Aerial Detection in Maritime Scenarios Using Convolutional Neural Networks

Gonçalo Cruz[1(✉)] and Alexandre Bernardino[2]

[1] Portuguese Air Force, Sintra, Portugal
gccruz@academiafa.edu.pt
[2] Computer and Robot Vision Laboratory, Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisboa, Portugal
alexandre.bernardino@tecnico.ulisboa.pt

Abstract. This paper presents a method to detect boats in a maritime surveillance scenario using a small aircraft. This method relies on Convolutional Neural Networks (CNNs) to perform robust detections even in the presence of distractors like wave crests and sun glare. The CNNs are pre-trained on large scale public datasets and then fine-tuned with domain specific images acquired in the maritime surveillance scenario. We study two variations of the method, with one being faster and the other one being more robust. The network's training procedure is described and the detection performance is evaluated in two different video sequences from UAV flights over the Atlantic ocean. The results are presented as precision-recall curves and computation time and are compared. We show experimentally that, as in many other domains of application, CNNs outperforms non-deep learning methods also in maritime surveillance scenarios.

Keywords: Convolutional neural network · Maritime detection · UAV · Aerial imagery

1 Introduction

In the last years, there has been a huge development on the use of Unmanned Aerial Vehicles (UAVs) both for recreational and business purposes. Although the gathering of information using these sensors is quite successful, transforming that data into information is more difficult. Often, a person is needed to inspect the data and, for instance, mark objects of interest in a surveillance mission. Although the flight duration of a UAV varies, an overwhelming amount of data can be gathered, making its analysis very tiresome.

Our work focuses on maritime surveillance scenarios, more specifically on the detection of vessels using an optical color sensor on-board a small UAV. This combination allows us to have a relatively simple and low cost system, especially when compared to systems that use radar technology. A typical surveillance flight is characterized by large intervals of time where nothing of interest is visible, interrupted by the appearance of objects of interest difficult to spot.

J. Blanc-Talon et al. (Eds.): ACIVS 2016, LNCS 10016, pp. 373–384, 2016.
DOI: 10.1007/978-3-319-48680-2_33

To make it more challenging, a surveillance scenario may need an immediate action to be taken (send rescue assets, follow a target of interest, etc.) and therefore the monitoring must be done in near real-time.

Some approaches have been proposed to automatically process the data. Most of these approaches are based on a set of handcrafted features that are created by an expert and need to be adjusted if the scenario of application changes. Our work presents the application of convolutional neural networks to learn these features and reliably detect vessels in a maritime surveillance scenario. Our detection technique will be deployed aboard the UAV to avoid compression and transmission artifacts, though, this restrains the amount of computing power available. To address these issues, our work focuses on presenting a method capable of detecting vessels with a high degree of certainty, despite the variability of the vessel's appearance and the presence of sun glare or wave crests. We evaluate the performance of our method and compare it with a state-of-the-art approach based on handcrafted blob analysis rules, currently employed in our missions. We also measure the computation time and assess its ability to run on an embedded platform.

This paper is organized as follows. Section 2 presents a brief overview of related work about convolutional neural networks and detection on UAV imagery. In Sect. 3, we detail the process used to perform the detection. Section 4 provides more detail about the networks that and datasets used, the training and the deployment stages. Section 5 evaluates the performance of the system and finally Sect. 6 provides some concluding remarks.

2 Related Work

2.1 General Object Detection

Object detection is one of the most important tasks in computer vision but it is still challenging in many scenarios [13]. One of these scenarios is the analysis of aerial imagery where factors like the amount of clutter, the variability of the appearance of the objects in the image and the variable lighting conditions, still limit the accuracy of detection. Traditionally, the detection has been performed by engineering a set of features (texture, intensity, *etc.*) that are relevant, extract these features from the images and feed them to a classifier. Therefore, the performance was quite dependent on the quality of the features. Often, the process of choosing the adequate features is laborious, specially because the designer has to guarantee that these features provide a good separability between different classes.

