Ships Detection on Aerial Imagery using Transfer Learning and Selective Search

Desak Ayu Sista Dewi[a1]**, Dewa Made Sri Arsa**[b2]**, Anak Agung Ngurah Hary Susila**[b3]**, I Made Oka Widyantara**[c4]

[a]Department of Industrial Engineering, Universitas Udayana
[b]Department of Information Technology, Universitas Udayana
[c]Department of Electronic Engineering, Universitas Udayana
e-mail: [1]sistadasd@unud.ac.id, [2]dewamsa@unud.ac.id, [3]harysusila@unud.ac.id, [4]oka.widyantara@unud.ac.id

Abstrak

Lalu lintas di perairan air seperti pelabuhan dan laut sangat penting untuk dilakukan pengamatan karena dapat membantu meminimalisir kecelakaan kapal laut yang tidak diinginkan. Oleh karena itu, sebuah metode pendeteksian otomatis kapal laut diusulkan dalam penelitian ini. Beberapa metode deep learning dikaji dalam penelitian ini, seperti MobileNetV2, DenseNet121, VGG16, dan ResNet50. Kemudian, metode yang sudah dilatih digunakan untuk mendeteksi kapal laut. Untuk mempercepat pendeteksian, kami menggunakan metode selective search daripada sliding window untuk mengambil sampel kandidat objek dari gambar perairan yang diberikan. Eksperiment dilakukan dengan menggunakan data Shipsnet dan diuji pada data satelit. Pada penelitian ini, evaluasi lintas domain juga dilakukan untuk dataset yang diambil menggunakan Google Earth. Hasil eksperiment mengindikasikan bahwa MobileNetV2 memiliki performa klasifikasi dan deteksi terbaik dengan akurasi sebesar 99.07%. Metode MobileNetV2 juga dapat mendeteksi kapal laut pada skenario eksperiment lintas domain.

Kata kunci: *convolutional neural network, deep learning, selective search, pendeteksian kapal laut*

Abstract

The traffic in the water area such as harbor and sea strait is highly important to be monitored because it helps to minimize unwanted ships accident. As a result, we proposed an automatic detection method to localize ships contained in sattelite image. We examine several deep learning methods as the classification backbone, namely MobileNetV2, DenseNet121, VGG16, and ResNet50. Afterwards, we employed the trained model for detecting the ships. To make the detection faster, inspite of using a sliding window, we use selective search to sample the object candidates from the given scene. The experiments were done using Shipsnet dataset and tested on aerial images. We also conducted a cross domain evaluation where the images were taken using Google Earth. The results indicate that MobileNetV2 has the best performance on classification and detection tasks with 99.07% of accuracy. The MobileNetV2 is also able to detect the ships on cross-domain scenarios.

Keywords: *convolutional neural network, deep learning, selective search, ship detection*

1. Introduction

The development of technology for ship has been advanced rapidly in order to improve the navigation control, navigation security, and reduce manpower cost. The breakthrough in artificial intelligence, especially deep learning, brings a fresh breath in developing robust autonomous ship technology through ships recognition, classification, and ships detection. For example, automatic ships detection can be used for mapping the movement of ships in ocean area, so the unwanted event like collision can be prevented.

Ships detection has important role in monitoring the ocean and becomes significant in remote sensing. Various studies have been done to detect ship automatically, however two challenges still affecting the method performance, such as complex background and the small object scales [1]. Deep learning becomes more popular since AlexNet won the ImageNet challenge in 2012 [2] while previously the approaches dominated by the used of scale-invariant feature transform (SIFT), histogram of oriented gradients (HOG) along with support vector machine (SVM) and Adaboost classifiers [3].

The approach in object detection using deep learning can be divided into three approaches [4]. The first approach is a two-stage method. In two-stage method, the phase is began with finding object candidates, for example selective search, in given scene. Then, the score, label, and coordinate offset are computed for each object candidates. R-CNN is one of the example using the two-stage method [5] while later improved in faster and optimised method [6]. The second approach is one-stage method which aiming to decrease the processing time in the first method. The phase for selecting the candidate proposal is removed, so the objects are detected directly from the feature map. YOLOs [7]–[9], SSD [10], and RetinaNet [11] are the example of the best method which adopting the second approach. The third method is an attention-based mechanisms. This mechanism is inspired by the method proposed in [12] for allowing content-based summarization of information given a variable length source sentence using an encoder-decoder in a neural sequence.

