)}, B_g^{(i)}) = \frac{\text{area}(P_{l,f}^{(i)} \cap B_g^{(i)})}{\text{area}(P_{l,f}^{(i)} \cup B_g^{(i)})} \quad (4)$$

where $\text{area}(P_{l,f}^{(i)} \cap B_g^{(i)})$ denotes the intersection between the object proposal and the ground truth bounding box, and $\text{area}(P_{l,f}^{(i)} \cup B_g^{(i)})$ denotes the union.

Once this stage was computed, the selected subset of filters $\mathcal{F}_c$ for each target class c was stored to be used in the inference stage for unseen images.

The influence of the different configuration parameters for the proposed method is evaluated in Section V-B, which provides a summary of the values selected.

D. STEP 3 – INFERENCE STAGE

Once steps 1 and 2 (corresponding to the training stage) have been completed, it is possible to use the proposed method to calculate the location of the ships. In the inference stage (see Figure 3), an input sample is forwarded through the trained model, and if the prediction for any of the target classes is positive (in our case, if a ship is detected), the feature maps of the network for that class are used to obtain its precise localization.

This is done by following the same steps as in Equation 1, but performing the sum of the gradients obtained from the selected subset of filters $\mathcal{F}^c$. The function $FS(i, c, l)$ calculates the localization of targets for a given class c for an input image i using the pre-calculated subset of filters $\mathcal{F}^c$ from layer l, as follows:

$$FS(i, c, l) = Blobs\left(\left(\left(\frac{1}{|\mathcal{F}^c|} \sum_{f=1}^{|\mathcal{F}^c|} \tilde{G}_{l,f}^{(i)}\right) > \beta\right) \oplus s\right) \quad (5)$$

where $|\mathcal{F}^c|$ represents the cardinality of the set $\mathcal{F}^c$.

As can be seen, Equation 5 is similar to Equation 1. However, Equation 1 calculates the prediction for a single filter, whereas Equation 5 performs the combination of the set of selected filters $\mathcal{F}^c$.

Note that during the inference stage, the proposed approach performs classification and localization simultaneously, as it is based on the filter activations obtained by classifying the image, that is, it is not necessary to perform any additional forward pass of the image through the network in order to calculate the localization.

IV. DATASET

The proposed method for the precise detection of ships was evaluated with an extended version of the MASATI (MAritime SATellite Imagery) dataset [1], which we will denominate as MASATI v2. For this work, we increased the size of this dataset by adding 1,177 new samples in order to balance the number of prototypes for the different classes. This new dataset contains a total of 7,389 satellite images in the visible spectrum. In this new version, the labeling with the bounding box for the ships' location was also included, in addition to the labeling at the class level. The new version of this dataset is freely available for the scientific community at http://www.iuii.ua.es/datasets/masati.

Images of different sizes were captured from Microsoft Bing maps in RGB, as these sizes were dependent on the region of interest to be registered in the image. In general, the average image size had a spatial resolution of around 512×512 pixels. The dataset was compiled at different times of the year and from different regions in Europe, the USA, Africa, Asia, the Mediterranean Sea and the Atlantic and Pacific Oceans.

Methods for automatic ship detection from optical imagery are affected by many factors, such as lighting or weather conditions. The proposed dataset considers a great variety of possible situations, thus enabling the proposed CNN approaches to obtain generic features. Figure 4 shows some examples from the MASATI v2 dataset.

Each image was manually labeled according to the following seven classes: *ship*, *coast & ship*, *detail*, *multi*, *sea*, *coast*, and *land*. Table 1 shows the sample distribution of each class. The "*ship*" class represents images in which a single ship appears within the image. The *multi* class describes images with two or more instances of ships. The ships in these two classes have lengths of between 4 and 10 pixels, with a bounding box area of between 6 and 154 pixels. The "*coast & ship*" class represents images in which one ship is close to the coast and has similar dimensions to the two classes mentioned previously. The "*detail*" class contains images with a single ship with a length of between 20 and 100 pixels. This class was used only to enhance the training process.

The most challenging class is "*multi*", since these images contain many examples of ships per image (a total of 1,966 ships are labeled in this class, and each image contains an average of 7 ships, although some images contain up to 82 ships). In addition, this class includes examples of both open sea and coast.

We evaluated the proposed method by creating two sets, one denominated as "*simple set*", which included all the classes, with the exception of the samples from "*multi*", and