In the last years, a different path has been tracked with the use of convolutional neural networks (CNN). Instead of using hand-crafted features, these networks learn which features to use. In particular, CNNs have established the top results in several competitions, like ILSVRC [13]. This technique not only has outperformed most algorithms in generic classification and detection benchmarks but also in problems like character recognition [7] and pedestrian detection [14].

2.2 Detection on Aerial Images

Following the developments on other fields of Computer Vision, there have been several approaches to detection using aerial images. One interesting approach uses a cascaded classifier to detect people on foot and land vehicles [3]. In [11], people detection on land is also accomplished but using Histograms of Oriented Gradients, though, it depends on the small appearance variability of human body. In a maritime scenario, the objects can vary from a castaway or small dinghies to large cargo ships. Approaches like [12] depend on the movement of the targets to perform detection but this is not very well suited for the maritime environment as some targets may be still and undesired events like wave crests and sun glare may have a significant movement. It is therefore difficult to characterize possible targets with respect to size, shape, colors or textures.

Even with the aforementioned peculiarities, several specialized maritime detectors have been proposed. In [1], a set of engineered features is created to distinguish nautical objects from the ocean. However, the authors use several other layers to discard clutter. Similar approach is followed in [9] to detect marine mammals, with the authors using color features on a first stage and shape features on a second stage. Another interesting application is the detection of castaways, which is presented in Westall *et al.* [19]. In this case the author use a Hidden Markov Model to detect the head of castaways, which is typically has a size of 3 pixels.

Like in many other applications, the first kind of neural networks used in the analysis of aerial imagery were shallow neural networks.

Shirvaikar and Trivedi, as early as 1995, have proposed a system for the detection in aerial images that used the image as input and convolved the image with the input layer of the network [15]. This operation produced a 2D map that indicated the possibility of having a given object of interest in the image. More recently, Maire *et al.* [8] have proposed convolutional neural networks for the detection of marine mammals in aerial images. The authors also provide a meta-algorithm to refine the dataset used for the training of the network.

In our work, we test two approaches. The first is a sliding window approach based on [8], where we perform an exhaustive search over the entire image. The other approach creates several candidate regions that are supplied to the classifier, based on [17].

We compare the results against [10], which is the specialized detector that is in use on-board our UAVs and verify if any of the two techniques may be used to process video on-board, in near real time.

3 Detection System

The detection system that we propose is composed of two main parts. The first part is responsible for generating patches of images to classify. For this task we present and test the use of a sliding window approach and a method to propose salient candidate regions. The second part consists of a convolutional neural network that classifies each of the aforementioned image patches. In the

next two subsections, we detail the two different approaches to generate image patches and point their advantages and disadvantages.

3.1 Sliding Window

The first solution uses a sliding window approach to a given image captured by the camera, in a similar fashion to what was already described in other works as [18].

With a sliding window technique, overlapping image patches are extracted and fed to a classifier with the trained model, respectively represented in Fig. 1(a) and (b). Usually overlapping sliding windows are used and thus there is a significant possibility that a given object of interest is present on several patches. When a situation like Fig. 1(c) occurs, different patches may be correctly classified as belonging to the class *boat*, though there is only one boat on the image. Consequently, we merge the bounding boxes (BBs) that are near each other, into a single BB, obtaining a result similar to Fig. 1(d).

This approach is computationally expensive, as many regions are extracted and need to be classified. For instance, in our full resolution images (1920×1080 pixels), we use windows of 256 by 256 pixels with a overlap of 50 %, thus we get approximately 120 images patches to classify. With the classification of that many sub parts of the image, a lot of computational power is wasted, specially if we consider that many of these regions contain only ocean without any particular interest. This waste of resources could be even worse, if the search was done at several scales. In our procedure, we decided to search at only one scale. We allowed this simplification as the UAV has information of its position and parameters of the camera, therefore we may expect a typical size of the boat in the image.

