Satellite SAR ship detection using Convolutional Neural Network

Sanaz Salati
June, 2019

Introduction

My Internship is based at CGG Company in Crawley, UK. CGG is a geophysical company that provides geological, geophysical, and reservoir services to customers primarily from the oil and gas industry. The company has mainly three business: equipment, acquisition, and geology, geophysics and reservoir. NPA is the satellite mapping group of CGG developing industry-leading solutions that help de-risk exploration activities and offshore operations. NPA provides remote sensing services to the oil and gas, mining, environmental and civil engineering industries. Detecting ships is an important task in oil and gas and environmental pollution monitoring services. It is one of the requirements for de-risk exploration with satellite mapping and offshore operations. My project is supervised by the NPA group and they have provided SAR data in order to detect ships using machine learning and overcome with shortcomings of traditional methods of ship detections.

Satellite SAR ship detection

Ship detection plays a significant role in the maritime, environmental and energy sectors. With the development of remote sensing technology, there have been increasing attention to applying earth observation data for ship detection. Synthetic Aperture Radar (SAR) data hold a great promise for this aim, as it is independent of cloud coverage and weather condition and could be collected in day and night (Yang et al., 2018; Yang et al., 2012; Jiao et al., 2018). Traditional methods of ship detection using SAR are often associated with difficulties in nearshore areas and with false positives such as icebergs, small islands, and speckle noise (Crisp, 2004).

The deep learning method uses neural networks with many layers to model representations of input data (Goodfellow et al., 2016). Deep learning Convolutional Neural Network (CNN) has been widely applied for object detection; however, few attempts have been made on ship detection from SAR imagery (Jiao et al., 2018; Chang et al., 2019; Wang et al., 2018). SAR is one of the most reliable datasets for ship detection due to its imagery capability.

The size of the dataset is important to train a good model in CNN so that there would be sufficient representatives for each class to avoid overfitting and obtain discriminant features. One problem in some research is using a low number of samples which complicated the regularization (Tang et al., 2015; Yang et al., 2012). Few studies used a large number of the dataset collected from Google Earth (Zou and Shi, 2016; Zhang et al., 2016). However, it is difficult to access these datasets as they are not public. For this project, we have collected 2000 SAR satellite imageries of different locations in the world from various satellite platforms such as ALOS, Cosmo-SkyMed, Envisat, ERS, Planetscope, Radarsat-1, Seasat, Sentinel-1, and Sentinel-2. This dataset is with more diversity in ships and having various SAR image resolution. Thus, a large number of SAR dataset are available to learn powerful models and improve ship detection performance.

The deep learning models for object detection that have applications in detecting ships are of two types: the region proposal classification (Li et al., 2017) and the sliding window (Redmon and Farhadi, 2016). In this project, we apply the most advanced You Look Only Once (YOLO) method because unlike other approaches; YOLO sees the entire image during training and testing periods and therefore encodes contextual information about classes as well as their appearance.

The aim of this project is to design and train CNN to process the SAR images and identify ship locations. We design several CNN architectures using YOLO (Redmon and Farhadi, 2017) to detect ships. After comparing architectures an optimal one for detection small features in large satellite data would be chosen to build a pipeline for ingesting SAR imagery and outputting ship locations.

2. Method

2.1 Pre-processing

Each image undergoes pre-processing procedure including calibration to sigma 0, foster filter (3x3), Terrain correction, and conversion from 32 bit float to 16 bit unsigned. Results of previous research on ship detection using constant false alarm ratio (CFAR) method are used to create labels for imageries. After creating labels, image tiles were created from SAR images to provide a large training set.

There is a possibility that some labels be inaccurate for training dataset because the accuracy of CFAR depends on the estimation accuracy of the background's probability density function; thus, it does not show good performance under low-contrast condition. We would refine the model after accessing the ground truth.

2.2 YOLO object detection

In this project, we use You Look Only Once (YOLO) method developed by Redmon et al (2016) and modify it to improve its performance to detect ships in satellite SAR data. Different versions of YOLO developed by authors, would be applied and be compared with each other regarding their performance in detecting ships.