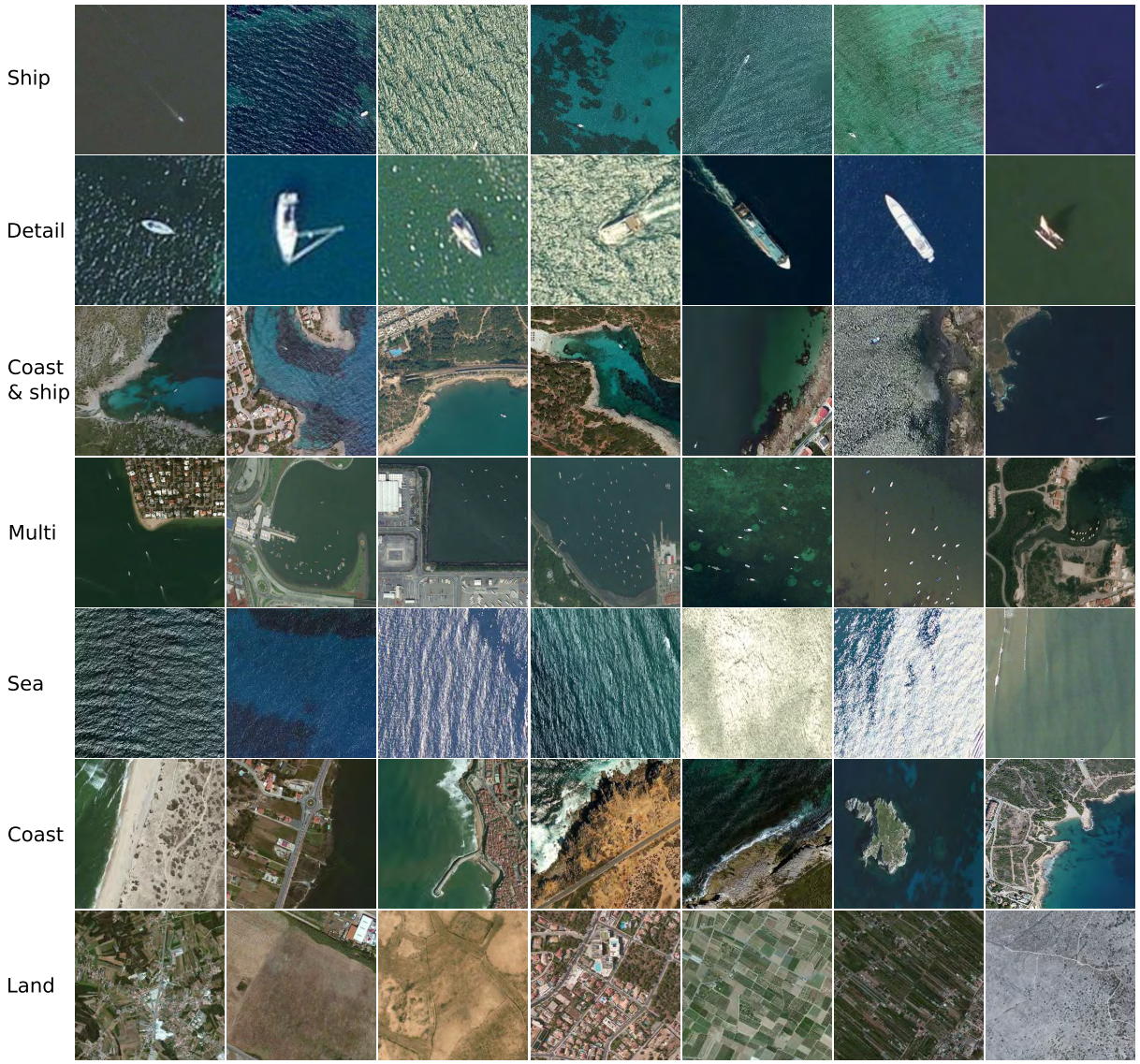


FIGURE 4. Image examples of different classes from the MASATI v2 dataset. The first four rows show the ship classes, while the three lower rows show the non-ship categories. The dataset samples are highly varied and the categories "*Ships*", "*Ships & coast*" and "*multi*" contain challenging images.

TABLE 1. Distribution of the classes in the MASATI v2 dataset.

Class	# samples	Description
Ship	1,027	Sea with a ship (no coast)
Detail	1,789	Ship details
Coast & ship	1,037	Coast with ships
Multi	304	Multiple ships (with and without coast)
Sea	1,022	Sea (no ships)
Coast	1,132	Coast (no ships)
Land	1,078	Land (no sea)

another denominated as "*complex set*", which included all the classes. This allowed us to carry out a better evaluation of the proposed method by first analyzing the precision of the detection of a single instance of small objects and then analyzing its behavior when multiple objects from the same class appeared.

Each of these two sets was divided into two, using 80% of the samples for training and the remaining 20% for the evaluation. These partitions did not overlap (i.e., the test set did not contain any of the samples seen during training) and the same percentage of samples of each class was kept in each partition. The same training and validation partitions were used to perform the experiments with all the methods, including the compared approaches.

With regard to the ship categories, we manually labeled bounding boxes with the exact locations of each ship in the image. This was done by using the LabelImg[2] tool, which

[2]Tzutalin. LabelImg. Git code (2015). https://github.com/tzutalin/labelImg

generates XML files in PASCAL VOC format. These data were used to validate the results of the proposed localization method.

V. EXPERIMENTS

In this section, we show the experimentation carried out for the different parts of the proposed method using the dataset described in Section IV. We first evaluated the first stage of the process, i.e. the results of the categorical CNN network, after which we analyzed the filter selection process by evaluating different parameter values. Finally, we compared the results obtained by the proposed approach with other state-of-the-art methods.

A. CATEGORICAL CNN

The first step of the proposed method involves training the categorical CNN network. As indicated in Section III-B, this is done by training the VGG-16 and VGG-19 networks using the dataset and the classes described in Section IV.

In order to evaluate the performance of this experiment, three evaluation metrics that are widely used for classification were chosen: Precision, Recall and F-measure (F_1). These metrics can be calculated using the following equations:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$F_1 = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FN} + \text{FP}} \tag{8}$$

where TP (True Positives) denotes the number of positive class samples correctly classified, FN (False Negatives) denotes the number of positive class samples that were misclassified, and FP (False Positives) denotes the number of predictions of the positive class that are incorrect.

Table 2 shows the average results (in percentages) obtained for the two sets considered (simple and complex sets). As can be seen, both networks obtain excellent average results, close to 100%, when discriminating between the different classes in the simple set.

Reliable results are also obtained in the case of the complex set, although they are slightly lower owing the complexity of the new samples. Note that the VGG-16 network obtains better results than VGG-19 for the simple set, but that this behavior is reversed for the complex set. This difference may be motivated by the complexity of the network and the number of parameters to learn, since it may perform overfitting for simple data, although in this case, the differences are very small.

Having shown how the networks to be used are trained, we shall now evaluate the second step of the proposed method: the filter selection stage.

B. FILTER SELECTION

In this section, we evaluate the filter selection process by analyzing the influence of the different hyperparameters. In order to simplify this analysis, we use the simple set, given that the results obtained for the complex set and the observed trends for the different hyperparameters were quite similar. Finally, the results obtained for the complex set are also reported.

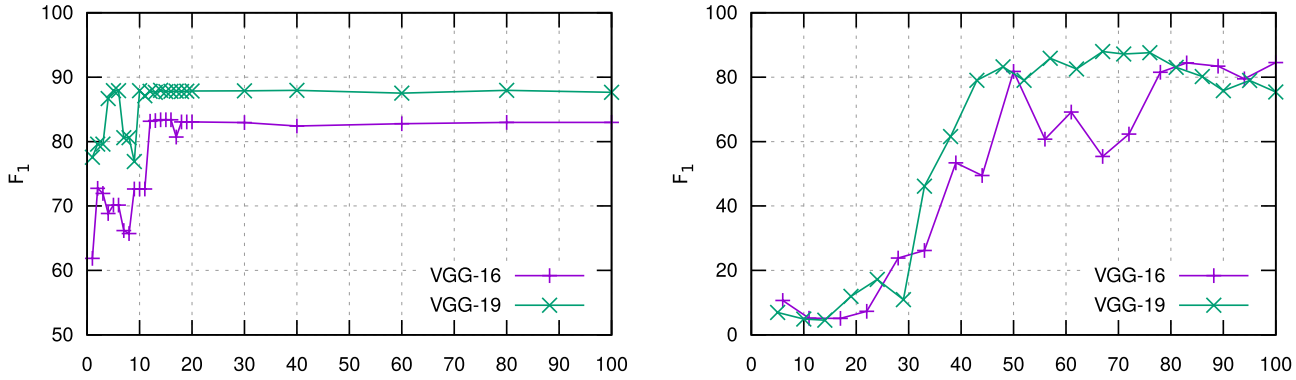
TABLE 2. Results obtained with the categorical CNN for the two sets considered (simple and complex sets).

