A Deep Convolutional Neural Network Applied to Ship Detection and Classification

Conference Paper · November 2020

DOI: 10.17648/sobena-2020-122852

6 authors, including:


Mikael Proença Strauhs
Federal University of Rio de Janeiro
1 PUBLICATION 0 CITATIONS
SEE PROFILE


Antonio Rafael Paulino de Lira
Federal University of Rio de Janeiro
1 PUBLICATION 0 CITATIONS
SEE PROFILE


Ramiro Fernandes Ramos
Federal University of Rio de Janeiro
6 PUBLICATIONS 0 CITATIONS
SEE PROFILE


João Vitor Marques de Oliveira Moita
Federal University of Rio de Janeiro
10 PUBLICATIONS 7 CITATIONS
SEE PROFILE

28th International Congress on Waterborne Transportation, Shipbuilding and Offshore Constructions

Rio de Janeiro/RJ, 27th to 29th October 2020

A deep convolutional neural network applied to ship detection and classification

Mikael Strauhs, UFRJ, Rio de Janeiro/Brasil, mikaelstrauhs@poli.ufrj.br
Antonio Rafael Paulino de Lira, UFRJ, Rio de Janeiro/Brasil, rafaellira@poli.ufrj.br
Ramiro Fernandes Ramos, UFRJ/COPPE, Rio de Janeiro/Brasil, ramirofr@oceanica.ufrj.br
João Vitor Marques de Oliveira Moita, UFRJ/COPPE, Rio de Janeiro/Brasil, joaov@oceanica.ufrj.br
Severino Virgínio Martins Neto, UFRJ, Rio de Janeiro/Brasil, severino_virginio@poli.ufrj.br
Jean-David Caprace, UFRJ/COPPE, Rio de Janeiro/Brasil, jdcaprace@oceanica.ufrj.br

Abstract
The task of detecting a ship is relevant for a variety of applications in both military and civilian fields, from maritime traffic surveillance to sea pollution monitoring. Despite the recent significant attention, deep learning-based object detection algorithms have received, they are still rarely applied in the detection of ships. Expectedly, *we see even fewer applications when the goal is to detect specific ship classes, although those could deliver extra valuable information. In this work, we build a ship detection algorithm capable of distinguishing six common ship types: ore carrier, bulk cargo carrier, general cargo ship, container ship, fishing boat and passenger ship. To achieve this objective, we firstly built a deep convolutional neural network in python, taking advantage of the TensorFlow framework. Secondly, we analyzed our model's ability to generalize on a new set of images, apart from the training and testing sets. The results showed that our proposed model is close to state-of-the-art performance since it was able to perform well on the test set - mAP = 97.62%. There is still room for improvement on the model robustness, associated primarily with possible training set limitations. In practice, this paper will contribute to the advance of research and applications on ship detection.*

1. Introduction

Ship detection plays an essential role in monitoring and managing marine traffics, helping prevent or act against various illegal activities such as smuggling, dumping of pollutants and illegal fishing. Automated Identification System (AIS) fulfills the task of detecting any vessel with a connected VHF transponder aboard, as demonstrated in (*Ramos et. al*, 2019), but fails to detect ships that either sail with disconnected transponders or are not legally required to install a transponder. Satellite, radar, infrared or visible light images can be alternatively monitored to accomplish the same task. More efficiently and less costly, the same data can be used to train machine learning models, e.g. neural networks.

Despite the recent significant attention, deep learning-based object detection algorithms have received, they are still rarely applied in the detection of ships. Also, we see even fewer applications when the goal is to detect specific ship classes. The main reason behind that statement is the cost of computational power and high-quality data. Modern deep learning models usually require several optimized Graphics Processing Units (GPUs) for training in order to accomplish convincing results, which is not affordable for many people. We often face a similar issue with trainable data, where large-scale publicly available ship datasets are still a shortcoming, especially when we desire ship classes to be individually annotated.

Rosenblatt et. al (1957) introduce the perceptron concept, establishing its technical and economic feasibility, and allowing for the whole field of neural networks and deep learning to be developed.

Krizhevsky et. al (2012) present a Deep Convolutional Neural Network able to unprecedently outperform traditional methods in ImageNet Computer Vision competition, one of the most convincing results responsible for a shift in deep learning.

Bochkovskiy et al. (2020) address computational power problems through creating a Convolutional Neural Network (CNN) that operates in real-time on a conventional GPU, and for which training requires only one conventional GPU, while achieving state-of-the-art performance.