YOLO is an object detection method, which sees the entire image during training and testing and incorporates contextual information about classes. This method could directly detect the bounding boxes and class probability from images. YOLO is an end to end training CNN uses a single neural network to predict bounding box and class probability (Figure 1). It frames object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. Comparing to region-based CNN (R-CNN) (Li et al., 2017), YOLO has shown better performance in object detection. R-CNN has several stages, which makes it difficult to optimize. It first uses region proposal to generate bounding boxes in an image, then it runs classifier on proposed boxed. Post-processing is used to refine bounding boxes, eliminate duplicate detection, and rescore boxes. YOLO is fast and can be optimized end to end directly on detection performance and makes less than half the number of background errors.

Each SAR image is divided into grid cells with each grid cell predicts one object if the centre of this object falls into that grid cell. Every grid cell predicts B bonding box, which has five components (x, y, w, h, confidence). The x and y are indicative of the centre of the box relative to grid cell location, while w and h are width and height of the bounding box. The predicted confidence score represents how confident model is that the box has a ship (as an object) and how accurate it thinks the box is that it predicts.

The predicted confidence score is zero if there is no ship in that cell. Otherwise, the confidence score would be equal to the intersection over union (IOU) between ground truth and predicted box (Redmon and Farhadi, 2016). Class probabilities are also predicted for each grid cell regardless of the number of bounding boxes. Class-specific confidence score is calculated as:

$$P_r\,(\text{class}_t|\text{object}) * P_r\,(\text{object}) * \text{IOU} = P_r\,(\text{class}_t) * \text{IOU}$$

Where $P_r(\text{class}_t|\text{object})$ is class probabilities that are conditioned on the grid cell containing an object, P_r (object) indicates probabilities of box containing an object, IOU is the intersection over union between ground truth and predicted box, and P_r (class_t) is the class probability.

The confidence score of a specific class gives the probability of that class appearing in the box and how well the predicted bounding box fits the object. The bounding box width and height are normalized by the image width and height to fall between 0 and 1. Bounding box coordinates are offsets of a particular grid cell location so they are also bounded between 0 and 1.

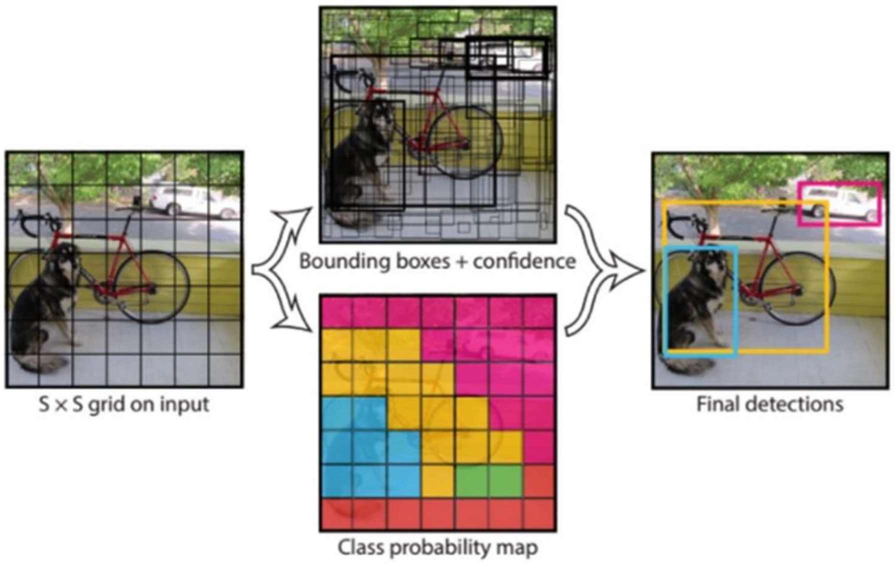

Figure 1. YOLO system models detection as regression problems. Input image is divided into an S ×S grid and for each grid cell model predicts B bounding boxes, confidence for those boxes, and C class probabilities. (Redmon and Farhadi 2016).

Mean square error is applied to measure bounding box coordinates error and dimension errors, and Cross entropy is used as the loss function to measure classification errors.