Model	Dataset	Precision	Recall	F_1
VGG-16	*Simple*	99.31	99.57	99.44
	Complex	97.43	96.12	96.77
VGG-19	*Simple*	99.11	99.02	99.06
	Complex	97.42	97.39	97.40

The categorical networks trained for the simple set in the previous section are now employed to analyze the localization accuracy for the ship class obtained using the proposed method. In this case, we have merged the samples from the ''ships'' and ''coast & ship'' classes. This is because, as can be seen in Figure 3, the image is classified first and, in the case of obtaining a class that contains a ship, the filter selection method is used to recover its position in the image. The ''detail'' category was used only to improve the accuracy of the categorical networks (in order to provide more examples of ships at different scales). Since the ships in this class are centered and occupy almost the entire image, finding their location is not an issue.

The results are also evaluated using the F_1 metric (Equation 8), but in this case we measure the objects (or ships) whose location was correctly detected. This is done by calculating the bounding box of the predicted objects (P), which is then paired with the bounding box of the ground truth (B) with which it has a higher IoU (using Equation 4). A predicted bounding box P is considered to be properly localized if $\text{IoU}(P, B) \geq \lambda$. In this case, we set $\lambda = 0.5$ (a threshold value commonly used in this type of tasks, such as in PASCAL VOC), and calculate the metric F_1, considering the correct detections to be TP (when $\text{IoU} \geq \lambda$), the wrong detections to be FP (i.e., when a P does not overlap with any B), and those cases in which a ground truth object is not detected to be FN. Note that if multiple detections of the same object are predicted, only the first one is counted as a positive while the rest are counted as negative.

We first evaluated the influence of the training set size used to select the filters. This was done by conducting an experiment using an incremental training set, i.e., we took only one training image, performed the filter selection process and evaluated the result obtained. This process was then repeated with two training images, and so on, until 100 images had been evaluated (we stopped the experiment at this size since the results did not improve). In order to evaluate the influence of the training size, we froze the remaining parameters of the method, selecting the penultimate convolution layer of each network, a size $|\mathcal{F}^c| = 4$, $\beta = 0.8$ and a square structuring element of size 7×7. Figure 5a shows the result of this experiment. As can be seen, in both cases it is sufficient to



(a) Training set size (b) Network depth (%)

FIGURE 5. Localization results (F_1 %) obtained by varying (a) the size of the training set and (b) the layer of the network from which the filters are selected (given in percentage with respect to 100% of the total network depth).

label between 14 and 20 training images with bounding boxes to obtain the best accuracy and we, therefore, eventually set the size of the training set to only 14 labeled images.

Figure 5b shows the influence of the CNN layer selected for the localization calculation (variable l of Equation 3). This was done by computing the result obtained with all the layers from each of the two network models evaluated, while the remaining parameters were set to the aforementioned values. Since the VGG-16 network has a total of 18 layers and VGG-19 has 21, in this figure we represent the result as a function of the layer depth, where 100% of the depth means the last layer of the network. As can be seen, the results in the first part of both networks were not good (up to 40% of depth). However, as expected, better localization results were obtained in the last layers of the networks (from 70% of depth), from which the higher level representations of the images were extracted. The layer "*block5_conv2*" was, therefore, eventually selected for VGG-16, and "*block4_conv3*" for VGG-19 (see the full network architecture in [20]).

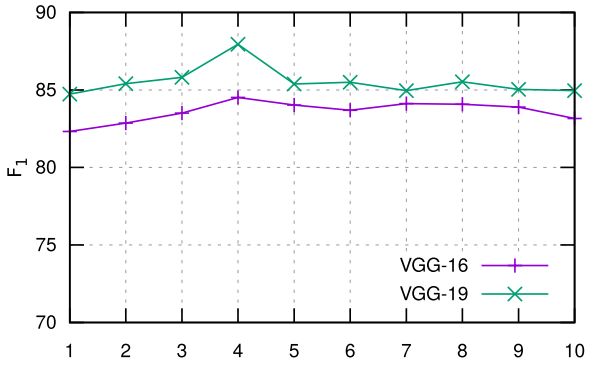
Another important variable to analyze is the number of filters selected in order to obtain the localization, that is, the size of the set $|\mathcal{F}^c|$ in Equation 5, which can be adjusted by modifying the threshold value α. For this experiment, we also used the best layer previously selected (which, in both cases, contains 512 filters), a training set of 14 images, $\beta = 0.8$ and a structuring element of 7×7. Figure 6a shows the results obtained by varying the number of filters used to calculate the location. As can be seen, a maximum is obtained for the two network models when using between 3 and 5 filters, and the best results are, in both cases, obtained with 4 filters. These filters were selected by setting the α threshold to 0.458 for VGG-16 and 0.454 for VGG-19.

Figures 6b and 6c show a histogram of the average IoU obtained for each of the filters in the selected layer of the VGG-16 and VGG-19 networks, respectively. The vertical coordinate of this graph was truncated to a maximum value of 30 filters in order to facilitate its visualization, since the first column, corresponding to the range $[0, 0.005]$ of IoU, contains 33% of the filters of VGG-16 and 6 % of those of VGG-19. Upon analyzing the results of VGG-16, it will be noted that 63.87 % of filters have an IoU that is lower than 0.2 and that only 13.48 % exceed 0.4, with the maximum value obtained by an individual filter being 0.4603. In the case of VGG-19, a lower percentage of filters does not exceed 0.2 (53.52 %) and more filters exceed 0.4 (25.00 %), with a very similar maximum value of 0.4604.

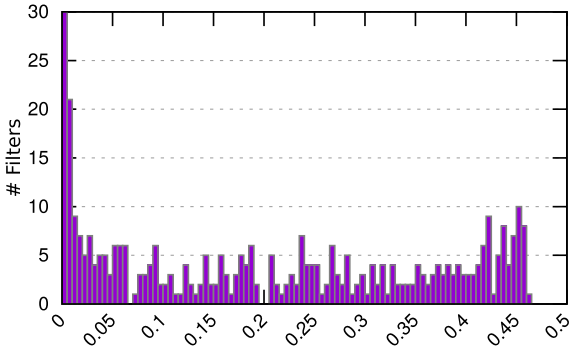
Another parameter to be analyzed is the size of the threshold β (see Eqs. 1 and 5). As before, we set the rest of the parameter values to the best ones found and varied this parameter only in the range [0, 1]. Figure 7a shows that better results are obtained with higher values for this threshold, i.e., when selecting only those pixels with the highest activations. The specific value selected for VGG-16 was $\beta = 0.94$, while that for VGG-19 was $\beta = 0.82$.

