Ship Detection in Synthetic Aperture Radar Images with Deep Learning

Arda Kaşıkçı, Ömer Batuhan Özbay
Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi, 34220 Istanbul, Türkiye
{arda.kasikci, batuhan.ozbay}@yildiz.edu.tr

***Özetçe* —Sentetik açıklıklı radar (SAR), günümüzde uzaktan algılamada kullanılan en önemli aktif görüntüleme sistemlerinden biridir. SAR, bulutlardan, gece ve gündüzden bağımsız olarak çalıştığından gemi tespitinde kullanılabilecek en iyi görüntüleri vermektedir.**

Bu makalede, Sentinel-1 ve Gaofen-3 uydularından alınan ve yaklaşık 40.000 görüntüden oluşturulmuş SAR veri setinin yapısı anlatılmaya çalışılmıştır. Veri setinden gemi tespiti yapılabilmesi için uygun olan derin öğrenme algoritmaları belirlenmiştir. Bu veri setinden gemi tespiti için ilk olarak YOLO algoritmasının farklı versiyonları (v3Tiny, v4, v5) kullanılmış olup bu algoritmalar kendileri arasında AP, AR ve tespit süresi gibi belirli parametreler kullanılarak hız ve performans açısından kıyaslanmıştır. Bu kıyaslamalara SSD-MobileNet, EfficientDet D0 ve Faster R-CNN algoritmaları da eklenip gemi tespiti için en uygun algoritma bulunmaya çalışılmıştır.

Anahtar Kelimeler—gemi tespiti, derin öğrenme, sentetik açıklıklı radar, SAR, nesne tespiti.

***Abstract*—Nowadays, synthetic aperture radar (SAR) is one of the most important active imaging systems in remote sensing. Since SAR is not affected by clouds, day and night, it gives the best images that can be used in ship detection.**

In this paper, the structure of the SAR dataset, which was obtained from Sentinel-1 and Gaofen-3 satellites and composed of approximately 40,000 images, was tried to be explained. Deep learning algorithms fitting for ship detection from the dataset have been decided. First, different versions of the YOLO algorithm (v3Tiny, v4, v5) were used for ship detection. Then these algorithms were compared among themselves in terms of speed and performance using certain parameters like average precision (AP), average recall (AR) and detection time. SSD-MobileNet, EfficientDet D0 and Faster R-CNN algorithms were added to these comparisons and the most suitable algorithm for ship detection was tried to be found.

Keywords—ship detection, deep learning, synthetic aperture radar, SAR, object detection.

I. Introduction

Radars capture targets by actively sending microwaves to the surface, thus leading to continuous imaging. SAR, which is a type of radar, is best suited for ship detection because its resolution is stable even when it is far from the observed targets [1], [2].

SAR is an active remote sensor, so it carries its own illumination and it is not dependent on sunlight. This feature makes it functional in all-weather and day-and-night operating conditions [3]. SAR sensors can generate large amounts of data in a short time. We need this data for automatic detection of targets of interest. These data can be used especially as a field of study for the detection of ships in the seas and oceans, many of which include open field images [4]. Ship detection is essential for marine surveillance in areas such as oil spill detection, illegal fishing, management of maritime traffic and maritime piracy [5].

Two types of methods are used for ship detection. The first one was the traditional methods used in the past. These methods are mainly based on continuous false alarm (CFAR) and the general approach of this method is to distinguish between land and ocean in images [3], [6]. This approach causes a reduction in the detection speed of ship [7], [8]. The other method, deep learning, is to replace pre-trained object detection tools for ship detection and train them with SAR images [5].

In this paper, we tried to find the most suitable deep learning method. The most important metric in this study is detection time, since the goal of this project was to find the most suitable deep learning method for real-time ship detection. Other metrics that were used in this paper are explained in later parts of this paper.

II. Dataset

The dataset used in the project is the dataset in the article "A SAR Dataset of Ship Detection for Deep Learning under Complex Backgrounds" [5]. Dataset is publicly shared on the Github page. In the dataset there are 59.535 ships from 102 Gaofen-3 images and 108 Sentinel-1 images. These images are cropped to 39,729 images and these images are 256 x 256 pixels. The images from the dataset are varied in resolution, polarization, incident angle and background.

In Figure 1, there are some labeled sample images from the dataset.

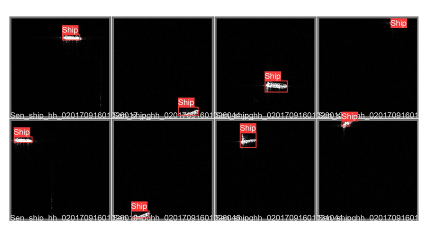

Figure 1 Some examples of images from the dataset

Images from the dataset are labeled by SAR experts. These labels are in YOLO format. There is a txt file for every single image in this format and there are five values for every single object in a single image in that txt file. These values, in order: class number, center of the object on x axis (x), center of the object on y axis (y), width (w) and height (h). Since there is only one class in this dataset, which is ship, class number is always zero.