Beside of object detection approaches, the detection of ships become harder causes by the remote sensing image covering a large sea area which make the ship is in small size [13], [14]. Therefore, it becomes computationally expensive when applying directly a dense feature extraction. Consequently, the two-stage schema was preferred here. (E.g.Refs.11,12,19,21,31)

This study examines extensively the robust deep learning for detecting ships in aerial imagery through a transfer learning mechanism using a two-stage method. The pretrained deep learning was chosen as a feature extraction and the top of that we built a classification layer. It used on the first stage. On the detection phase, which is the second stage, we utilized selective search algorithm to detect objects contained on the given image. Then, the detected object will be filtered by the model outputted from the first stage and the objects classified as not ship will be deleted. Four deep learning architectures were used on the first stage

This paper is written as follows. The proposed method is introduced in section 2. Then, we provide the experiment setup in section 3. In section 4, the experiment results are presented with extensive discussion. Finally, we conclude our findings in section 5.

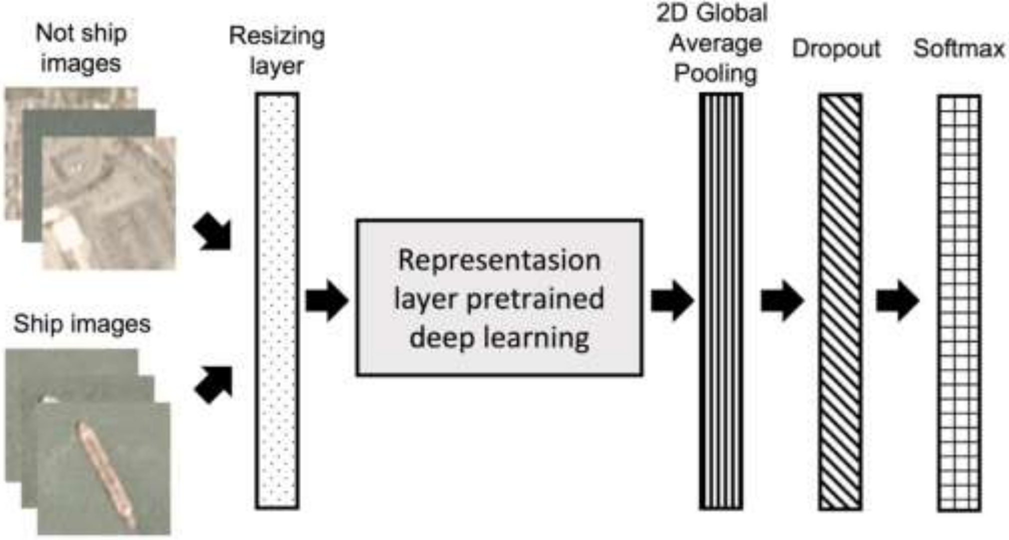
Not ship images
Ship images
Resizing layer
Representasion layer pretrained deep learning
2D Global Average Pooling
Dropout
Softmax

Figure 1. The deep learning architecture used in this study

2. Research Method

2.1 Proposed workflow

This study used a two-stage approach. The first stage purpose is to build a model which can classify an image rather ship or not ship. In this study, the classification model was built using a transfer learning where the pretrained model is the based method. The deep learning architecture built on this study can be seen in Fig. 1. The pretrained model was used as a representation layer. Before the images feed to this layer, we resize the image to accommodate the variation of input size and to fit the required input size for the pretrained model. After that, we applied a global average pooling for the feature map from the last layer of the pretrained model. The pooling operation was important to get one dimensional structure of features to be connected to fully connected layer. Then we insert a dropout layer to prevent the vanishing gradient problem and overfitting. Finally, a dense layer was placed at the last as a classification layer and contained a softmax activation layer. This layer will produce a probability of the input image belong to certain classes.

In the second stage, we firstly detect the object candidates using selective search algorithm. Then, all candidates are filtered by classifying each of them using the classification model. The object candidates, which classified as not ship, will be excluded on merging process. In the merging process, we combine the overlap region of objects into one object.

2.2. Pretrained Deep Learning

1) MobileNetV2: MobileNet is a model which use depth-wise separable convolutions, a combination of several convolutional layers which consist as deptwise convolution and pointwise convolution [15]. Each input channel on MobileNet applied depthwise convolution on one layer, then followed by pointise convolution to merge the output from input layer. Most known type of MobileNet currently are MobileNet and MobileNetv2. MobileNetV2 have 1 additional layer on MobileNet’s residual block. This additional layer used for data channel expansion before the data is sent into depthwise convolutional layer.