Finally, we also analyzed the influence of the size of the structuring element s (see Eqs. 1 and 5) that is used for the dilation of the result obtained from the filters' activation before calculating the bounding box with the position of the detected objects. The influence of this parameter was assessed by varying the size of the structuring element between 3×3 and 13×13, and setting the remaining parameters to the best ones found in the previous experiments. Figure 7b shows the result of this experiment. As can be seen, the result remains fairly stable when varying this parameter, and improves only slightly for the kernel size 7×7, which was why we eventually selected this size.

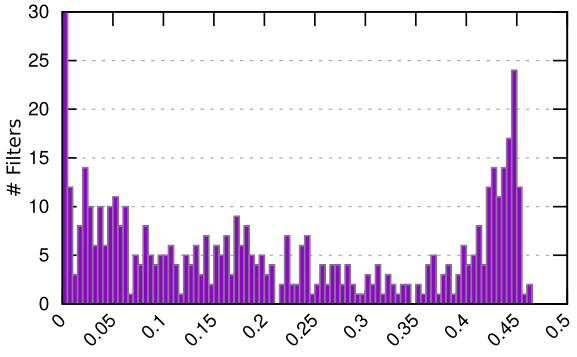
Table 3 shows the best hyperparameters found after carrying out the experimentation for both the simple and the complex sets. As can be seen, the results obtained with each of the networks for the two sets are very similar. In both, the same number of training images was used, the same layer was employed to extract the filters and the same kernel size was utilized. Variations occur only as regards the number of filters selected and the threshold β. In the case of the complex set, it would appear to be beneficial to combine more filters so as to obtain a more precise detection. With respect to the threshold β, a high value allows only the most likely detections to be selected, so in the case that the number of



(a) # filters ($|\mathcal{F}^c|$)

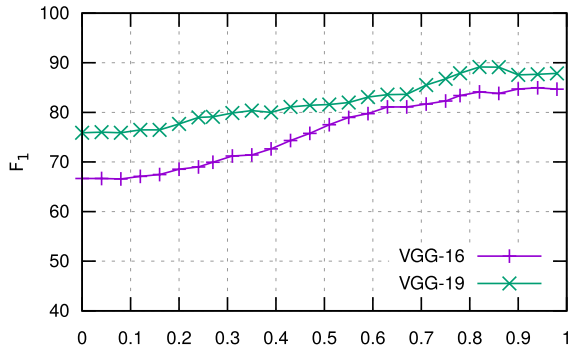


(b) Average IoU for VGG-16

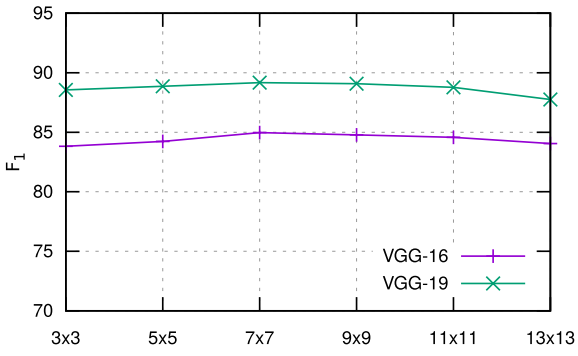


(c) Average IoU for VGG-19

FIGURE 6. (a) Localization results (F_1 %) obtained when varying the number of filters in the set $|\mathcal{F}^c|$. Figures (b) and (c) show the histogram with the average IoU obtained by each of the filters in the selected layer from the VGG-16 and VGG-19 networks, respectively.



(a) Threshold β



(b) Structuring element s size

FIGURE 7. Localization results (F_1 %) when varying (a) the threshold β and (b) the structuring element s size.

targets is reduced (as in the simple set), it is, therefore, better to use a higher value. However, in the case of the complex set, it is better to reduce this threshold slightly when attempting to detect many more objectives.

Figure 8 shows an example of the filters obtained for an input image, along with the process of adding up the result until the final prediction is obtained. A challenging image of a coast with a ship (located in the upper-left part) has been selected, for which most of the methods compared (as will be seen in the next section) make mistakes. The first row of this image shows the input image and the output obtained, while the second row shows the gradient obtained for the four filters selected. The last row shows, in the first column, the result for the filter 35, and in the following columns, the result of the incremental sum with the previous predictions. A higher activation value is indicated using dark red. As will be noted, when using only filter number 35, the prediction made is wrong (it detects a coast projection as a ship), but thanks to the combination of the four filters, the algorithm correctly detects the position of the ship.

C. COMPARISON

Having analyzed the different parameters of the proposed method and determined the best configuration (see Table 3),

TABLE 3. Configuration for each of the networks with the Filter Selection method obtained for the simple and complex sets. The size of the training set is not a parameter of the algorithm but it was evaluated to analyze its influence on the results. The α parameter also includes the number of selected filters in parentheses.

Parameter	VGG-16		VGG-19	
	Simple set	Complex set	Simple Set	Complex set
Training set size:	14	14	14	14
Layer l:	*block5_conv2*	*block5_conv2*	*block4_conv3*	*block4_conv3*
Threshold α:	0.458 (4)	0.457 (5)	0.454 (4)	0.453 (5)
Threshold β:	0.94	0.64	0.82	0.61
Kernel s **size:**	7×7	7×7	7×7	7×7