3.2 Salient Candidate Regions

As we stated on the previous subsection, the sliding window approach performs an exhaustive search on captured image but wastes computational resources on

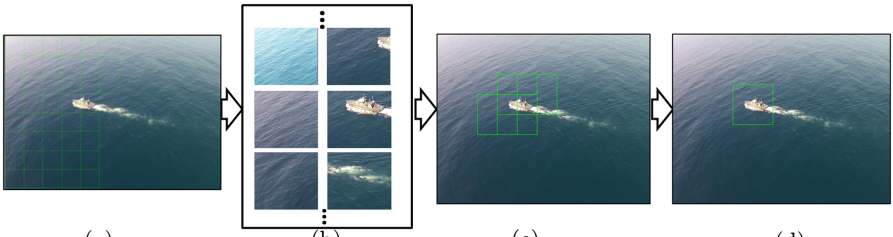


Fig. 1. Detection pipeline using sliding window: (a) captured image with several versions of the sliding window overlayed, (b) Image patches that were extracted and fed to the classifier, (c) regions that are classified as containing a boat and (d) BB obtained by merging several positively classified regions.

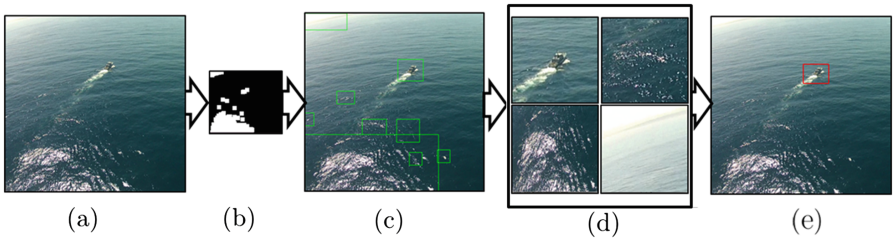


Fig. 2. Detection pipeline using candidate regions: (a) captured image, (b) binary image where white pixels correspond to salient regions, (c) candidate regions marked by green BBs, (d) image patches that are fed to the neural networks and (e) BB of the region that was validated by the neural network. (Color figure online)

areas that are easy to classify. We followed a strategy inspired in what is proposed in [17]. Instead of performing an exhaustive search over the complete image, we obtain candidate regions that are then classified as belonging to the class *boat* or *not boat*. We select regions that are salient, *i.e.* patches of the image that stand out from the background, and feed these patches to the neural network, which validates or discards these patches. This process is depicted in Fig. 2.

To create these candidate regions we first transform the image into the HSV colorspace and then threshold pixels with a hue between 105 and 240, considering that hue is represented by 8 bits. This threshold operation, which is similar to the first step of the method used on-board our UAVs [10], tries to isolate pixels that do not contain any shade of blue. After obtaining a binary image (represented in Fig. 2(b)), we perform a simple dilation and compute the connected components regions, which become the candidate regions (delimited in Fig. 2(c) by green BBs). Given that we want this step to be fast and we only want to identify regions of the image (not details), we used a reduced size version of the image. More specifically, we resize the image before the transformation to HSV colorspace and map the regions back to the original image size before extracting image patches.

Like in the sliding window approach, if several BBs are classified belonging to the class *boat*, the BBs are merged into one. Even though the computational complexity is a significant motivation behind the introduction of the candidate regions, it is not the only one. The other goal that we would like to achieve is to tailor the dimension of the BB to the object to detect. With the candidate regions approach, we allow each BB to have a variable size and aspect ratio, although some attention is needed as the input size of the convolutional neural network is fixed. To circumvent this limitation, we have followed three different paths. In case the candidate region is exactly the size of the input, we crop the image in that area. If the image region to classify is bigger than input of the network (256 by 256 pixels), then we shrink the image. If the image is smaller than 256×256, we use the area around the candidate region until the necessary dimension is attained.