Figure 2 A ship image and its labels in YOLO format

As seen in Figure 2, there are 3 ships in the image and their class number, x, y, w and h values are written. It is important part of the format that these values are normalized, it should be noted the image's upper left corner is (0,0) and bottom right corner is (1,1).

While the dataset is being trained,it is split into 73% into training and 27% into test dataset. 28933 images were used for training and 10796 images were used for testing. Normally, it is thought that images should be split between Gaofen and Sentinel satellites, too, but in the dataset there are some images named like "newshipxxxxxxx" and "shipxxxxxxxx". It is unknown which satellite produced these images, so we did not split into training and testing by satellites. It is done fully randomly.

III. Experimental Results

Training and testing for ship detection were performed on Google Colab with Tesla P100-PCIE-16GB GPU, Intel(R) Xeon(R) CPU and 13GB RAM. It should be noted that to re-create this study, it is necessary to buy some cloud storage from Google. All hyperparameters used in YOLO methods are given in Table 1. Since the other methods have different hyperparameters, they are given in Table 2.

A. Evaluation Criteria

To compare the methods that we used to train the dataset, we used 14 different performance criteria. Most of these metrics are based on Precision and Recall. To understand the metrics used in this study, Precision and Recall must be understood.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

As seen, Precision is calculated as the number of true positives (TP) divided by the total number of true positives and false positives (FP). Precision gives the ratio of correct positive predictions out of all positive predictions. Recall is calculated as the number of TP divided by the total number of TP and false negatives (FN). Recall gives the ratio of correct positive predictions out of actual positives. Ideally both Precision and Recall are expected to be 1. Since it is not possible for both FP and FN to be 0, there is a necessity for a new metric that takes these values into account.

Mean Average Precision (mAP) is one of the most popular object detection metrics, which became a standard in object detection, is used in this study. Mean Average Precision is mean of the AP for all the objects that should be detected. So, to understand mAP, AP should be explained first. AP can be calculated with equation given down below. Since in this study there is only ship detection, which means there is only one class, AP equals mAP.

$$\mathrm{AP} = \sum_{n} (R_n - R_{n-1}) P_n \tag{3}$$

In this study, different AP metrics are calculated. AP,AP^{50},AP^{75}, which are calculated with different intersection over union (IoU) values like [50%:5%:75%], 50% and 75%, are used in this study as well. In addition to these metrics, AP^S,AP^M,AP^L metrics, which show the AP of ships that are small, medium and large sizes, are also used to compare. These metrics are calculated in COCO evaluation metrics. AP per class and mAP metrics are evaluated as well, as required in the PASCAL challenge [9].

Average recall (AR) can be calculated with the equation given below.

$$\mathrm{AR} = 2 \int_{0.5}^{1} R_{IoU}(o)\, do \tag{4}$$

In this equation, $R_{IoU}(o)$ is a function that returns the recall for a given IoU. Just like AP, there are different variations of AR that are used in this study. AR^S,AR^M,AR^L are used to just like their AP counterparts, which does the same job just for average recall. There are 3 other AR variations that are used in this study, which are AR^1,AR^{10} and AR^{100}. AR^1 metric take into account just one detection per image, AR^{10} and AR^{100} metrics do the same, but 10 and 100 detection per image. These are the 13 detection metrics that are used in this study, but the most important one for us is detection time. Whole test dataset is used to calculate detection time, as usual. Both time and evaluation metrics are taken into account to make the performance analysis.

Table 1 and 2 shows hyperparameters that used to train the dataset for ship detection. Figure 3 shows some complicated examples of images for ship detection.

Table 1 Hyperparameters used in YOLO methods

Parameters	YOLOv3Tiny	YOLOv4	YOLOv5
Batch Size	64	16	32
Momentum	0.9	0.949	0.937
Max Batches	2000	2000	30(Epoch)
Learning Rate	0.001	0.001	0.01

Table 2 Hyperparameters used in EfficientDet D0, SSD-MobileNet V2 and Faster R-CNN

Parameters	EfficientDet D0	SSD-MobileNet V2	Faster R-CNN
Batch Size	8	16	2
Number of Steps	30000	30000	30000
Learning Rate Base	0.01	0.01	0.05
Warmup Learning Rate	0.0001	0.0001	0.0001
Warmup Steps	2000	2000	2000
Momentum Optimizer	Cosine Decay	Cosine Decay	Cosine Decay
Momentum Value	0.9	0.9	0.9

