COUNTING VEHICLES BY DEEP NEURAL NETWORK IN HIGH RESOLUTION SATELLITE IMAGES

Yohei Koga[1], Hiroyuki Miyazaki[1], Ryosuke Shibasaki[1]

1 Center for Spatial Information Science(CSIS),
University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa-shi, Chiba, Japan
Email: y.koga@csis.u-tokyo.ac.jp, heromiya@csis.u-tokyo.ac.jp,
shiba@csis.u-tokyo.ac.jp

KEYWORDS: vehicle detection, sliding window, Objectness, BING, CNN

ABSTRACT: Recently, much more high-resolution satellite images became available. We can detect vehicles in such high-resolution images, and estimation of earnings by counting vehicles in commercial facilities is becoming popular. For this purpose, we need to detect and count vehicles in satellite images accurately. In applications for detecting vehicles, deep neural network has achieved state-of-the-art performance like in general image classification and object detection. To evaluate the accuracy, we tested two methods: Simplified HDNN (SHDNN), which generates sliding windows and classifies them by CNN, and BING-based CNN (BING-CNN), which extract region proposals by BING and classifies them by CNN. In our experiment, while the SHDNN has achieved better performance than the BING-CNN, the BING-CNN was much faster than the SDHNN. And we found some issues to work on for improving the accuracy of them.

1. INTRODUCTION

Number of vehicles is an important indicator to estimate industrial and commercial outcomes, such production and amount of sales, for sustainable strategy and planning of socioeconomy. High-resolution satellite images are useful resources to count vehicles as the techniques of vehicle detection using the images has been demonstrated by several studies. Conventional algorithms like SIFT and HOG need manual selection of features while automated feature selection algorithms by deep learning achieve state-of-the-art performance in many applications.

In applications for detecting vehicles, Chen et al. demonstrated Hybrid Deep Convolutional Neural Networks (HDNN), a method using sliding windows in a gradient image and classification by HDNN, which is the enhanced architecture of Convolutional Neural Networks (CNN) (Chen et al.,2014). Qu et al. applied Binarized Normed Gradients (BING; Cheng et al., 2014) to extraction of *region proposals*, which are clusters of pixels likely to be individual objects, for vehicle detection in high-resolution satellite images (Qu et al., 2016). Sliding windows usually achieves rather better performance than region proposals while sliding windows is more time-consuming than region proposals. In this paper, we present a performance comparison of the Simplified HDNN (SHDNN) and the BING-based CNN (BING-CNN) through experiments in different conditions by landscape and spatial resolution.

2. METHODS AND ALGORITHMS

2.1 Methods

We tested two methods. The one is the SHDNN, the other is the BING-CNN. The SHDNN is based on the HDNN(Chen et al.,2014) and the BING-CNN is based on the combination of BING(Cheng et al., 2014) and CNN (Qu et al., 2016). In the original paper of Chen et al., they used enhanced architecture of CNN, named HDNN. In this experiments, we applied the same simple architecture of CNN to these two methods for fair comparisons. Implementation detail is explained in the chapter three.

