Ship Segmentation in Areal Images for Maritime Surveillance

Carlos Pires[1]
c.david.pires@tecnico.ulisboa.pt
Alexandre Bernardino[1]
alex@isr.tecnico.ulisboa.pt
Bruno Damas[1]
bdamas@isr.tecnico.ulisboa.pt

[1] Institute for Systems and Robotics, ISR
Instituto Superior Tecnico, IST
Universidade de Lisboa,
Lisbon, PT

Abstract

In this paper, we study and implement a method to detect ships during maritime surveillance missions. We implement a cascade model with a detection part followed by a segmentation stage. We use two convolutional neural networks, one for each section. With detection, we select the most likely regions to contain a ship, and after we segment those regions to identify the targets. We train the model with maritime datasets, and then we test it on the Airbus Ship detection challenge. The cascade model is capable of real-time ship segmentation, achieves a score of 0,82 in the challenge, and processes one image in 0,1 seconds.

1 Introduction

Maritime surveillance is a need for a country with a coast to prevent, discourage, and punish catastrophic and illegal events. Therefore, it is necessary to control all maritime activities. According to [1], Portugal has a coast with 943 kilometers and an exclusive economic zone (EEZ) with almost two million square kilometers divided into three regions. So, due to the amount of area, it is crucial to develop efficient and cost-effective tools. Unmanned Aerial Vehicles (UAVs) are flexible and extensible systems that can incorporate a vast number of sensors [2]. With them, we release a considerable amount of human resources. Image segmentation breaks the image down into meaningful regions, and we can separate a ship from the rest of the frame and estimate the ship size and route. Then, it is possible to compare the data with the automatic identification system (AIS). Plus, after applying segmentation, we can use techniques on the pixel level to improve ship detection performance. Object segmentation is a challenging task, especially when we are using images captured by UAVs because it is affected by factors like scale, perspective, and illumination variations. Images with glare and waves may confuse the model, which makes the segmentation task harder. The goal of this work is to implement an automatic real-time maritime surveillance system using drones with onboard cameras.

The main contribution is the development of a two stages system to perform ship segmentation. We present a cascade model with detection and segmentation, which makes the ship identification faster. The algorithm has two different stages: first, we use a fast detection network to search for possible ship locations regions, then we pass these regions through a segmentation network, thus narrowing the image region where segmentation is performed, which significantly improves the overall image processing time.

2 Related work

Since 2012, when A. Krizhevsky *et al*. [3] trained a convolutional neural network (CNN) on ImageNet and achieved outstanding results, the use of deep learning methods for detection and segmentation has increased exponentially. They have shown that with more layers (more depth), networks exhibit significant performance improvement. In detection algorithms, we try to generate a bounding box (BB) around the detected objects. In [4], R-CNN, Region with CNN features, takes an image as input and identifies where the primary target is. [5] presents a different approach to perform object detection with a network called YOLO - You Only Look Once. They treat object detection as a regression problem to separate BBs and to associate class probabilities. YOLO applies a single CNN to the full image, divides the image into regions, predicts BBs, and a score for each one.

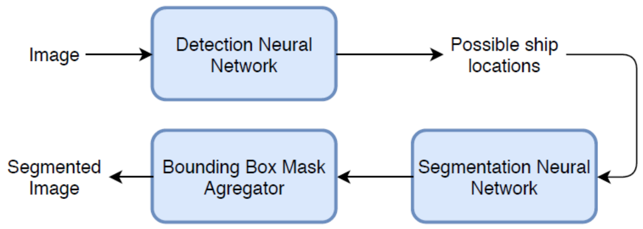

Figure 1: Segmentation system global architecture.

To successfully train a deep neural network many training samples are required. [6] presents an architecture and a training strategy with effective use of data augmentation to use the available training samples efficiently. The classification network proposed, U-Net, performs image segmentation by class and predicts a mask, which separates all the objects. This network has a U structure, and on the left side is done a contracting path to capture context, encoder. The right side has an expanding part that enables precise localization.

The authors of [7] use recorded detect data to help to detect a vessel. They investigate how temporal features could improve ship detection in video sequences captured by small aircraft. So, they build a convolutional Long short-term memory to learn those features and increase ship detection. [8] proposes a ship detection method in challenging surveillance conditions. They track vessels with a correlation filter complemented with image segmentation. To compensate for drifts in the correlation filter, they apply blob analysis to re-center the target in the tracking window. [9] presents a robust detector for aerial images. They modify a CNN and create tracks with the successive detections to predict the position of the objects in future frames.

3 Methodology

Our algorithm contains two phases: first, it applies image detection to identify possible locations of the ship, and then, these regions are segmented to check if there are ships. With these two parts, we intended to turn the segmentation task faster and better because we discard a considerable amount of background. So, now we are focused on segmenting particular regions of the image with a high probability of containing a ship. Then, the possibility of over-segmentation and miss segmentation will decrease. Figure 1 shows the system global architecture.

Our approach uses a detection network before segmentation to delete a considerable part of the image background. So, the cascade model with detection and segmentation has to be quicker than full image segmentation. We will use the YOLO network since it can process images in milliseconds, and it is computationally efficient. We implemented the YOLO-tiny version, which has 24 layers, and most are convolutional and max-pooling layers. In YOLO, we can adjust the detection threshold: when we set it to a specific value, the network only displays objects detected with a confidence score above that value. So, we can build a rigid detection model which only detects objects with a high detection probability or a permeable model that detects objects with a low confidence score. In the first case, we could miss a vessel in the image, and in the second case, the network can identify ocean regions as a ship. In this first stage, it is necessary to have a high recall. Even if we get some false positives, there is no problem because, in the segmentation stage, we will filter them. Therefore, regions with waves or sun glare are mostly like to be in the region proposals. These areas can be easily confused as a ship.