2) DenseNet: DenseNet is one of pretrained neural network method which utilize repeated feature usage, that makes DenseNet is an easily trained model and has high efficiency parameter [16]. Combination of feature maps that come out from different layers increased the input variation and the efficiency of DenseNet model. Size of used layers on DenseNet is very narrow, with few additions of feature maps to collect knowledge from networks but maintained other feature map left.

DenseNet starts with a basic convolution and pooling layer, followed by a dense block, then continued into transition layer. After that, add another dense block followed by a transition layer. This process can be repeated several times, then add a dense block followed by a classification layer. DenseNet121 consist of 1 7x7 convolution layer, 58 3x3 convolution layers, 61 1x1 convolutional layers, 4 average pooling, and 1 fully connected layer. In short, DenseNet121 has 120 convolutions and 4 average pooling.

3) ResNet: ResNet stands for Residual Network. ResNet was made in 2016 by Researcher team from Microsoft [17]. ResNet inspired by VGG and have its own convolutional layer with 3x3 filter size. Similar to VGG, ResNet also have some variant depending on the amount of used layer, i.e., ResNet50. ResNet50 used 50 neural networks layers. ResNet50 use 3-layer bottleneck block which replaced 2-layer block in ResNet34. ResNet creation must follow 2 main principles, such as (i) output which we got from the same size feature map must have the same filter size on each layer and (ii) if the size of feature map is only half from the start, then the amount of filter needs to be multiplied by 2 to maintain the complexity of each layer.

4) VGG16: VGG is an abbreviation of Visual Geometry Group which developed by Oxford University. VGG architecture has already tested on ILSVR2014 and come out as first runner up on classification task [18]. There are 2 most used VGG type, such as VGG16 and VGG19. Number that appeared after the VGG is the amounts of layers that used in the model. VGG16 used 16 layers on its creation, which consist of 5 convolutional block that connected into multilayer perceptron classifier. This multilayer perceptron is made by 2 hidden layers and 1 output layer.

2.3. Selective search

We used the selective search proposed in [19] as the initial region of interest. This method was developed to rapidly capture all objects in any scales contained on the scenes using various strategies. Selective search was developed to capture multiple scales of objects, improve diversification on detecting varying types of objects, and decrease the detection time. Multiple scales detection was handles by hierarchical grouping algorithm. This approach surpasses the efficiency of the sliding window method, as it eliminates the need to run multiple window sizes across all pixels in the image. The process involves initially segmenting the image based on color, texture, intensity, and other low-level features. Secondly, a greedy algorithm is applied to iteratively merge segments into larger regions. In the third step, a set of region proposals is generated, considering various scales and shapes to capture potential object boundaries. The fourth step involves measuring objectness scores and ranking the region proposals. The final set of region proposals is obtained by selecting the top-ranked proposals.

3. Experiment Setup

The experiment was conducted in such environment and scenarios. The methods were developed using Tensorflow and ran on GPU RTX2060 6GB, Intel core i5, and RAM 16 GB. For the dataset, we used Shipsnet dataset which can be found on Kaggle. In the resizing layer, we resize the image into 96x96. The proposed method was trained using Adam optimizer and 0.001 of learning rate. The loss function used in this study is cross entropy. Given C number of classes, the prediction $\hat{y}$, and target y, then the loss can be computed using the following equation.

$$\mathcal{L}_{ce} = \sum_{i}^{C} y\log(\hat{y}) \quad (1)$$

Furthermore, the evaluation was done in two phases. In the first phases, the evaluation was done by examining the performance of the methods on classifying ships images and not ships images. The Shipsnet dataset is not balanced where it provides 1,000 images for ships and 3000 images as not ships. Then we augmented the ships images to make the data in balanced condition. Moreover, the k-Fold cross validation was chosen with k is equal to 10. For each fold, we measure the classification accuracy and the loss value.

The accuracy was computed using equation 2 where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative respectively. Equation 3 show the categorical cross entropy loss function used in this study. N is the size of output, y_i correspond to the actual target value, and $\hat{y}_i$ is the predicted value from the model.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Loss = -\sum_{i=1}^{N} y_i \cdot log\hat{y}_i \quad (3)$$

In the second phases, the examination was done qualitatively. The images' contained ships were extracted from scene given from Shipsnet dataset. Moreover, we conduct a cross domain evaluation where the images taken from Google Earth in Bali's local harbour to show the generalization performance of the trained deep learning.

