Machine Learning:
Detect and Segment Objects

Xushan Hu
huxushan@bu.edu
Boston, MA

Introduction:

Detect and segment objects in images are two foundational and long-standing challenges in Computer vision, in other words, real-world image-based surface defect classification task. Detecting objects focuses on finding certain classes of objects, while the latter problem is concerned with getting a complete, pixelwise labeling of the image. The objects include different segments in the final labeling. Detecting and segmenting an object from multiple classes have gained interest in recent years.

Here are lots of researches to deal with these two problems. A semi-supervised framework which exploit both appearance cues learned from rudimentary detections of object-like regions, and the intrinsic geometric structures within multi-view data. This framework generates a diverse set of object proposals in all views which underpins a robust object segmentation method to handle objects with complex shape and topologies, as well as scenarios where the object and background exhibit similar color distributions.[1] Another framework combines a flexible probabilistic model, for representing the shape and appearance of each segment, with the popular “bag of visual words” model for recognition.[2]

Popular deep learning models for detecting and segmenting objects are Support Vector Machine, Mask-RCNN, R-FCN(Region-based Fully Convolutional Networks), SSD(Single Shot Multibox Detector) and so on.

Coco dataset, Kitti dataset, Open images dataset, AVA v2.1 dataset and iNaturalist Species Detection dataset. These datasets are pre-trained models which are provided by Tensorflow. This report will introduced you the models based on Tensorflow platform.

Model Analysis:

There are a number of different models implemented in Tensorflow:
1. The official models are a collection of example models that use TensorFlow's high-level APIs. They are intended to be well-maintained, tested, and kept up to date with the latest stable

TensorFlow API. They should also be reasonably optimized for fast performance while still being easy to read. We especially recommend newer TensorFlow users to start here.[3]
2、The research models are a large collection of models implemented in TensorFlow by researchers. They are not officially supported or available in release branches; it is up to the individual researchers to maintain the models and/or provide support on issues and pull requests.[3]

Support Vector Machine:
SVM model is a classification method which is developed from generalized portrait algorithm. For detecting and segmenting objects, SVM changes classifier to detector. SVM map the data to a predetermined very high dimensional space via a kernel function and find the hyper plane that maximizes the margin between the two classes, So SVM is extremely efficient and robust in the content-based image classification. [4]
The process of SVM is as follows:

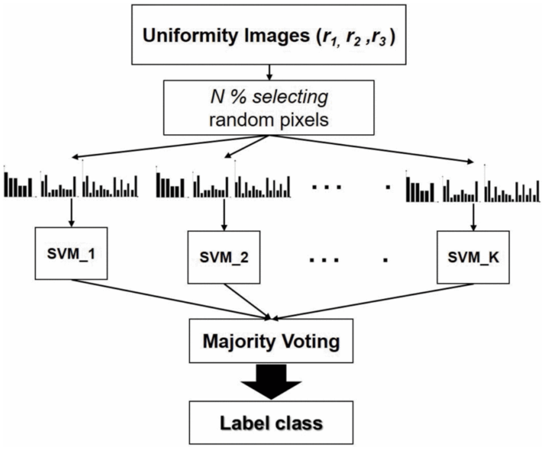
Uniformity Images (r_1, r_2, r_3)
N % selecting random pixels
SVM_1
SVM_2
SVM_K
Majority Voting
Label class

Figure1: An architecture of the SVM

The result of SVM classifier is usually represented as some featured values and rates.
Pros:

1. SVM theory provides a way to avoid the complexity of high-dimensional space, directly use the inner product function of this space (which is also the kernel function), and then use the solution method under the condition of online separability to directly solve the decision-making problem of the corresponding high-dimensional space. When the kernel function is known, it can simplify the difficulty of solving high dimensional space problems.
2. SVM is based on the theory of small sample statistics, which is in line with the purpose of machine learning. Moreover, SVM has better generalization ability than neural network.