Shao et al. (2018) introduce a high-quality ship dataset, which covers six-ship types and ensures variations of visible proportion, scale, viewpoint, illumination, background and occlusion.

Chang et. al (2019) face ship detection problem with an interesting deep learning approach, where Synthetic Aperture Radar (SAR) imagery is used as input data to a You Only Look Once (YOLO) v2 modified model.

Zhao et. al (2019) develop coupled Convolutional Neural Networks (CNNs) for small and densely clustered Synthetic Aperture Radar (SAR) ship detection targets, consisting of a ship proposal network allied with a ship discrimination network.

In this article we built a ship detector, starting from the selection of a pre-trained deep convolutional neural network (*Bochkovskiy et al*., 2020) and a publicly available dataset. Thereafter, using a single conventional GPU, we further train the model to detect the corresponding ship classes. Finally, we aim to evaluate our model and discuss results in two different levels: How well it detects each ship class on images similar to the ones seen on training; How well it detects each ship class on a new set of images, which will be selected from the internet and annotated by us.

The remainder of this paper is organized as follows. The data and methodology are described and analyzed in detail in Section 2. The results on both test sets, including evaluation metrics, detection samples and appropriate discussion are presented in Section 3. We conclude this paper in Section 4.

2. Data and Methodology

In this paper, annotated images are used to train and test a deep convolutional neural network. In this section, the data and the methodology applied to this data are presented.

The Convolutional Neural network interprets images in a way that resembles human capacity, an analogy can be made with the human brain, which has millions of cells that serve only to make us see. They are divided into two types, cones and rods, the first is responsible for the colors red, green and blue, and the second for the shades of gray. For an image to be seen, it needs to stimulate millions of these two types of cells that will send signals with different frequencies according to the stimulus. Receiving millions of electrical signals, the human brain superimposes and organizes this data forming an image, but the image alone is not enough to understand it, the human brain also has training focused on interpreters that happen since childhood. Seeking to carry out this visualization process, CNN works by inverting the process, an image passes through several filters, it is as if the image formed by the brain were transformed again into signals and after they are treated, we managed to reach the source of stimulus and in a way that it differs from the human brain, it is possible to recognize the core by it and based on the information we have passed this structure is now able to classify, locate, detect and segment.

2.1. Data

The data used to train and test our model is a subset of the SeaShips dataset (*Shao et al.*, 2018). It consists of 7000 images having a resolution of 1920x1080 pixels and 9221 annotations of six ship types: ore carrier, bulk cargo carrier, general cargo ship, container ship, fishing boat and passenger ship.

The choice of this data set was based on the standardization of the size of the photos, adequate resolution, variety of types of ships and the objects were already demarcated. Although the format of the labels was not in the proper format, this was easily circumvented. The set of these factors reinforced the choice of the dataset for training the neural network.

In order to build the dataset, the authors selected the images from monitoring cameras in a deployed

coastline video surveillance system in Zhuhai City, China.
Table 1 lists the number of images that contain each ship category in the dataset. Note that the sum of the values on the Images column exceeds the number of images because some images contain more than one ship category and the same applies to the Percentage column.

Table 1 – Number of images that contain each ship category

Ship Category	Images	Percentage
Ore carrier	2084	29.74%
Fishing boat	1539	21.98%
Bulk cargo carrier	1811	25.87%
General cargo ship	1426	20.37%
Container ship	898	12.83%
Passenger ship	455	6.50%
Total	7000	100%

In every image, the number of annotations ranges from one to five, although in 75% of the images there is only one bounding box and in almost 95% of the images there are either one or two annotations, as listed with more detail in Table 2.

Table 2 – Number of images of each ship count

Ship Count	Images	Percentage
1	5254	75.06%
2	1361	19.44%
3	303	4.33%
4	74	0.01%
5	8	0.001%

These statistics help us better understand the input data, which is essential for both training the model and analyzing its performance.
Moving forward, we randomly selected 70% of the images as our training set and 30% as the testing set.
Figure 1 is a sample image from the training set. It contains three annotated ships from the classes container ship, bulk cargo carrier and ore carrier.