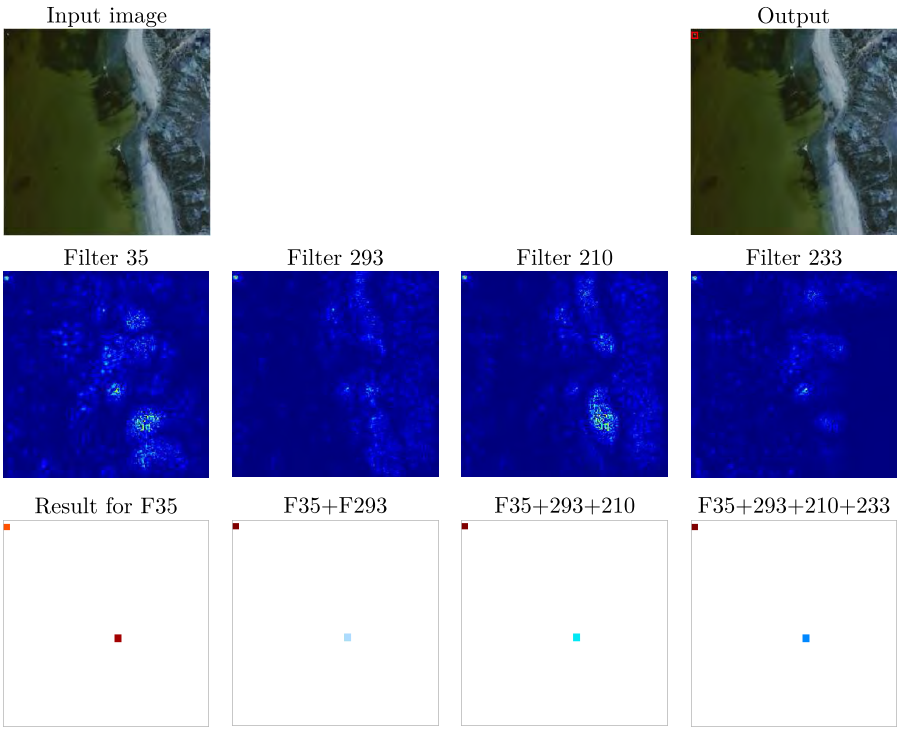


FIGURE 8. Example of the process performed to calculate the location of a ship. The first row shows the input and the output images, marking the bounding box of the detected ship in the upper-left part. The second row shows the gradients obtained by the four selected filters for the input image. The third row shows the process of incrementally adding up the result obtained. A higher activation value is indicated in dark red.

we will now compare the results obtained for the simple and complex sets with those of other state-of-the-art methods. In particular, we have compared our approach with the following methods (already described in the introduction):

- Visual saliency with backpropagation (BP) [36] using the VGG-16 and VGG-19 models.
- SelAE [5]: This approach uses a Selectional Auto-Encoder (SAE) network specialized in the segmentation of oil spills. It returns a probability distribution to which a threshold is applied in order to select the pixels to be segmented.
- Faster R-CNN (FRCNN) [30] and RetinaNet [33], which yielded competitive results for ship detection in SAR images in [50]. Both models use a ResNet50 network initialized with pre-trained weights from ILSVRC. Training included data augmentation, and the size of the anchors was adjusted to the average size of the bounding boxes from our dataset in order to improve the accuracy with small objects.
- YOLO v2 [34], YOLO v3 [35] and YOLT [7] initialized with pre-trained weights from ILSVRC. These models were also trained with data augmentation, adjusting the size of the anchors as occurred with the previous methods. In the case of YOLT, the parameter "*min retain prob*" was set to 0.35, as [7] stated that the highest F_1 score was obtained using values of between 0.3 to 0.4.

For this comparison, in addition to the metrics (Precision, Recall and F_1) previously used at the object detection level, we show the average value of the IoU obtained, along with the Average Precision (AP), given that these metrics are widely used to evaluate object detection methods, such as in PASCAL VOC challenge. The most recent PASCAL's challenge AP metric has been used (by interpolating all points rather than using a fixed set of uniformly-spaced recall values) [51].

TABLE 4. Comparison of the results obtained for the simple set using the proposed method (VGG-16/19 + FS) and other state-of-the-art methods. These results were calculated by employing a threshold of $\lambda = 0.5$ in the IoU metric to consider a correct detection. The two best results for each metric are marked in bold type.

Method	Precision	Recall	F_1	Avg(IoU)	AP
VGG-16 + BP	69.16	81.92	75.00	62.10	81.54
VGG-19 + BP	64.78	75.00	69.52	59.09	74.79
FRCNN	80.79	63.08	70.84	70.91	52.85
RetinaNet	70.61	61.92	65.98	64.50	51.89
YOLO v2	75.77	66.15	70.64	67.03	57.52
YOLO v3	**88.67**	87.31	**87.98**	**82.60**	85.99
SelAE	68.35	93.85	79.09	63.20	93.06
YOLT	74.44	77.31	75.85	**86.98**	67.35
VGG-16 + FS	77.12	**94.62**	84.97	69.91	**94.36**
VGG-19 + FS	**84.01**	**95.00**	**89.17**	75.50	**94.98**

This metric calculates the mean value in the recall interval [0, 1], which is equivalent to the area under the curve (AUC) of the Precision-Recall curve (PRC).

Table 4 shows the results of the comparison with other methods obtained for the simple set. For each metric, the two best results are marked in bold type. In general, the proposed method appears among the two best in all the metrics, with the exception of the average IoU, although this indicates only that the accuracy of the detected area is slightly lower, being necessary to analyze the other metrics in order to count the number of correct detections. Upon observing F_1, it will be noted that the best results are obtained with VGG19+FS (our proposal) and with YOLO v3, and that the proposed method is 1.19% better than YOLO v3. Note that the proposed approach has been trained using only 14 images labeled with the location, while YOLO v3 used the entire dataset with the bounding boxes. In the case of the AP metric, note that the proposed method has obtained the best results for the two network models to which it has been applied.

With regard to the results obtained for the complex set (see Table 5), the proposed method also obtained competitive results. The best results with the F_1 metric were obtained by VGG-19+FS followed by YOLT. The result obtained with VGG-16+FS was also, in this case, among the best. The YOLO v2 and v3 methods did not perform so well when dealing with multiple objectives, and in this case, other approaches that are more oriented toward the detection of multiple small objects, such as YOLT or SelAE, obtained better results. The latter were the two that obtained the best results for the AP metric, although the proposed approach also obtained results close to them. It should be noted that the remaining methods used the complete training set, labeled with the location of the ships, while the proposed method used only 14 labeled images for this purpose.

We also analyzed the capability of the different methods evaluated to generalize when processing images with a different number of targets to those it was trained to detect, and also verified whether they can extrapolate the knowledge learned using a small subset of the training data to the full dataset. The results of this experiment are shown in Table 6. In this case, we analyzed only the F_1 and AP metrics for each of the tests performed.

The first two columns in this table show the results obtained when evaluating the different methods with the complex set but using the models trained with the simple set. In this case, the methods that best generalize are SelAE and VGG19+FS, with the latter being only 1.6% of F_1 below. Please recall that SelAE was trained using all the images and by applying data augmentation (signifying that it may help to generalize better). As shown previously, some methods, such as YOLO and BP, are very dependent on the training set, which cannot generalize well when processing images with a greater number of targets, even though the objects and the type of images are the same (up to 50% worse in the case of YOLO v3 or 46% in the case of VGG16+BP).

In the central and last columns in this table, the learning and generalization capabilities are evaluated by training the different methods on a reduced set of data (using the same number of images as in the proposed method, that is, only 14). It is, therefore, also possible to evaluate how the other methods behave when a large amount of training data is not available. As can be seen, the results obtained worsen considerably for all the methods compared, decreasing by between 30% and almost 70% in some cases. For the simple set (central columns), the compared method that works best with few data is YOLO v3 followed by RetinaNet, and for the complex set (columns on the right) it is YOLT, which obtains a fairly stable result in both cases. However, upon comparing these results with those of the proposed method, there is a very significant difference, showing their generalization capability.