Cons:

1. For the mapping F of each high-dimensional space in this space, there is no suitable method to determine F , i.e. kernel function, so for general problems, SVM only turns the complexity of high-dimensional space into the difficulty of finding kernel function.
2. Even after the kernel function is determined, the quadratic programming of the solution function is required to solve the problem classification, which requires a lot of storage space.

Mask R-CNN (bounding boxes, scores and masks)**:**
Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework[5]. Mask R-CNN is a general framework for object instance segmentation, which can detect objects in an image accurately while generating a segmentation mask for each instance.[6]

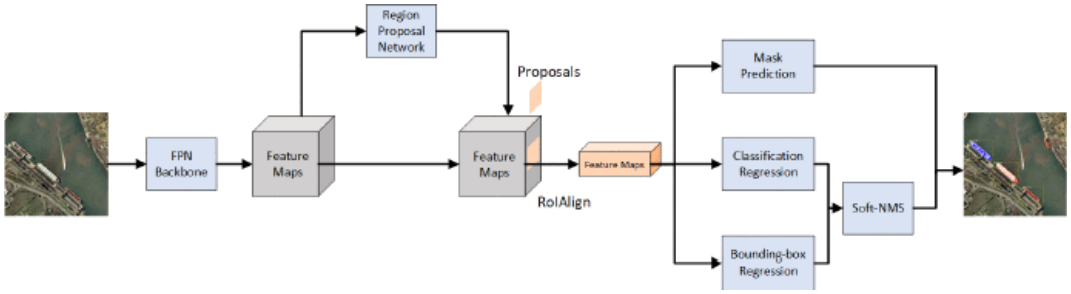

Figure2: Framework of Mask R-CNN

Here is an example of inshore ship detection to represent the result:

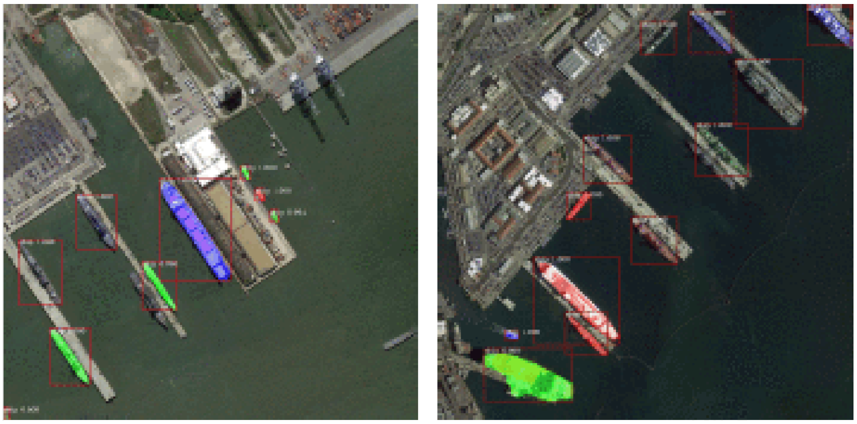

Figure3: Result of Inshore Ship detection

Pros:
1. Region of interest is generated in the first stage by using region proposal network, and then region proposal is sent to pipeline for object classification and boundary box regression. It has high performance.

2. Mask R-CNN is simple to train and easy to generalize to other tasks.
Cons: The process is really slow.

R-FCN:

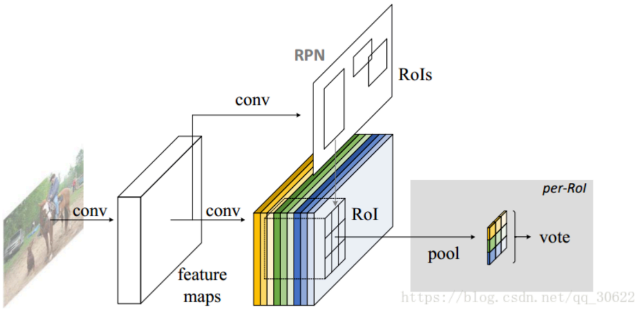

Figure4: Whole framework of RFCN

This model is used for objects detection. The deep network can be divided into two sublayers by using ROI pooling layer: a shared subnetwork and a subnetwork without share computation. All the learnable layers are convolutional and shareable in the whole picture, and can still encode spatial location information for object detection.
Pros:
1. It has higher accuracy and faster operation speed than other neutral network.
Cons:
1. It is hard to understand on semantic segmentation and objects detection.