4 Convolutional Neural Networks

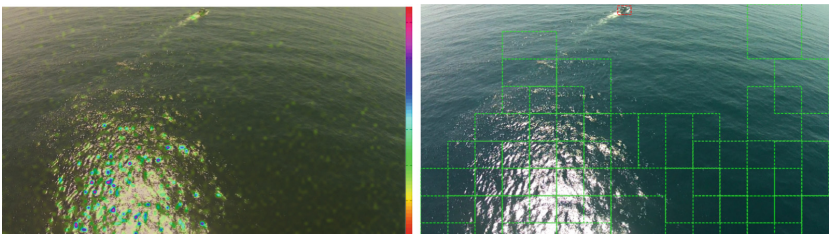
After the extraction of image patches (either with sliding window or with candidate regions), these have to be classified as belonging to the class of interest or not. To perform this task we have tested two popular convolutional neural network architectures, AlexNet and GoogLeNet, which were retrained with samples of the maritime scenario.

4.1 Dataset Selection and Training

The training of the convolutional neural network was done using CAFFE [5] framework, using 36698 image patches with a resolution of 256 by 256 pixels. These were extracted from 6 video sequences acquired by a UAV on a maritime surveillance scenario, at slightly different altitudes[1]. This set contained 30209 negative examples and 6489 examples of boats and were divided into a training and validation subset according to a 75 and 25 % ratio. The validation subset was used to avoid overfitting the network to the training data.

One out every five labelled objects contained in the video sequences was included in the positive sample set. In order to improve robustness to rotation, the positive set was augmented by including rotated versions (90°, 180° and 270°) of the original images. The process of selection of negative samples started with the computation of the image signature [4] for a given video frame, that typically highlights areas of the image that stand out from the background, as represented in Fig. 3(a). Subsequently, areas with high saliency are assigned a higher probability of being chosen in a random selection process that picks samples for the training and validation set, as visible in Fig. 3(b).

After obtaining the dataset, we have used CAFFE [5] to train the different networks. In particular, we have used a GoogLeNet network [16] and an AlexNet network [6] previously trained on the ImageNet Large Scale Visual Recognition



(a) (b)

Fig. 3. Extraction of examples for the training and validation set: (a) Image Signature Map for a given image with areas of higher saliency represented in blue and purple; (b) patches of negative examples selected for the dataset represented with a green BB and the positive example represented with a red BB. (Color figure online)

[1] The scale variability of the observed ships due to altitude variations is rather small.

Challenge (ILSVRC) 2012[2], which contains approximately 1.2 million images belonging to 1000 classes. Additionally, we only considered two possible classes (*boat* and *not boat*) and we have used a lower learning rate in this case, as we would like the network to change slowly, to take advantage of the large scale training.

4.2 Deployment of the Network

After obtaining a trained network, the detection of a boat in an image was achieved by assigning a probability to each candidate region. This probability was computed by the last layer of the networks (a softmax layer), that converts networks scores into probabilities of image to belong to a given class. In our case, the straightforward approach would be to consider a detection if the probability of the class *boat* was bigger than 50 %. Because we wanted to control the trade-off between the false positives and missed detections, we made the decision dependent on a threshold that can be defined by an operator. If a given patch was considered to contain a boat, a BB was created and the probability computed by the network for that image patch was associated to this BB, as shown in Fig. 4(a) and (b). In Fig. 2, from all the candidate regions represented in Fig. 2(c) and (d), only one is considered as belonging to the class *boat*. The validated area is presented in Fig. 2(e).

The first results for this detection scheme were performed on a common desktop equipped with a GPU card, to improve the training and classification speed of the network. On a second step, the implementation of the detection algorithm was done on a Jetson TK1 board, which is specially suited for embedded applications due to small power consumption. In our case this is a serious requirement as the power on-board is very restricted.