Table 3 Performance comparison for YOLO variations

Model	mAP	Detection Time (ms)
Yolov3Tiny	0.7518	85
Yolov4	0.85	22
Yolov5	0.92	50

Table 4 Faster R-CNN, SSD-MobileNet V2, EfficientDet-D0 and YOLOv5 comparison table

Metrics	Faster R-CNN	SSD-MobileNet V2	EfficientDet-D0	YOLOv5
mAP(%)	80.03	86.98	90.49	**92.00**
Detection Time (sn)	0.09	**0.017**	0.04	0.05
AP (%)	35.18	40.02	46.40	**52.82**
AP^{50}(%)	78.58	85.89	89.63	**90.91**
AP^{75} (%)	25.35	30.67	42.44	**55.68**
AP^{S} (%)	32.39	36.31	43.48	**49.70**
AP^{M} (%)	39.39	45.04	50.60	**57.81**
AP^{L} (%)	16.30	44.42	49.56	**66.66**
AR^{1} (%)	36.58	40.24	43.37	**48.06**
AR^{10} (%)	44.69	50.81	54.50	**60.88**
AR^{100} (%)	44.71	50.94	54.71	**61.17**
AR^{S} (%)	41.99	47.08	51.22	**57.12**
AR^{M} (%)	48.76	56.32	59.61	**66.80**
AR^{L} (%)	3.67	62.67	61.00	**71.67**