Figure 9 shows a comparison of the results obtained with the different methods. An example of each method is selected (see the columns in the figure) for some of the images that were most difficult. The bounding boxes of the detections obtained are marked for each result (TP in green and FP in red), and a colored circle has been added in a corner to indicate whether the detection was successful (green), whether the detection failed (red), or whether the targets were detected but false positives were also obtained (blue). As will

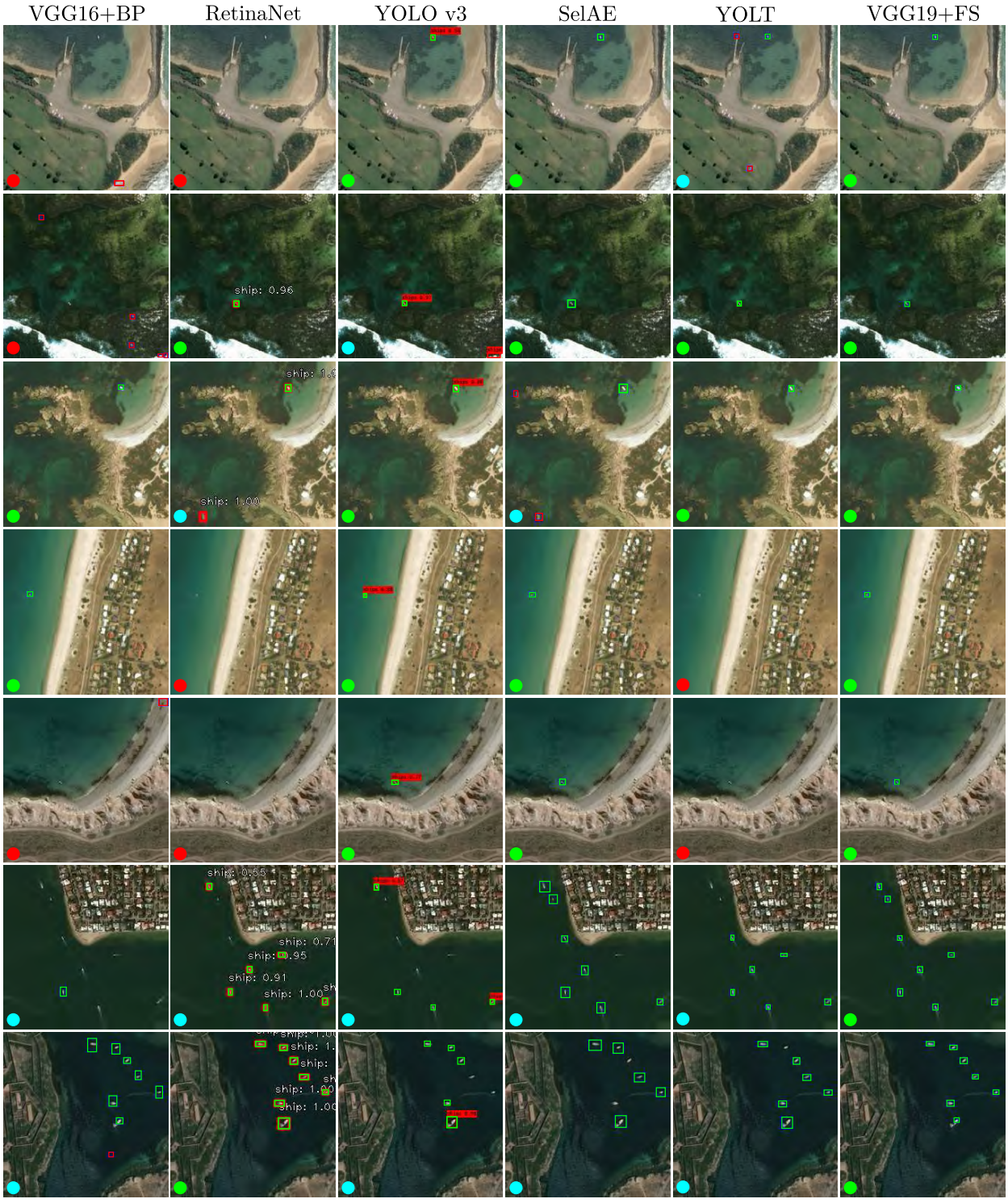


FIGURE 9. Example of results obtained by the different methods, including the proposed method (VGG19 + FS). Examples of each type of method for some of the images that were most difficult are shown. The bounding boxes of the correctly detected ships (TP) are marked in green and the incorrect detections (FP) in red. FN are not marked. A colored circular indicator has also been added to facilitate the visualization of a correct (green), incorrect (red), or partially correct (blue) detection.

be noted, the most reliable methods are SelAE, YOLO v3 and VGG19+FS, which detected all the targets and yielded only some FP. Some images that may appear to be simple, such as those shown in the 4th and 5th rows, are problematic for the BP, RetinaNet and YOLT methods, principally owing to the small size of the objects to be detected. The last two

TABLE 5. Comparison of the results obtained for the complex set using the proposed method (VGG-16/19 + FS) and other state-of-the art methods. These results were calculated by employing a threshold of $\lambda = 0.5$ in the IoU metric to consider a correct detection. The two best results obtained for each metric are marked in bold type.

Method	Precision	Recall	F_1	Avg(IoU)	AP
VGG-16 + BP	78.03	72.14	74.97	69.60	71.12
VGG-19 + BP	82.58	66.67	73.78	60.44	66.03
FRCNN	44.92	65.63	53.33	70.59	44.37
RetinaNet	77.55	59.38	67.26	56.38	53.79
YOLO v2	76.72	46.35	57.79	42.79	39.45
YOLO v3	**96.36**	55.21	70.20	50.29	54.27
SelAE	69.82	**92.19**	79.46	**81.08**	**85.90**
YOLT	77.46	**84.11**	**80.65**	**95.22**	**74.67**
VGG-16 + FS	79.89	78.65	79.27	70.45	71.52
VGG-19 + FS	**93.53**	75.26	**83.41**	67.76	74.18

TABLE 6. Evaluation of the generalization capabilities of the different methods analyzed. In the first columns, we compare the results obtained when training with the simple set but using the complex set (with more targets) for testing. The central and last columns show the results obtained when training with a reduced amount of data but evaluating on the full test set. For each metric and column, the two best results are marked in bold type. In all cases, a threshold of $\lambda = 0.5$ is used in the IoU metric to consider correct detection.