4. Results and discussion

The classification results can be seen in Fig. 2. The accuracy presents the performance of the given trained model to distinguish ship and not ship while the loss value, which is given in Fig. 3, provides the information how far the predicted value to the original value. In the accuracy, the higher its value, the more data are classified as the original class. The lower the loss value means that the predicted value is closer to the true value. From Fig. 2, we found that the best model is MobileNetV2 with average accuracy about 99.07% and standard deviation about 0.38%. MobileNetV2 provides the highest accuracy compared to other. Moreover, the second-best method is DenseNet121 with 98.28% and 0.49 of accuracy and standard deviation, followed by VGG16 and ResNet50 where their accuracies are 94.70%±0.92 and 80.53%±4.25 respectively. This result demonstrates that MobileNetV2 has better feature representation than alternative methods for ship classification.

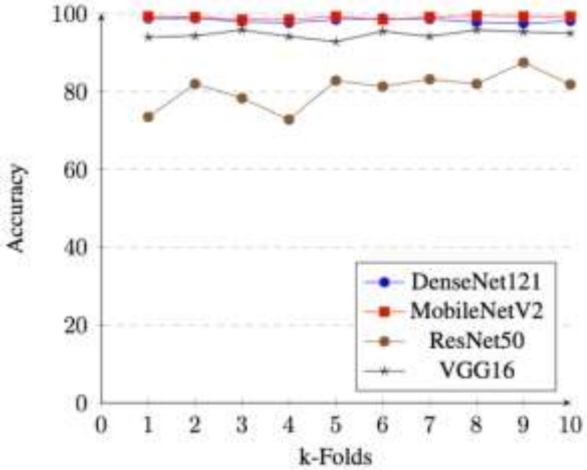

Figure 2. The accuracies for all k-folds at k = 10 for all methods. Note: higher is better

Similar results were also figured out in the loss value for all folds as shown in Fig. 3. The lowest loss value is produced from MobileNetV2, and the highest loss value is computed by VGG16. This loss value results show that MobileNetV2 prediction is evenly matched the actual value. VGG16 shows has higher loss than other methods. DenseNet121 has closer performance to MobileNetV2 since its loss value has no significant differences. Moreover, Table 1 shows the number of parameters for each deep learning models. MobilenetV2 has lowest number of parameters, indicating less resource utilization for inference. On the other hand, VGG16 has the highest number parameters among others.

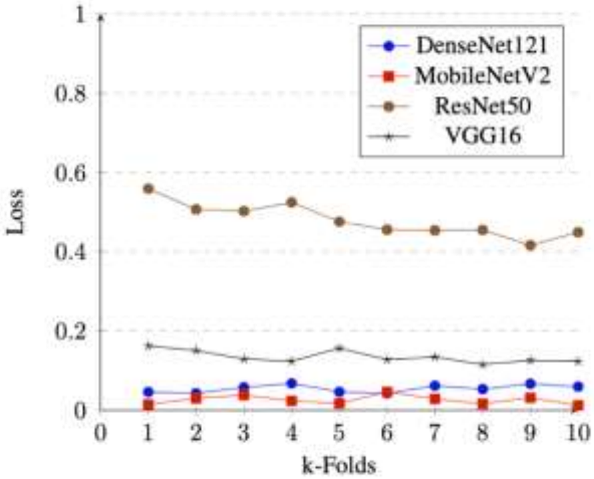

Figure 3. The loss for all k-folds at k = 10. Note: lower is better

Furthermore, Fig. 4 depicts the detection results of all methods. In the top row, where only one ship is appeared in the scene, MobileNetV2 detects the ship accurately while VGG16 is failed to detect the ship. ResNet50 detects the ship as two objects and DenseNet121 has false positive detections. When the scence become more complex (row 2 to 4), MobileNetV2 accurately detect the ships. DenseNet121 produces a lot of false positives, which means it detects the background as the ships. ResNet50 shows good detection on the the complex scene. VGG16 contains falls negatives in the second and fourth row.