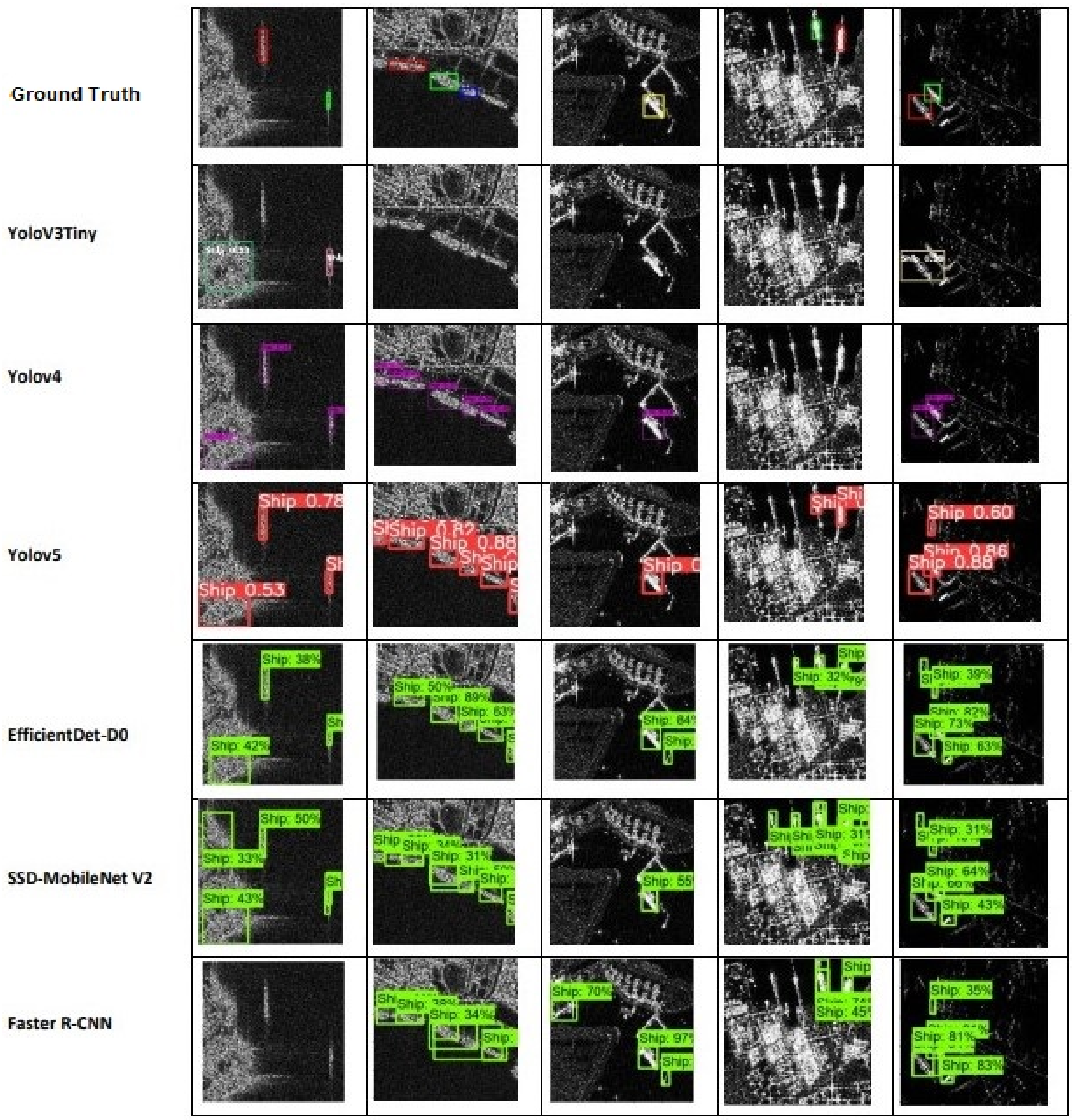

Figure 3 Comparison of example ship detections for every used methods

IV. Performance Analysis

The metrics used for performance analysis were obtained with the object detection tool [9], which makes these metric calculations much easier.

As seen from Figure 3, some complex images were picked to show some examples. As expected Yolov3Tiny shows least promise. EfficientDet, SSD-MobileNet and Faster R-CNN predicted some ships that don't exist. It can be said that Yolov5 is better at ship detection in complex backgrounds.

It can be seen from Table 3 that Yolov5 performs best as it seems from mAP metric. Even though Yolov4 is the fastest, it is decided that Yolov5 should be used for the comparison with the rest of the methods.

It should be noted that, even though the hyperparameters that was used in this paper were not equal, it does not mean that it was not a fair study. We firmly believe that the working environment should be a critera as well. EfficientDet D0 and Faster R-CNN were tried to train with the same batch size as SSD-MobileNet v2 but it was pushing the working environment much harder than expected. The batch size was lowered in this study because of this problem. The batch sizes that were given in Table 2 were actually the biggest batch sizes that could given.

Table 4 shows that the most successfull method was YOLOv5 based on the most popular metric, mAP. EfficientDet-D0 was pretty close, but it was detecting ships a little bit faster. The biggest difference makers in these two methods were AP^L and AR^L metrics which show the ability to detect the larger ships. YOLOv5 is actually much better at detecting large ships. The other difference maker metric is AP^{75} which shows us the AP metric calculated for IoU values greater than 0.75. This means YOLO draws ship boundries better than EfficientDet.

The fastest method in this study is SSD-MobileNet V2, fitting the purpose of design. SSD-MobileNet v2 detects two times faster than its closest competitor.

Another point that worth noting is that Faster R-CNN performs poorly in comparison. It's much more slower than the other methods, and has really low success rate at detecting large ships as it can be seen from AP^L and AR^L metrics. We think the reason for that because of the low batch size, it was not trained as well as the other methods.

V. Conclusion

In this study, we tried to detect ships fast and accurately with the use of ship data set. For this purpose, different versions of YOLO models (v3Tiny, v4, v5), Faster R-CNN, SSD-MobileNet v2 and EfficientDet-D0 models were trained and evaluated by comparing test results.

After doing the performance analysis, we came to the conclusion that YOLOv5 model should be used, considering the environment in which we provide the hardware information for ship detection. In addition to that, we think that EfficientDet-D0 will be a good alternative to YOLO in cases where the use of TenserFlow libraries may be a necessity.

References

[1] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geoscience and remote sensing magazine*, vol. 1, no. 1, pp. 6–43, 2013.

[2] C. Oliver and S. Quegan, *Understanding synthetic aperture radar images*. SciTech Publishing, 2004.

[3] K. El-Darymli, E. W. Gill, P. Mcguire, D. Power, and C. Moloney, "Automatic target recognition in synthetic aperture radar imagery: A state-of-the-art review," *IEEE access*, vol. 4, pp. 6014–6058, 2016.

[4] D. J. Crisp, "The state-of-the-art in ship detection in synthetic aperture radar imagery," Defence Science and Technology Organisation Salisbury (Australia) Info . . . , Tech. Rep., 2004.

[5] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, "A sar dataset of ship detection for deep learning under complex backgrounds," *remote sensing*, vol. 11, no. 7, p. 765, 2019.

[6] K. El-Darymli, P. McGuire, D. Power, and C. R. Moloney, "Target detection in synthetic aperture radar imagery: A state-of-the-art survey," *Journal of Applied Remote Sensing*, vol. 7, no. 1, p. 071598, 2013.

[7] U. Kanjir, H. Greidanus, and K. Oštir, "Vessel detection and classification from spaceborne optical images: A literature survey," *Remote sensing of environment*, vol. 207, pp. 1–26, 2018.

[8] K. Ouchi, "Current status on vessel detection and classification by synthetic aperture radar for maritime security and safety," in *Proceedings of the 38th Symposium on Remote Sensing for Environmental Sciences, Gamagori, Aichi, Japan*, 2016, pp. 3–5.

[9] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 3, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/3/279