	Train with simple set and test with complex set		*Train with 14 images of the simple set (and test with simple set)*		*Train with 14 images of the complex set (and test with complex set)*	
Method	F_1	AP	F_1	AP	F_1	AP
VGG-16 + BP	28.72	16.62	9.99	7.65	3.51	3.89
VGG-19 + BP	27.77	16.02	7.38	7.34	4.75	2.80
FRCNN	59.08	41.61	34.22	20.09	14.36	5.00
RetinaNet	60.58	43.84	51.32	42.78	29.19	16.67
YOLO v2	34.81	20.92	40.60	37.55	16.28	8.30
YOLO v3	37.45	23.05	64.49	50.28	7.50	3.87
SelAE	**72.18**	**56.77**	10.90	24.93	32.76	44.29
YOLT	47.91	30.45	48.17	32.26	43.59	27.08
VGG-16 + FS	64.27	44.46	**84.97**	**94.36**	**79.27**	**64.52**
VGG-19 + FS	**70.58**	**54.49**	**89.17**	**94.98**	**83.41**	**71.18**

rows show examples for the *multi* class, and in this case, the SelAE, YOLT and VGG19+FS methods also obtain the best detection results.

VI. CONCLUSIONS

This work presents a weakly-supervised approach for object detection that can be applied to CNN classification models. The proposed method is specialized in the detection of small objects (that is, objects that occupy a very small percentage of pixels within the image) from satellite images. The localization is performed by applying a Filter Selection process in order to obtain the set of filters that allow the target class to be detected with greater precision. The gradients are calculated on these filters with respect to the input image, and are then normalized and combined. A thresholding and a morphological operation are subsequently applied to eventually obtain the location. This method makes it possible to adapt a network that has already been trained for classification to a network for object detection, using only a few images labeled with the corresponding bounding boxes for localization.

This approach was evaluated with an updated version of the MAritime SATellite Imagery (MASATI) dataset, which was extended for this work. We have specifically increased the number of samples from the 6,212 that were employed in the previous version of MASATI to 7,389 in this new version, principally by adding new samples to the "coast & ship" and "multi" classes. We have additionally labeled the ground-truth with the location of ships, which was not provided in the previous version.

The results obtained when analyzing the different parameters of the proposed method show that, in general, this method needs to be trained with between only 14 and 20 images containing the location of ships in order to obtain precise results. When employing more than 20 images, the score remains stable and there are no significant improvements. In addition, it was also observed that the best location results are obtained when using the last (deepest) layers of the network. With regard to the filters in the selected layer, the method needs to combine between only 4 and 5 filters to calculate the locations of ships.

When compared to other state-of-the-art methods, the proposed approach is able to achieve the best average scores for the detection of a single target. It obtains similar results to YOLT and YOLO v3, but with the difference that it requires only a few labeled samples. When calculating the location of multiple targets, the method obtains reliable results. It yields the best results according to the F1 metric, and similar results to YOLT, SelAE and YOLO v3 according to the AP and the IoU metrics. In addition, when analyzing the generalization capacity by evaluating the method for the localization of multiple objectives but using the model trained for single objectives, or when training with a reduced set of images, the proposed method is also among those that obtain the best results.

As future work, we intend to carry out more exhaustive experiments with the proposed method by evaluating it with other generic object detection datasets, analyzing the results with larger targets, and also evaluating the extension to multi-class.

REFERENCES

[1] A.-J. Gallego, A. Pertusa, and P. Gil, "Automatic ship classification from optical aerial images with convolutional neural networks," *Remote Sens.*, vol. 10, no. 4, p. 511, Mar. 2018.

[2] A.-J. Gallego, A. Pertusa, P. Gil, and R. B. Fisher, "Detection of bodies in maritime rescue operations using unmanned aerial vehicles with multispectral cameras," *J. Field Robot.*, vol. 36, no. 4, pp. 782–796, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21849

[3] J. Jiao, Y. Zhang, H. Sun, X. Yang, X. Gao, W. Hong, K. Fu, and X. Sun, "A densely connected end-to-end neural network for multiscale and multiscene SAR ship detection," *IEEE Access*, vol. 6, pp. 20881–20892, 2018.

[4] J. Zhao, Z. Zhang, W. Yu, and T.-K. Truong, "A cascade coupled convolutional neural network guided visual attention method for ship detection from SAR images," *IEEE Access*, vol. 6, pp. 50693–50708, 2018.

[5] A.-J. Gallego, P. Gil, A. Pertusa, and R. B. Fisher, "Segmentation of oil spills on side-looking airborne radar imagery with autoencoders," *Sensors*, vol. 18, no. 3, p. 797, 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/3/797

[6] Z. Deng, H. Sun, S. Zhou, and J. Zhao, "Learning deep ship detector in SAR images from scratch," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 4021–4039, Jun. 2019.

[7] A. Van Etten, "You only look twice: Rapid multi-scale object detection in satellite imagery," *CoRR*, vol. abs/1805.09512, May 2018. [Online]. Available: http://arxiv.org/abs/1805.09512

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[9] F. Y. M. Lure and Y.-C. Rau, "Detection of ship tracks in AVHRR cloud imagery with neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Aug. 1994, pp. 1401–1403. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=399451

[10] J. M. Weiss, R. Luo, and R. M. Welch, "Automatic detection of ship tracks in satellite imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp., Remote Sens.-Sci. Vis. Sustain. Develop.*, vol. 1, Aug. 1997, pp. 160–162. [Online]. Available: http://ieeexplore.ieee.org/document/615827/

[11] J. Zhang, K. Huang, Y. Yu, and T. Tan, "Boosted local structured HOG-LBP for object localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1393–1400.

[12] R. G. Cinbis, J. Verbeek, and C. Schmid, "Multi-fold MIL training for weakly supervised object localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 2409–2416.

[13] S. J. Hwang and K. Grauman, "Reading between the lines: Object localization using implicit cues from image tags," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1145–1158, Jun. 2011.

[14] F. Yang, Q. Xu, and B. Li, "Ship detection from optical satellite images based on saliency segmentation and structure-LBP feature," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 602–606, May 2017.

[15] U. Kanjir, H. Greidanus, and K. Oštir, "Vessel detection and classification from spaceborne optical images: A literature survey," *Remote Sens. Environ.*, vol. 207, no. 15, pp. 1–26, Mar. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0034425717306193

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14539%5Cn10.1038/nature14539

[17] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," Oct. 2016, *arXiv:1610.02357*. [Online]. Available: https://arxiv.org/abs/1610.02357

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, Dec. 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

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

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, Sep. 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[21] F. Wu, Z. Zhou, B. Wang, and J. Ma, "Inshore ship detection based on convolutional neural network in optical satellite images," *IEEE J. Sel. Topics Appl. Earth Observat. Remote Sens.*, vol. 11, no. 11, pp. 4005–4015, Nov. 2018.

[22] X. Yang, H. Sun, X. Sun, M. Yan, Z. Guo, and K. Fu, "Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network," *IEEE Access*, vol. 6, pp. 50839–50849, 2018.