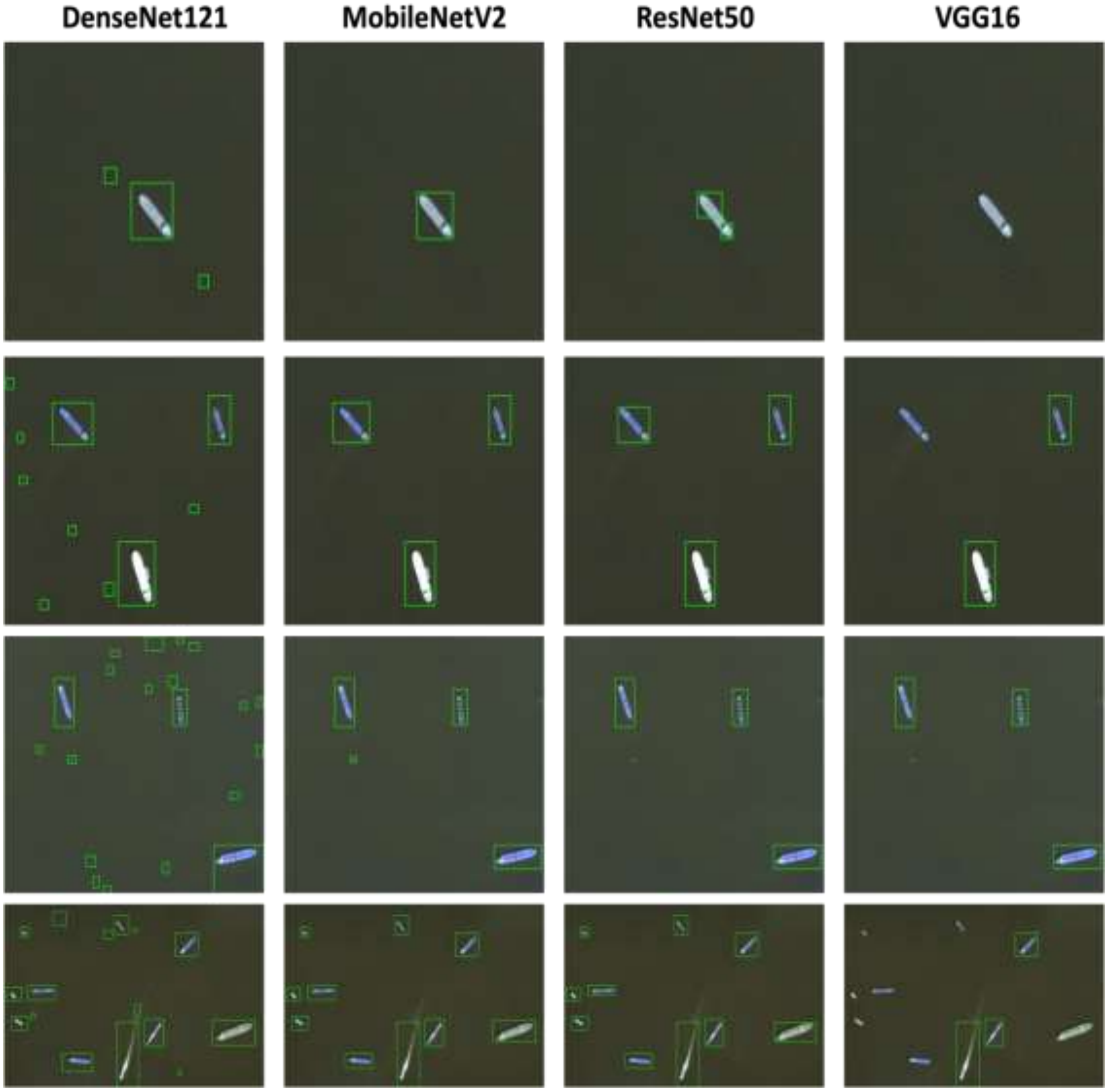

Figure 4. Detection results on satellite imagery image for all methods. The green box indicates the detected ship.

Table 1. Number of parameters of all deep learning models used in this study.

Method	Number of parameters (millions)
DenseNet121	6,9
MobileNetV2	2,2
ResNet50	23,5
VGG16	134,2

Figure 5 shows detection results on cross domain of MobileNetV2 and VGG16. The cross-domain evaluation was conducted to see the generalization performance of the best and the poorest methods based on the results in Fig. 4. As depicted in the top row of Fig. 5, MobileNetV2 is able to detecth the ships eventhough produces come false positive, while VGG16 misses some ships. The first-row images show similar properties to the images in Fig. 4, where the sea has green colour. When we use different scene (second row of Fig. 5), MobileNetV2 is hardly locating the ships and VGG16 detects no ships. This observation means that MobileNetV2 and VGG16 overfit to the training data. Therefore, a special treatment might be needed to improve the performance on cross domain dataset.