Dataset	Recall	True Positive	False Negative	False Positive
kaggle	0,9	63	7	504
Seagull	0,86	115	19	1233

Table 1: Detection network results with a threshold=0,001.

Label	Dataset	N^{o}BB	IoU
From dataset	Kaggle	382	0,94
Our	Kaggle	386	0,89
Our	Seagull	494	0,91

Table 2: Segmentation results with encoder densenet121.

Next, we need to identify the ship using segmentation. Thus, we process an image to identify a class for each image pixel. In our case, there are two labels: background and ship. Following that, we find a cluster that represents the vessel, and we can get the target size and shape. We use the U-Net to perform semantic segmentation because it is efficient and produces good results with few training samples. In image segmentation, we have to convert the feature map into a vector to classify the pixels and then reconstruct the image from this vector. U-net uses the same feature maps that are used for the contraction to expand the vector to a segmented frame. Then, we preserve the structural integrity of the image and decrease the output distortion. Networks like U-net have an encoder-decoder architecture. For the encoder, we will use different architectures and study how it affects performance. Sometimes we have multiple BBs per image, and we need to group them to reconstruct the full segmented mask. We use the BB mask aggregator module.

4 Experiments

Since we are using deep learning methods, we need to train both networks with samples. So, we used RGB images from the Seagull and Kaggle dataset, [10] and [11], respectively. Once we have two networks, we divided the training phase into two steps. First, we trained the YOLO network with images from both datasets, 45000 each. Then, we passed a set of 20000 images through YOLO, and we used the BBs results to train the U-net. For both networks, we split the set in 70%-30% for training and validation. Plus, for segmentation, we tried multiple encoders, like ResNet, Densenet121, and Inceptionresnetv2, all pre-trained on ImageNet. Additionally, we tested various loss functions, focal, dice, and cross-entropy losses, also varying the batch size.

We trained and tested both networks using a Keras implementation with Tensorflow, running on an Nvidia GTX1070 Ti. As a test set, we chose images from both datasets, 575 frames. Since in the Seagull dataset, the label is a BBs, we had to segment each image manually. Furthermore, we tested our cascade model in the Airbus Ship detection challenge [11]. We submitted the segmentation results for a set of images, then based on the f2-score and IoU metrics, we got a Kaggle score.

5 Results

To evaluate our method, we tested both networks. For the detection stage, we searched for the best threshold value, with th=[0,00001; 0.5]. Table 1 shows the accuracy and recall for the best threshold, th=0,001. Despite a high number of BBs provided by YOLO, we remove between 93-95% of the background image, which turns the next stage quicker. To evaluate the segmentation task, we passed through the U-net the BBs from the detection network. Table 2 shows the IoU result for three test sets with densenet121 as an encoder, batch size of 16, and a combination of focal and weighted dice losses. Plus, we submitted our solution to the Kaggle challenge, and we compared it with full image segmentation. Table 3 displays the difference between the two methods, and the cascade model has a better score in less time. According to [11], the best method achieved a score of 0.85, and our method got a lower score, 0,82. However, we decreased the processing time per image by reducing the search space in the segmentation stage. The proposed model provides real-time ship segmentation, by achieving an average processing rate of 10 images second.

Method	Kaggle score	Time/image [s]
Full image segmentation	0,71	1,47
Detection + Segmentation	0,82	0,1

Table 3: Airbus kaggle challenge results.

6 Conclusions

This paper presents a contribution to performing fast and accurate maritime ship segmentation. This method has two main stages, in the first part, we extract possible ship location regions, and then we segment these regions to identify the ship. The initial stage allows removing unnecessary parts of the image to identify the target, which speeds up the segmentation part. In aerial images, a ship is a small target, and it is easily confused with the background. With this cascade model, we decrease detection failures and improve the segmentation mask. We tested our method on the Airbus Ship detection challenge, and we achieved a score of 0,82. The best results got a score of 0,85 in the challenge. However, our cascade model is capable of real-time detection since, on average, an image is processed in 0,1 seconds.

Acknowledgements

This work was supported by FCT with the LARSyS - FCT Project UIDB/50009/2020 and project VOAMAIS (02/SAICT/2017/31172).

References

[1] Direção-Geral de Recursos Naturais, Segurança e Serviços Marítimos. Zonas marítimas sob soberania e ou jurisdição portuguesa. https://www.dgrm.mm.gov.pt/am-ec-zonas-maritimas-sob-jurisdicao-ou-soberania-nacional, n.d. Accessed: 2020-09-7.

[2] Gonzalo Pajares. Overview and current status of remote sensing applications based on unmanned aerial vehicles (uavs). *Photogrammetric Engineering and Remote Sensing*, 81(4):281 – 329, 2015.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[4] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[5] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015.

[7] G. Cruz and A. Bernardino. Learning temporal features for detection on maritime airborne video sequences using convolutional lstm. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):6565–6576, 2019.

[8] J. Matos, A. Bernardino, and R. Ribeiro. Robust tracking of vessels in oceanographic airborne images. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–10, 2016.

[9] G. Cruz and A. Bernardino. Evaluating aerial vessel detector in multiple maritime surveillance scenarios. In *OCEANS 2017 - Anchorage*, pages 1–9, 2017.

[10] R. Ribeiro, G. Cruz, J. Matos, and A. Bernardino. A data set for airborne maritime surveillance environments. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(9):2720–2732, 2019.

[11] Airbus. Airbus ship detection challenge. `https://www.kaggle.com/c/airbus-ship-detection/overview`. Accessed: 2020-09-5.