[23] Y. Yu, H. Ai, X. He, S. Yu, X. Zhong, and M. Lu, "Ship detection in optical satellite images using haar-like features and periphery-cropped neural networks," *IEEE Access*, vol. 6, pp. 71122–71131, 2018.

[24] S. Agarwal, J. O. D. Terrail, and F. Jurie, "Recent advances in object detection in the age of deep convolutional neural networks," 2018, *arXiv:1809.03193*. [Online]. Available: https://arxiv.org/abs/1809.03193

[25] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," 2018, *arXiv:1809.02165*. [Online]. Available: https://arxiv.org/abs/1809.02165

[26] D. Chaves, S. Saikia, L. Fernández-Robles, E. Alegre, and M. Trujillo, "A systematic review on object localisation methods in images," *Revista Iberoamericana Automática Informática Ind.*, vol. 15, no. 3, pp. 231–242, 2018. [Online]. Available: https://polipapers.upv.es/index.php/RIAI/article/view/10229

[27] P. Pham, D. Nguyen, T. Do, T. D. Ngo, and D.-D. Le, "Evaluation of deep models for real-time small object detection," in *Neural Information Processing*, D. Liu, S. Xie, Y. Li, D. Zhao, and E.-S. El-Alfy, Eds. Cham, Switzerland: Springer, 2017, pp. 516–526.

[28] 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)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Computer Vision—ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 21–37.

[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 1–9.

[31] C. Eggert, S. Brehm, A. Winschel, D. Zecha, and R. Lienhart, "A closer look: Small object detection in faster R-CNN," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 421–426.

[32] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1951–1959.

[33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[34] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul.2017, pp. 6517–6525. doi: 10.1109/CVPR.2017.690.

[35] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *CoRR*, vol. abs/1804.02767, Apr. 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[36] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2013, *arXiv:1312.6034*. [Online]. Available: https://arxiv.org/abs/1312.6034

[37] Y. Bai and B. Ghanem, "Multi-scale fully convolutional network for face detection in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 2078–2087.
[38] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2018, pp. 764–773 [Online]. Available: http://ieeecomputersociety.org/10.1109/ICCV.2017.89
[39] Q. Li, L. Mou, Q. Liu, Y. Wang, and X. X. Zhu, "HSF-Net: Multiscale deep feature embedding for ship detection in optical remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 7147–7161, Dec. 2018.
[40] G. Huang, Z. Wan, X. Liu, J. Hui, Z. Wang, and Z. Zhang, "Ship detection based on squeeze excitation skip-connection path networks for optical remote sensing images," *Neurocomputing*, vol. 332, pp. 215–223, Mar. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092523121831508X
[41] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," *J. Mach. Learn. Res.*, vol. 15, no. 4, pp. 315–323, 2011.
[42] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.
[43] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. Int. Conf. Learn. Represent.*, *Proc. ICLR*, Apr. 2014, pp. 1–10.
[44] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from deep networks via gradient-based localization," 2016, *arXiv:1610.02391*. [Online]. Available: https://arxiv.org/abs/1610.02391
[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627435.2670313
[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1–9.
[47] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, Montreal, QC, Canada, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Cambridge, MA, USA: MIT Press, 2014, pp. 3320–3328.
[48] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*. Berlin, Germany: Springer, 2010, pp. 177–186.
[49] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*. [Online]. Available: https://arxiv.org/abs/1212.5701
[50] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, "Automatic ship detection based on retinanet using multi-resolution gaofen-3 imagery," *Remote Sens.*, vol. 11, no. 5, p. 531, 2019. [Online]. Available: http://www.mdpi.com/2072-4292/11/5/531
[51] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2014.

SAMER ALASHHAB received the B.Sc. degree in computer science from Al-Ahliyya Amman University, in 2001, the Higher Diploma degree in computer information systems, and the M.Sc. degree in computer information systems from the Arab Academy for Banking and Financial Sciences, Amman, Jordan, in 2004 and 2005, respectively, and the master's degree as an Expert in developing applications for smart devices from the University of Alicante, Spain, in 2013, where he is currently pursuing the Ph.D. degree in computer science. He has worked as a Researcher for the Computer Science Research Institute, University of Alicante. His research interests include pattern recognition, machine learning, and artificial intelligence. He is a member of the Spanish Association of Recognition of Forms and Analysis of Images (AERFAI).

ANTONIO-JAVIER GALLEGO received the B.Sc. and M.Sc. degrees in computer science, and the Ph.D. degree in computer science and artificial intelligence from the University of Alicante, in 2004 and 2012, respectively. He has been a Researcher on 10 research projects funded by both the Spanish Government and private companies. He is currently an Assistant Professor with the Department of Software and Computing Systems, University of Alicante, Spain. He has authored or coauthored 40 works published in international journals, conferences, books, and book chapters. His research interests include deep learning, pattern recognition, and computer vision.

ANTONIO PERTUSA received the B.Sc. degree in computer science and the Ph.D. degree from the University of Alicante, Spain, where he is currently an Associate Professor with the Department of Software and Computing Systems. He has been a Researcher on over 15 research and development projects funded by the Spanish Government agencies and private companies. He has authored or coauthored more than 40 works in international journals, conferences, and book chapters. His research interests include signal processing, deep learning, and pattern recognition methods applied to computer vision, music information retrieval, remote sensing, and medical knowledge extraction. He is a member of the executive committee of the Spanish AERFAI Association and is the Secretary of the University Institute of Computing Research (IUII), University of Alicante.

PABLO GIL (M'12–SM'14) received the B.Sc. degree in computer science engineering and the Ph.D. degree from the University of Alicante, Alicante, Spain, in 1999 and 2008, respectively, where he is currently an Associate Professor with the Department of Physics, Systems Engineering, and Signal Theory. From 2016 to 2018, he was the Secretary of the Computer Science Research Institute, University of Alicante, and is currently the Head of that research institute. He has also been a Researcher on over 18 research and development projects funded by the European Commission, Spanish Government agencies, and private companies. He has authored or coauthored more than 100 works in international journals (29 indexed in JCR), conferences, and book chapters. His research interests include computer vision, 3-D vision, deep learning, and perception for robots. He was a Guest Editor of a special issue for the *Journal of Sensors* and is an Associate Editor of the *International Journal of Advanced Robotics Systems and Mathematical Problems in Engineering*. He is a member of the Spanish Automatic Committee of IFAC and a Senior Member of the IEEE Robotics and Automation Society, Education Society, and the IEEE Sensor Council. Since 2018, he has been the Secretary of the IEEE-RAS Spanish Chapter. His awards and honors include the Teaching Excellence Prize at the University of Alicante, in 2011, and the Best Paper Award in the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2017).