A lot of models can be trained by datasets which are provided by Tensorflow zoo:
Pros and Cons:
Pros:

1. Tensorflow can use Python and Numpy
2. It has abstraction of calculation graph.
3. It has shorter complier time.
4. Use TensorBoard to do the visualization.
5. Support data parallel and model parallel at the same time.

Cons:

1. Tensorflow is slower than other frameworks.
2. It is hard to understand.
3. There are less pre-trained models.
4. The calculation chart is purely Python based, so it is slow.

Recommendation:

1. Do some changes based on the existed good baseline models to increase operation speed and shorten operation time.

2. Consider the models' accuracy and performance.
3. For users of Tensorflow zoo, First, making sure setting the environment correctly and check the situation of GPU. Second, optimizing the training code to shorten the processing time. Third, By using multithreaded queue to process data on CPU, GPU can use enough data at any time and focus on training, which can greatly improve the training speed of the model.

Conclusion:

Tensorflow zoo is a good starter for user to apply different models of machine learning. In my report, I just analysis three models in the zoo: SVM model, mask R-CNN model and R-FCN model. SVM model is a very popular classification method in machine learning, which suitable to be used in the situation of kernel function known. However, it is hard to find the kernel function in the situation of high dimension space, and it need more storage space. Mask R-CNN model is an advanced version of Faster R-CNN model, it has high performance for generating object instance segmentation, however, the process is too slow. The last one is R- FCN, which is a detection model include a ROI pooling layer. It also reflect the spatial location information of the object. R- FCN seems to have the highest accuracy and fast operation speed among these models. In addition, there are a lot of other models in Tensorflow zoo and many datasets which are provided to users.

Learning from teammates:

Yunze also analysis detecting and segmenting models: CNN model and R-CNN model. CNN model is more suitable to detect only one imagine patch. R-CNN is an advanced version of CNN which uses region proposals to classification and localization. It is better than CNN model but the problem is the speed.
Danny introduced two approaches to machine language translation from Google: neural machine translation and statistical neural translation. The pros of neural machine translation is that it is truly end to end, however, it is slow to train models. In his report, he introduced us a good method Google NMT as an example of the second approach. It can be capable of translating between any language pair, however, it is not ready for the portable devices.
Peixi introduced us the unsupervised learning. Unsupervised learning deals with problems in which data doesn't have labels. The best use for unsupervised is around exploratory analytics.
Chenhui did some researches in detecting and segmenting images models. He gave us a project as an example and the DeeplabV3 is the best architecture in terms of segment detection in the project. It decreases the speed and increases the mlOU significantly.

Reference:

[1] Hulling Wang ; Tinghuai Wang "Boosting objectness: Semi-supervised learning for object detection and segmentation in multi-view images" , IEEE Xplore, 19 May 2016 https://ieeexplore-ieee-org.ezproxy.bu.edu/document/7471986

[2] Marco Andreetto ; Lihi Zelnik-Manor ; Pietro Perona “Unsupervised Learning of Categorical Segments in Image Collections”, IEEE Xplore, 27 December 2011 https://ieeexplore-ieee-org.ezproxy.bu.edu/document/6112771
[3]https://blog.csdn.net/qq_30460949/article/details/88924412
[4] Wang Xin-Lu ; Li Xiao-juan ; Liu Xiao-bo “Nude Image Detection Based on SVM”, IEEE Xplore, 04 September 2009
[5]Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick “Mask R-CNN”, 20 Mar 2017, Available: https://arxiv.org/abs/1703.06870
[6]Shanlan Nie, Zhiguo Jiang, Haopeng Zhang, Bowen Cai, Yuan Yao, “Inshore Ship Detection Based on Mask R-CNN”, IEEE Xplore, 05 November 2018