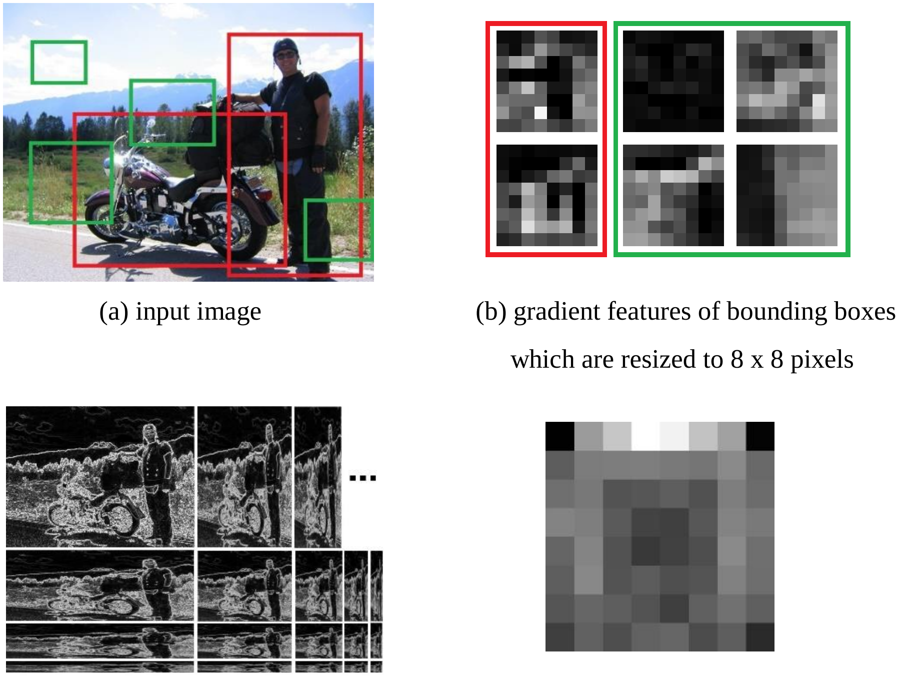
SHDNN: First we calculated gradient images from an input image, and generated sliding windows to cover each whole image. Next we moved each window to its intensity centroid so that the windows covered vehicles better. Then we discarded the repetitive windows, and classified remained windows by the CNN.

BING-CNN: First we extracted region proposals by BING. Then we classified the region proposals by the CNN.

2.2 Algorithms

Convolutional Neural Network(CNN): CNN is architecture of deep learning and it achieves state-of-the-art performance in many kinds of applications like image classification and object detection. CNN generally has convolutional (conv) layers, pooling(pool) layers, and fully connected(fc) layers. A conv layer has filters, and we calculate inner product of a filter and a receptive field in an image. Sliding the receptive field, a map is computed. At a pool layer, the map size is reduced. After going through some conv layers and pool layers, finally a fc layer output a result. Training CNN, filters of conv layers learn features of images. Pool layers enhance the robustness of CNN against shift variance and noise. Conv layers and pool layers work as a feature extractor, and fc layers work as a classifier.

BING(Binarized Normed Gradients for Objectness): BING extract region proposals from an image by *objectness*. Objectness means likelihood of being objects. Fig. 1 shows an example of BING. BING calculates objectness based on the assumption that all objects that are resized to same size have correlated features in a gradient image that an object is enclosed by edge. BING learns features of objects as a fixed-sized filter (eight by eight = 64 dimensions in the original paper), generates a feature map resizing original image to different quantized sizes and calculating the normed gradients of each resized images, and extracts region proposals scanning the map with the filter. BING also uses binary approximation for calculating features, and thus it can calculate features by bit operation. Therefore, BING can extract region proposals very fast.



(a) input image

(b) gradient features of bounding boxes which are resized to 8 x 8 pixels

(c) gradient map

(d) learned filter(8 x 8 pixels)

Figure 1. An example of BING

3. IMPLEMENTATION DETAIL

3.1 CNN architecture

As Fig. 2 shows, our CNN has seven layers: conv1, pool1, conv2, pool2, conv3, pool3, fc. The input image shape is 48 by 48 pixels with three channels. Conv1 has 20 filters whose size is seven. Conv2 and conv3 have 8 filters whose size is four. All pool layers' size and stride are two. Fc layer has two outputs. The CNN was trained with ground truth vehicle images as positive samples and random sampled background images as negative samples from train images. A random sampled image was judged as a negative sample when IoU(intersect over union) of a ground truth and it was under 0.4.