Optimizing sum-squared error in the output does not perfectly align with maximizing average precision as it weights localization error equally with classification error. Furthermore, many grid cells do not contain any object, and this pushes the confidence scores of these cells to zero, overpowering the gradient of cells without objects (Redmon and Farhadi, 2016). To overcome this problem, Redmon and Farhadi increased the loss of bounding box coordinates prediction and decreased the loss from confidence predictions for boxes that didn't have objects. To address equally weights errors in large and small boxes, they predicted the square root of bounding box width and height instead of width and height directly. A predictor is assigned for predicting an object based on which prediction has the highest current IOU with ground truth, leading to specialization between bounding boxes.

YOLOv2 (Redmon and Farhadi, 2017) and YOLOv3 (Redmon and Farhadi, 2018) are two other versions of YOLO. Anchor boxes are used in the last two versions of YOLO and K-means clustering is applied to automatically find priors. Five clusters are used for YOLOv2 and nine clusters are used for YOLOv3. YOLO v2 (Chang et al., 2019) uses an anchor box method to detect multiple ships cantered in a grid of SAR. Chang et al (2019) used a CNN of YOLO with 22 convolutional layers, five pooling layers, two route layers, and one reorg layer. Route layers merge layers and the final detection layer reorganize extracted features from convolutional layers to predict the probability and bounding box of ships.

2.3 Network architecture

The neural network is inspired by GoogLeNet model (Szegedy et al., 2014) with twenty four convolutional layers and two fully connected layers with randomly initialized weights (Figure 2). Twenty of convolutional layers are followed by average-pooling layers and the last layer predicts both class probability and bounding box coordinates. A linear activation function called leaky rectified linear activation is used for all layers as:

$$F(x) = \begin{cases} x & , if\ x > 0 \\ 0.1x, & otherwise \end{cases}$$

This activation function allows a small, positive gradient when the unit is not active. Drop-out and augmentation are also applied to avoid overfitting.

YOLOv2 has 19 convolutional layers and five maximum-pooling layers, while YOLOv3 has 53 convolutional layers.

Depend on obtained results, the number of the network layers could change to achieve high accuracy performance.

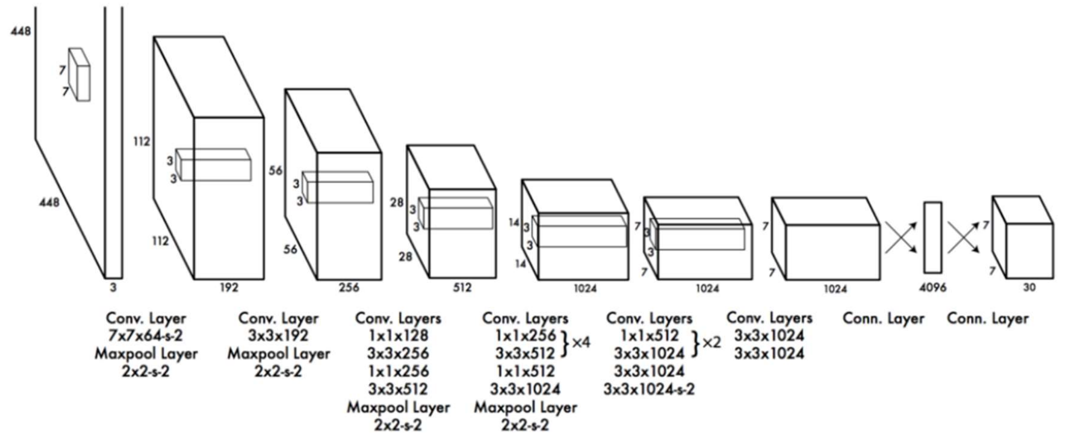

Figure 2. Network architecture of YOLO (Redmon and Farhadi, 2016)

3. Final deliverable

Final deliverable is stand-alone including python codes of created models and their tests. Outputs of the models would be attached to final report. I deliver my project report and python cods by 30/08/2019.