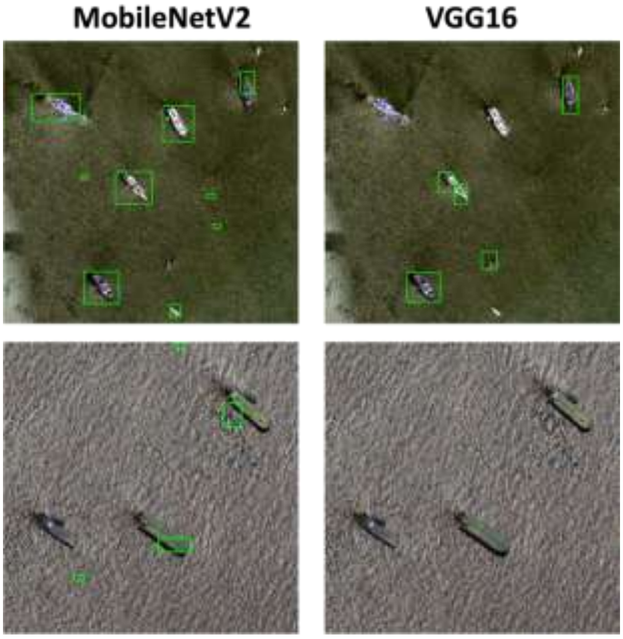

Figure 5. Cross-domain deteection of MobileNetV2 and VGG16. The images are taken using Google Earth

5. Conclusion

In this study, we proposed a workflow for ship detection by combining deep learning and selective search. We examine several deep learning methods, namely DenseNet121, MobileNetV2, ResNet50, and VGG16. The classification accuracy and loss show that MobileNetV2 has better performance than others with the accuracy of 99.07% ± 0.38. Beside of classification results, we also conducted a qualitative evaluation. The qualitative evaluation shows that MobileNetV2 has better accuracy, while DenseNet121 produces high false positive. For generalization performance, we compared MobileNetV2 and VGG16 on cross-domain evaluation where the images are taken using Google Earth on Bali's local harbour. The results indicate that MobileNetV2 is ablet to detect the ships, but still needs improvement when the scene characteristic is different. As future research direction, we are going to develop a method which has good generalization performance, so the method can be performed well in varying environments.

Acknowledgement

This study is funded by Penelitian Unggulan Program Studi 2021 entitled by “Deteksi kapal laut secara otomatis untuk monitoring laut berbasis convolutional neural network.”

References

[1] H. Guo, X. Yang, N. Wang, and X. Gao, “A centernet++ model for ship detection in sar images,” Pattern Recognition, vol. 112, p. 107787, 2021.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, vol. 25, pp. 1097–1105, 2012.

[3] Y.-L.Chang,A.Anagaw,L.Chang,Y.C.Wang,C.-Y.Hsiao,andW.-H. Lee, “Ship detection based on yolov2 for sar imagery,” Remote Sensing, vol. 11, no. 7, p. 786, 2019.

[4] H. Perreault, G.-A. Bilodeau, N. Saunier, and M. He´ritier, “Spotnet: Self-attention multi-task network for object detection,” in 2020 17th Conference on Computer and Robot Vision (CRV). IEEE, 2020, pp. 230–237.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[6] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” Advances in neural information processing systems, vol. 28, pp. 91–99, 2015.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779– 788.

[8] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.

[9] A. Farhadi and J. Redmon, “Yolov3: An incremental improvement,” in Computer Vision and Pattern Recognition, 2018, pp. 1804–02 767.

[10] Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. -Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in European conference on computer vision. Springer, 2016, pp. 21–37.

[11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dolla´r, “Focal loss for dense object detection,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[12] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” arXiv preprint arXiv:1409.0473, 2014.

[13] Y. Tan, H. Liang, Z. Guan, and A. Sun, “Visual saliency based ship extraction using improved bing,” in IGARSS 2019-2019 IEEE Interna- tional Geoscience and Remote Sensing Symposium. IEEE, 2019, pp. 1292–1295.

[14] F. Yang, Q. Xu, F. Gao, and L. Hu,“ Ship detection from optical satellite images based on visual search mechanism,” in 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, 2015, pp. 3679–3682.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proceedings of the IEEE confer- ence on computer vision and pattern recognition, 2017, pp. 4700–4708.

[17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[18] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[19] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition,” International journal of computer vision, vol. 104, no. 2, pp. 154–171, 2013.