5 Evaluation

To evaluate the performance of the proposed detection scheme, we have used two video sequences that were left out during the training process. These are representative of two different flight conditions, one where the aircraft is lower (50 m above sea) and closer to the object of interest and other where the aircraft is higher (300 m) and farther.

5.1 Setup Description

The used camera has a 1/2.3 in. RGB sensor and a wide angle lens, and was mounted on a fixed wing UAV, moving at approximately 15 m/s. The camera was pointing 90° to the left-hand side of the aircraft and approximately 45° downwards with respect to the horizontal plane. The first sequence (sequence A)

[2] The authors of each architecture provide the weights, resulting from training on the mentioned dataset, at https://github.com/BVLC/caffe/tree/master/models/.

(a) (b)

Fig. 4. Typical video frame for (a) sequence A and (b) for sequence (b). Both images contain a correct detection with the respective probability associated to the BB.

contains 1400 frames with a resolution of 1920 × 1080 pixels and the object of interest is a boat with a length of 27 m and 6 m wide. The second sequence (sequence B) has the same resolution but contains 2250 frames, where a cargo ship is visible. Each sequence was labelled by an human operator and a Ground Truth (GT) corresponds to a rectangular BB. It is also worth noting that in sequence A, the object of interest has an average width of 111 pixels and an average height of 50 pixel. In sequence B the average dimension is 18 by 20 pixels.

As visible in Fig. 4(a) and (b), each classified image patch is given a probability of containing a boat, thus we have the ability of choosing the threshold, *i.e.*, the operating point of the detector.

5.2 Results

To quantify the performance of the algorithm, having in mind the already mentioned trade-off, we have used an evaluation approach similar to what is presented in [2] and display the results as precision and recall curves. Additionally, we have evaluated the computation time, to assess the ability of the proposed method to process a video feed in near real-time.

This evaluation method considers a detection as correct if there is any overlap between ground truth BB and the BB produced by the detection algorithm. In Fig. 5, we present the results obtained with the proposed methods in sequences A and B. Each of the plots contains four curves and these correspond to what was obtained using the sliding window (SW) and the candidate regions (CR) approach. We have used each of these approaches with two trained networks (GoogLeNet and AlexNet).

In the plots, there are also two points that correspond to the algorithm currently running in real-time time on-board our UAVs and presented in [10] applied to the same sequences, designated by *Blob Analysis* and used as baseline. *Blob Analysis* algorithm consists of three steps. In the first step, a set of blobs possibly containing a boat is extracted. The second step consists on the application of spacial constraints to reject blobs that contain sky or sun reflections. In the last

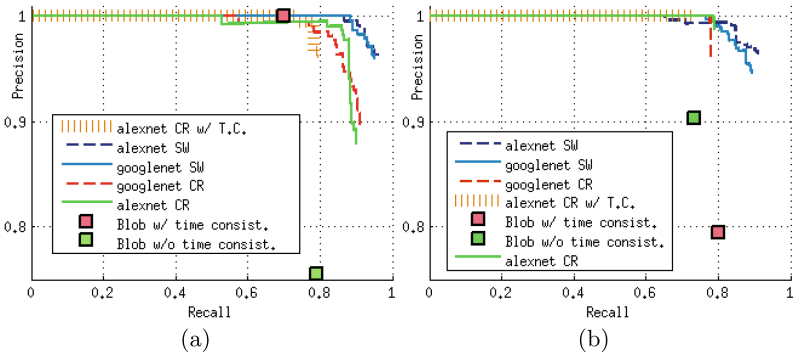


Fig. 5. Precision-Recall curves for (a) sequences A and (b) B.

step, the time consistency is checked by verifying if there are at least four detections in five consecutive frames. One of the points represents the results obtained by *Blob Analysis* applied to each frame individually (without time consistency), which is the closest comparison with the algorithm described in this paper. Nevertheless, we also present the results obtained with the entire *Blob Analysis*' detection pipeline and with time consistency applied to the candidate regions approach using AlexNet.