4 Research plan

First step: Starting point	Completed: 29/03/2019
1. Confirmation of my internship 2. Clarification of my project topic 3. Started to search for literatures relevant to the topic	
Second step: Reading and researching	Completed: Continues
1. Finding relevant papers and resources 2. Evaluating online resources such as GitHub 3. Choosing the most advanced and relevant object detection method (YOLO) and evaluating various versions of YOLO	
Third step: started my internship at CGG	Internship starting date:10/06/2019
1. Met my supervisor 2. Discussing the project and objectives 3. Getting few imageries to be familiarize with and started creating labels	
Fourth step: Writing and submission of project proposal/plan	Completed:28/06/2019
1. Literature review 2. Wrote project objectives 3. Wrote about available data 4. Wrote about the method (YOLO) 5. Getting familiar with working in Linux environment 6. Worked on codes for creating labels 7. Wrote python codes for creating labels 8. Labels and tiles are created	
Fifth step: achieving project objectives	To be completed: 10/08/2019
1. Literature review 2. Writing python scripts for several architectures 3. Testing different architectures of YOLO 4. Improving architectures 5. Comparing results of different architectures 6. Selecting the optimal one 7. Modifying YOLO to our aim for ship detection using SAR 8. Build a pipeline for ingesting SAR imagery and outputting ship locations	
Sixth step: Revision and submission python codes and report	To be completed:30/08/2019
1. Writing project report 2. Reviewing the report by supervisors 3. Editing report based on supervisor comments 4. Submission of python codes and report to GitHub	.
Completing the internship	31/08/2019
Presenting my summer project	9 or 10 of September, 2019

5 References

Chang, Y., Anagaw, A., Chang, L., Wang, Y., Hsiao, C., Lee, W. (2019). Ship detection based on YOLOv2 for SAR imagery. Remote Sensing, 11, 768, doi: 10.3390/rs11070786.

Crisp, D. (2004). The State-of-the-Art in Ship Detection in Synthetic Aperture Radar Imagery. Australian Government, Department of Defense: Canberra, Australia, p. 115.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep learning, The MIT Press, 800 pp, ISBN: 0262035618

Jiao, J., Yang, X., Hong, W. (2018). A Densely Connected End-to-End Neural Network for Multiscale and Multiscene SAR Ship Detection. IEEE Access, doi: 10.1109/ACCESS.2018.2825376.

Li, J., Qu, C., Shao, J. (2017). Ship detection in SAR images based on an improved faster R-CNN. In Proceedings of the 2017 SAR in Big Data Era: Models, Methods and Applications (BIGSARDATA), Beijing, China, 13–14 November, 1–6.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.

Redmon, J., Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.

Redmon, J., Farhadi, A. (2018). YOLOv3: An incremental Improvement. Arxiv.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June , 1–9, doi:10.1109/CVPR.2015.7298594.

Tang, J., Deng, C., Huang, G.B., Zhao, B. (2015). Compressed-Domain Ship Detection on Spaceborne Optical Image Using Deep Neural Network and Extreme Learning Machine. IEEE Trans. Geosci. Remote Sens, 53, 1174–1185, doi:10.1109/TGRS.2014.2335751.

Wang, Y., Wang, C., Zhang, H., Dong, Y., Wei, S. (2019). A SAR dataset ship detection for deep learning under complex backgrounds. Remote Sensing, 11 (7), 765.

Yang,W., Dai, D., Triggs, B., Xia, G.S. (2012). SAR-Based Terrain Classification Using Weakly Supervised Hierarchical Markov Aspect Models. IEEE Trans. Image Process, 21, 4232–4243, doi:10.1109/TIP.2012.2199127.

Yang, X., Sun, H., Fu, K., Yang, J., Sun, X., Yan, M., Guo, Z. (2018). Automatic ship detection in remote sensing images from Google Earth of complex scenes based on multiscale rotation dense feature pyramid networks. Remote Sens, 10 (1), 132, 2018.

Zhang, R.; Yao, J.; Zhang, K.; Feng, C.; Zhang. (2016). J. S-CNN Ship Detection from High-Resolution Remote Sensing Images. ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci, 423–430, doi: 10.5194/isprs-archives-XLI-B7-423-2016.

Zhu, X.X., Tuia, D., Mou, L., Xia, G.S., Zhang, L., Xu, F., Fraundorfer, F. (2017). Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. IEEE Geosci. Remote Sens. Mag, 5, 8–36, doi:10.1109/MGRS.2017.2762307.

Zou, Z., Shi, Z. (2016). Ship Detection in Spaceborne Optical Image with SVD Networks. IEEE Trans. Geosci. Remote Sens, 54, 5832–5845, doi:10.1109/TGRS.2016.2572736.