Figure 1 – SeaShips (*Chao et. al*, 2020) image sample

Furthermore, we built a relatively small dataset, consisting of 118 images from different sizes to be used as our second testing set. The images were selected from Google - using the International Maritime Organization registration code, consisting of three letters and seven numbers, according to the types of ships in our dataset - and precisely annotated by us using a labeling software. Figure 2 is a sample from our dataset.
It is important to note that while all images from SeaShips are photos taken in the same geographic Chinese region, the images from our dataset are photos taken all over the world.

Figure 2 – Image sample from our dataset

2.2. Model

The model used in this paper is a pre-trained YOLOv4. You Only Look Once (YOLO) is a family of end-to-end object detection algorithms, i.e they predict the class and position of the object directly by a single network.

The main justification behind our choice of model is the fact that YOLOv4 requires only a single conventional Graphics Processing Unit (GPU) for training, while still providing state-of-the-art speed and accuracy.

Before proposing YOLOv4*, Bochkovskiy et al.* (2020) describe the modern ordinary object detector as consisting of two parts: Backbone which is pre-trained on ImageNet and Head which is used to predict classes and bounding boxes.

For the YOLOv4 backbone and head the authors chose with proper theoretical justifications CSPDarknet53 (*Wang* et. al., 2020) and YOLOv3 (*Redmon* et. al., 2018), respectively.

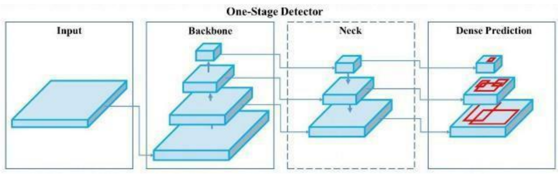

Figure 3 – One-stage detector architecture (*Bochkovskiy et al.*, 2020)

Figure 3 shows YOLOv4 architecture as a one-stage detector. PANet (*Liu et. al.*, 2018) and an SPP (*He et. al, 2015*) module were added between the backbone and head as the Neck part which mainly collects feature maps from different stages.

YOLOv4 has new features that the other applications which improve Convolutional Neural Networks (CNNs) do not use. In its construction, the developers assume the basics necessary characteristics that involved the following features: Weighted-Residual-Connections (WRC)(*Shen et al.*, 2016) analyzes and interacts with the layers Residue, seeking out for better results; Cross-Stage-Partial connection (CSP)(*Wang et al.*, 2019) works on optimizing the computations; Cross-Iteration Batch Normalization (CmBN)(*Yao et al.*, 2020) allows small batches to give a better result on the training iteration with a Taylor polynomial application; Self-adversarial-training (SAT)(*Chen et al.*, 2020) and Mish-Activation (*Misra et al.*, 2020) as a crucial function that improves the neural networks. In addition, there is increased Drop Block (*Ghiasi et al.*, 2018) regularization (a model that randomly eliminates nodes during the training, which has an effect of simulating various network architectures), CloU loss and Mosaic data augmentation (use the training data to generate other situation with the data, some examples are distortion, cutout and grid mask).

The model was pre-trained in MS COCO dataset (*Lin et. al.*, 2014) to detect 80 different classes and the resulting weights were the starting point for our ship detector. It is relevant to note that the only ship target among the classes is "boat".

Our goal was to first use our data to train the model to detect our six ships' classes rather than the eighty COCO classes. Secondly, analyze the final model's ability to generalize on images it had never seen before.

3. Results and Discussion

This section presents the results from using the training and testing data described in section 2.1 as input to our ship detector described in section 2.2

3.1. Mean Average Precision (mAP)

Once the training process of a deep convolutional neural network or any other machine learning model is over, it is naturally convenient to evaluate the model somehow.

Table 3 – mAP Results for IoU threshold = 50%

Test Set	mAP	Ore carrier	Bulk cargo carrier	General cargo ship	Container ship	Fishing boat	Passenger ship
SeaShips	0.9762	0.9670	0.9743	0.9855	0.9956	0.9820	0.9525
Ours	0.3558	0.1637	0.4447	0.2000	0.4242	0.4445	0.4576

Before doing that, a previous step is to understand the model's output. In our case, the detector receives an image or video as input and draws bounding boxes around whatever it identifies as one of the six ships' classes. Along with the bounding boxes coordinates, our detector outputs a confidence value ranging from zero to one. The closer to one, the more certain the network is of the given detection.

Over the years several metrics have been developed, shared and tested by many data scientists.

In that context, Mean Average Precision (mAP) has become the accepted way of evaluating the main object detection competitions such as for the COCO challenge.