As visible on the plots, all approaches have a good performance, allowing a precision near 100 % for a recall of 50 %. The performance of our approach also outperforms what is achieved with *Blob Analysis* without the time consistency heuristic. When comparing with the results produced by the entire *Blob Analysis* pipeline, our technique (without the time consistency heuristic) produced slightly worse results in sequence A (for the same recall, our approach achieved 99.4 % precision instead of 100 %). On sequence B, our results are significantly better. The difference in performance is caused by the fact that *Blob Analysis* was originally tuned for boats with bigger apparent size, whereas the CNN was trained with boats of different dimensions. When comparing both tests that used the time consistency heuristic, we verify that the results obtained with the CNN were better.

Comparing convolutional neural networks' results, we can see that the sliding window approach, in general, produced better results, the only exception being a subtle difference on sequence B for recall between 0.7 and 0.8 (this is caused by the better fit of the candidate region to the boat with very small apparent size when compared to the fixed size sliding window). It is also possible to see that on sequence B, there is a rapid drop on the performance of the candidate regions approach around 80 % recall. This corresponds to more challenging conditions, where features like wave crests and sun glare are wrongly classified as belonging to the class *boat* (as depicted in Fig. 6(b)). In Fig. 6(a) is presented another challenging condition: a missed detection in an image where glare is present and the boat has a small apparent size. A correct detection in the presence of glare is also presented in Fig. 6(c). Finally in Fig. 6(d), we present the correct detection

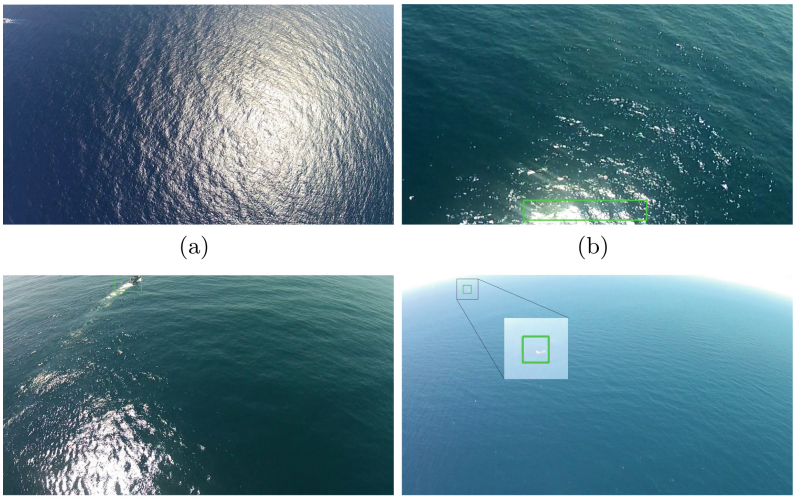


Fig. 6. Examples of challenging conditions present in the video sequences used in the evaluation: (a) missed detection of a ship with small apparent size in an image with the presence of glare, (b) false detection of ships in an image with significant presence of glare and correct detections of a ship in an image with significant glare and (d) ship with small apparent size very small contrast.

Table 1. Average computation time for each of the proposed approaches on a Nvidia Jetson TK1 board.

Approach	Average computation time (s)
AlexNet Sliding Window	10.58
GoogLeNet Sliding Window	15.11
AlexNet Candidate Regions	1.23
GoogLeNet Candidate Regions	1.86

of a boat with very small apparent size and very small contrast. It is also worth noting that in general, AlexNet yielded better results than GoogLeNet, even though in competitions like ILSVRC the opposite is true. We believe in our case, the amount of training samples is not enough to take full advantage of the deeper architecture of GoogLeNet.