The mAP is the mean value of the Average Precision (AP) across all classes. *Everingham et. al.* (2010) determines the AP summarizes the shape of the precision/recall curve and it is calculated for each class of the detector, by taking the mean of the detector's output precision value across a set of equally spaced recall values.

In this paper, we consider a single Intersection over Union (IoU) threshold of 0.5, i.e., objects predicted by the model must overlap at least 50% the corresponding ground truth object in order to be considered a correct detection.

Table 3 shows our detector's performance on both test sets described in section 2.1, in terms of mAP.

The following sections cover in more detail the results on each set.

3.2. SeaShips Test Set

In this section, we take a closer look at the performance of our detector on the SeaShips test set.

Each False Positive detection indicates that the model drew a bounding box in a region that either does not encompass a ship or encompass a ship from a different class.

Each False Negative, on the other hand, indicates that the model missed a ship that was annotated.

Table 4 lists the number of ground-truth bounding boxes (GT BB), False Negatives (FN) and False Positives (FP) for a confidence threshold of 0.25, that is, all detections with a confidence score under 0.25 are not considered.

Table 4 – FP/FN count for confidence threshold = 0.25

Ship Category	GT BB	FN	FP
Ore carrier	666	40	48
Fishing boat	686	33	42
Bulk cargo carrier	609	28	37
General cargo ship	440	14	18
Container ship	257	2	15
Passenger ship	150	8	12

Note that only 2 of the 257 container ships were missed by the model on the set confidence threshold. The category with the highest portion of ships missed by the model was the Ore carrier (6%). Besides, from the total number of bounding boxes drawn by the model, only 6% were false positives (172).

The relatively low FP and FN count translate not only into high precision values, as shown in Table 3, but also into high recall values.

3.2.1. Detection samples

Figures 4-11 are SeaShips (*Shao et. al*, 2020) images, modified by us through our detector in the form of drawn bounding boxes.

Figure 4 – Bulk Cargo Carrier detected through deep learning with a confidence score of 0,99

Figure 5 – Ore Carrier detected through deep learning with a confidence score of 1,00

Figure 6 – Container ship detected through deep learning with a confidence score of 1,00

Figures 4, 5 and 6 are three examples of correctly classified ships and precisely drawn bounding boxes, with very high confidence scores: 0.99, 1.0 and 1.0, respectively.

Figure 7 – Fishing boat detected through deep learning with a confidence score of 0,91

Figure 8 – Fishing boat detected through deep learning with a confidence score of 0,87

Figures 7 and 8 illustrate our model's ability to correctly detect and classify very small objects, which is a known challenge in object detection.

Figure 9 – Ore carriers under occlusion detected through deep learning. The ship on the left has a confidence score of 0,98, the ship in the middle has 0,82 and the ship on the right has 0,87.

Figure 10 – Ore carrier has a confidence score of 0,91 and Bulk cargo carrier has a confidence score of 0,87 under occlusion detections

Figures 9 and 10 illustrate how well our detector deals with another object detection challenge: occlusion. All five ships are correctly localized and classified, with over 0.8 confidence score.

Figure 11 – Container ship at dusk detected through deep learning with a confidence score of 0,99

Another challenge upon which our model is able to achieve good results in different light levels. Figure 11 shows a correctly classified and localized container ship at nighttime.

3.3. The test set developed in this study

In this section, we take a closer look at the performance of our detector on our test set.

Figures 4 to 11 from the previous section give an excellent idea of the general characteristics from the ships and background present in the dataset used to train our model. Most ships share peculiar designs, quite common in the area the photos were taken but rarely seen worldwide. Ore carriers for example load the ore on deck, exposed.
This regional factor from the training data affects the model’s ability to generalize on ordinary ships. We built a test set precisely to evaluate that robustness and the results are presented in this section.

3.3.1. Detection samples

Figure 12 – Bulk cargo carrier detected through deep learning with a confidence score of 0,88

Figure 12 includes a bulk cargo carrier and two tugs. The ore carrier was correctly classified and satisfactorily localized, although the bow and stern of the ship are not completely inside the bounding box.
Since our detector was not trained to detect tugboats, both were detected as fishing boats, which is not ideal but would be expected by the similarity of the classes.
At the back of the image, two ships were incorrectly detected as one fishing boat. However, given the size and resolution of these ships, even humans would struggle to correctly detect them.

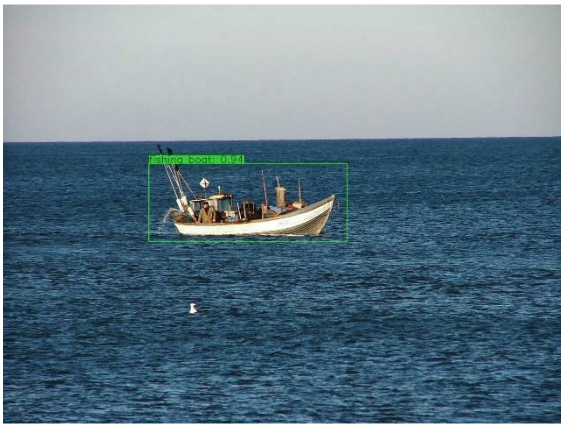

Figure 13 – Fishing boat detected through deep learning with a confidence score of 0,94

Figure 14 – General cargo ship detected through deep learning with a confidence score of 0,98

Figure 15 – Container ship detected through deep learning with a confidence score of 0,34

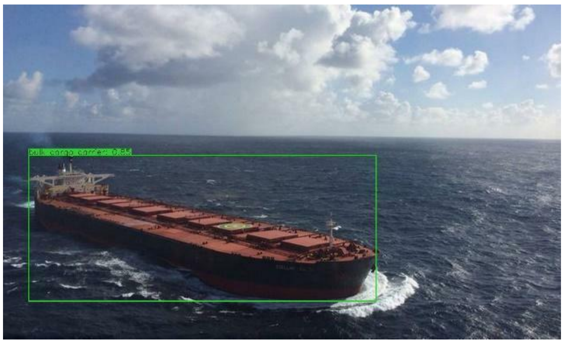

Figure 16 – Bulk cargo carrier detected through deep learning with a confidence score of 0,85

Figures 13 to 16 show fishing boat, general cargo ship, container ship and bulk cargo carrier detections, respectively. Despite the low lighting level in Figure 15, our model was able to detect the correct class with 0.34 confidence score.

Figure 17 – Passenger ship detected through deep learning with a confidence score of 0,29

Figure 18 – Passenger ship detected through deep learning with a confidence score of 0,92

Figures 17 and 18 are examples of successfully classified passenger ships. The red bounding box indicates that the confidence score is under 0.5 for Figure 17 and both bounding boxes missed some parts of the ships. Nevertheless, the fact that the model was able to collect enough features from the yachts to classify them as passenger ships shows generalization ability, given that yachts were not seen during training.

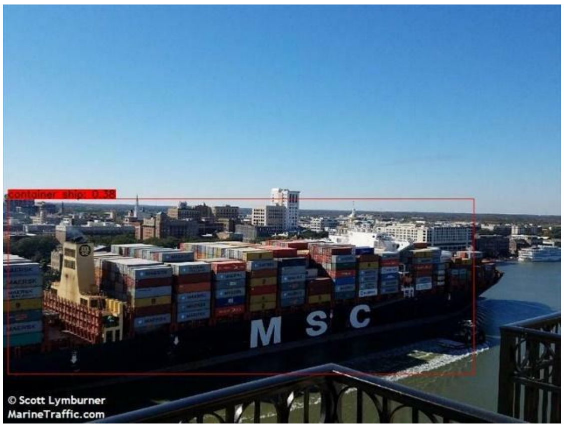

Figure 19– Container ship detected through deep learning with a confidence score of 0,38

Figure 19 is another example of an unexpected result. The structure of the ship and containers blend with the buildings in the background and the hull blends with the shadow on the water. Although the drawn bounding box misses a section of the bow and maybe included some of the background buildings, the model still classified the truncated container vessel correctly.

4. Conclusion

In this study, we verified that the proposed model is able and suitable to detect ships present on the Chinese SeaShips dataset (mAP 97.62%), although unable to achieve similarly convincing performance on the dataset we built (mAP 35.58%), as shown in Table 3.

We believe that the unique characteristics of the Chinese ships have limited the model's generalization ability. Comparing samples from the training set and our test set class by class, we quickly spot abrupt differences and therefore it is not hard to anticipate generalization problems.

Still, the model surprised us positively with some impressive detections such as the ones discussed in section 3.3.1. Also, common object detection challenges such as occlusion, lighting level and small size objects were handled quite well by our detector as shown in section 3.2.1.

In practice, this work can be very naturally applied to detect ships in real-time on the Chinese coast. Alternatively, our model could also be deployed aboard an autonomous boat to help navigation, with a few minor adjustments.