When we consider real world application of any of these approaches, there is another important factor: the amount of time needed to process one image. The computation time for each of these techniques is summarized in Table 1. Despite the better results of sliding window approach, its computation time is not adequate to be used in a near real time application on the selected hardware platform. Even though the candidate regions approach cannot process video in

real time, it can still detect objects of interest in a real world application since boats are typically visible for a interval greater than its computation time.

6 Conclusion

Our work has showed that the detection system based on neural networks achieves good performance when compared to other algorithms and is possible to use in our envisioned scenario, *i.e.* an embedded system on-board an UAV. The gains go beyond the measured precision-recall, since the output of the network provides a probability of the BB belonging to a given class.

One of the future tasks that we might carry out is to use the output of the network to integrate this algorithm with a tracker, that will run at a higher rate and allow the extraction of more information.

Another of the future tasks is to use a Recurrent Neural Network to explore the temporal correlation of successive frames in order to improve detection, instead of analysing each frame individually.

Acknowledgements. This work was partially supported by FCT project [UID/EEA/50009/2013]. The authors would like to thank the SEAGULL team and Computer Vision Lab team (VisLab) at ISR/IST, which were involved in obtaining the image dataset and annotating the ground truth.

References

1. Dawkins, M., Sun, Z., Basharat, A., Perera, A., Hoogs, A.: Tracking nautical objects in real-time via layered saliency detection. In: SPIE Defense + Security, p. 908903. International Society for Optics and Photonics (2014)
2. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. **34**(4), 743–761 (2012)
3. Gaszczak, A., Breckon, T.P., Han, J.: Real-time people and vehicle detection from UAV imagery. In: IS&T/SPIE Electronic Imaging, p. 78780B. International Society for Optics and Photonics (2011)
4. Hou, X., Harel, J., Koch, C.: Image signature: highlighting sparse salient regions. IEEE Trans. Pattern Anal. Mach. Intell. **34**(1), 194–201 (2011)
5. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. arXiv:1408.5093 (2014)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
7. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
8. Maire, F., Mejias, L., Hodgson, A.: A convolutional neural network for automatic analysis of aerial imagery. In: 2014 International Conference on Digital lmage Computing: Techniques and Applications (DlCTA), pp. 1–8. IEEE (2014)

9. Maire, F., Mejias, L., Hodgson, A., Duclos, G.: Detection of dugongs from unmanned aerial vehicles. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2750–2756. IEEE (2013)
10. Marques, J.S., Bernardino, A., Cruz, G., Bento, M.: An algorithm for the detection of vessels in aerial images. In: 2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 295–300. IEEE (2014)
11. Oreifej, O., Mehran, R., Shah, M.: Human identity recognition in aerial images. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 709–716. IEEE (2010)
12. Pollard, T., Antone, M.: Detecting and tracking all moving objects in wide-area aerial video. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 15–22. IEEE (2012)
13. Russakovsky, O., Deng, J., Hao, S., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV) **115**(3), 211–252 (2015)
14. Sermanet, P., Kavukcuoglu, K., Chintala, S., LeCun, Y.: Pedestrian detection with unsupervised multi-stage feature learning. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3626–3633. IEEE (2013)
15. Shirvaikar, M.V., Trivedi, M.M.: A neural network filter to detect small targets in high clutter backgrounds. IEEE Trans. Neural Netw. **6**(1), 252–257 (1995)
16. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. arXiv:1409.4842 (2014)
17. Szegedy, C., Toshev, A., Erhan, D.: Deep neural networks for object detection. In: Advances in Neural Information Processing Systems, pp. 2553–2561 (2013)
18. Vaillant, R., Monrocq, C., Le Cun, Y.: Original approach for the localisation of objects in images. IEEE Proc Vis. Image Signal Process. **141**(4), 245–250 (1994)
19. Westall, P., O'Shea, P., Ford, J.J., Hrabar, S.: Improved maritime target tracker using colour fusion. In: 2009 International Conference on High Performance Computing & Simulation, HPCS 2009, pp. 230–236. IEEE (2009)