From this point forward, a next step would be to develop a new large-scale dataset, covering more ship classes such as tugboats and oil tankers, and international vessels to hopefully allow for improved robustness after training.

5. Acknowledgments

The present work was realized with the partial support of ANP, National agency for Oil, Natural Gas and Biofuels, FINEP, Financier of Studies and Projects and MCTI, Ministry of science, technology and innovation through PRH18-ANP/MCTI, ANP Program for Oil, Natural Gas and Biofuels, ongoing at DENO, Ocean Engineering Department.

6. References

BOCHKOVSKIY, A.; WANG, C.; LIAO, H. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv: 2004.10934v1, 2020.

CHANG, Y.; ANAGAW, A.; CHANG, L.; WANG, Y. C.; HSIAO, C.; LEE, W. Ship Detection Based on YOLOv2 for SAR Imagery. Remote Sens. Volume 11. page 786, 2019; https://doi.org/10.3390/rs11070786.

CHEN, K.; ZHOU, H.; CHEN, Y.; MAO, X.; LI, Y.; HE, Y.; XUE, H.; ZHANG, W.; YU, N. Self-supervised Adversarial Training. arXiv:1911.06470v2, 2020.

EVERINGHAM, M.; GOOL, L. V.; WILLIAMS, C. K. I.; WINN, J.; ZISSERMAN, A. The PASCAL Visual Object Classes (VOC) Challenge. Int J Comput Vis 88, pages 303-338, 2010.

GHIASI, G.; LIN, T.; LE, Q. V. DropBlock: A regularization method for convolutional networks. arXiv: 1810.12890v1, 2018.

HE, K.; ZHANG, X.; REN, S., SUN, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. arXiv:1406.4729v4, 2015.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON G. E. ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems, pages 1097-1105, 2012.

LIN, T.; MAIRE, M.; BELONGIE, S.; BOURDEV, L.; GIRSHICK, R., HAYS, J.; PERONA, P.; RAMANAN, D.; ZITNICK, C. L., DOLLÁR, P. Microsoft COCO: Common Objects in Context. arXiv: 1405.0312, 2014.

LIU, S.; QI., L.; QIN, H.; SHI, J.; JIA, J. Path Aggregation Network for Instance Segmentation. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, pages 8759-8768, 2018.

MISRA, D. Mish: A Self Regularized Non-Monotonic Activation Function. arXiv:1908.08681v3, 2020.

RAMOS, R. F.; CEPEDA, M. F. S.; MOITA, J. V. M. d. O. MONTEIRO, G. P.; FIKSDAHLB, O.; CAPRACE, J.-D. Detecting Possible Near Miss Collisions in Santos Bay from AIS Big Data. Rio de Janeiro, SOBENA, 2019.

REDMON, J.; FARHADI, A. YOLOv3: An Incremental Improvement. arXiv:1804.02767, 2018.

ROSENBLATT, F.; STIEBER, A.; SHATZ, R. H. The Perceptron A Perceiving and Recognizing Automaton. Cornell Aeronautical Laboratory, Buffalo, New York. Report 85-460-1, 1957.

SHAO, Z.; WU, W.; WANG, Z.; DU, W; LI, C. SeaShips: A Large-Scale Precisely Annotated Dataset for Ship Detection. IEEE Transactions on multimedia. Volume 20. Number 10, pages 1520-9210, 2018.

SHEN, F.; ZENG, G. Weighted Residuals for Very Deep Networks. arXiv:1605.08831v1, 2016.

WANG, C.; LIAO, H. M.; YEH, I.; WU, Y.; CHEN, P.; HSIEH, J. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. arXiv: 1911.11929, 2019.

WANG, C.; LIAO, H. M.; WU, Y.; CHEN, P.; HSIEH, J.; YEH, I. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop (CVPRW), Seattle, WA, USA, pages 1571-1580, 2020.

YAO, Z.; CAO, Y.; ZHENG, S.; HUANG, G.; LIN, S. Cross-Iteration Batch Normalization. arXiv:2002.05712v2, 2020.

ZHAO, J.; Guo, W.; ZHANG, Z.; WENXIAN Y. A coupled convolutional neural network for small and densely clustered ship detection in SAR images. Sci. China Inf. Sci. 62, 42301 (2019). https://doi.org/10.1007/s11432-017-9405-6.