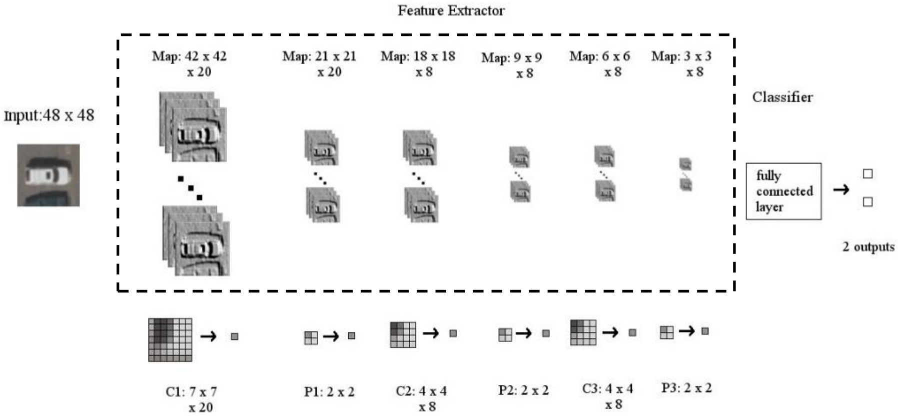


Figure 2. Our CNN architecture

3.2 SHDNN

First we computed three gradient images from an original input image and two thresholding images. In the thresholding images, the one’s pixel values over 60 were set to 60, and the other’s pixel values under 100 were set to 100. Next we generated sliding windows for each gradient image to cover each whole image. Sliding step was half of window size. Then we moved each window to its intensity centroid, enlarged the windows 1.414 times and moved each window to its intensity centroid again. Finally, we collected all windows, filtered the repetitive windows by 0.15 times of window size, and classified the remained windows by the CNN. We used these parameters from the original paper.

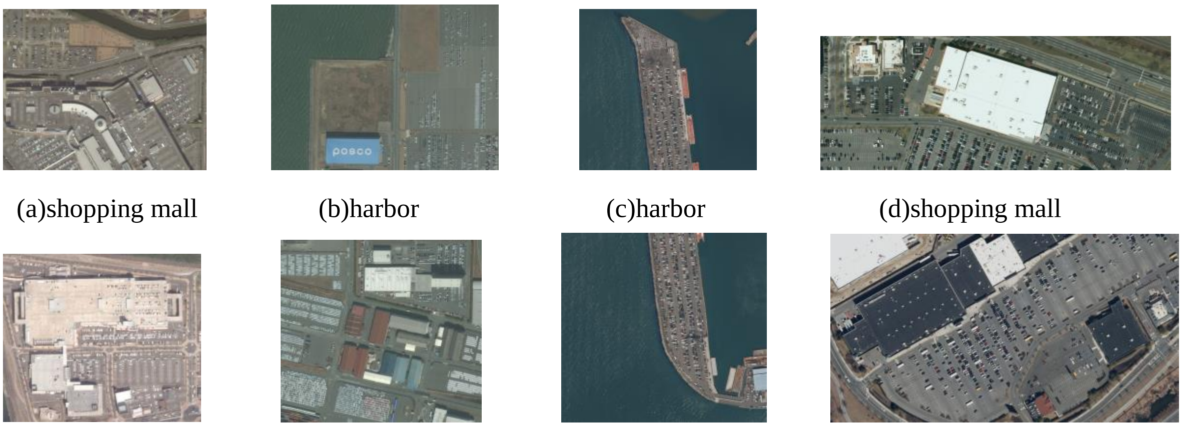
3.3 BING-CNN

First we trained BING using ground truth vehicle images from train images. Then we extracted region proposals by BING from an input image, and classified the region proposals by the CNN.

4. EXPERIMENT

4.1 Train and Test images

Fig. 3 shows our train and test images. We prepared two images of shopping malls and two images of a harbor in Japan from bing map, and two images of shopping malls and two images of a harbor in New York from USGS. Bing map images resolution is one meter and USGS images resolution is zero point five meters. And we split these in half and used the one as training images and the other as test images. We made ground truth data by visual interpretation. The ground truth window size in one-meter resolution images is 25 pixels and the 0.5-meter resolution images is 50 pixels.



(a)shopping mall (b)harbor (c)harbor (d)shopping mall

(e)shopping mall (f)harbor (g)harbor (h)shopping mall

Figure 3. (a)-(d) are train images and (e)-(h) are test images. (a), (b), (e) and (f) are one-meter resolution images in Japan and (c), (d), (g) and (h) are 0.5-meter resolution images in New York.

4.2 Training the CNN

We used 3324 of ground truth vehicle images as positive samples and 116340 of random sampled background images as negative samples. We used SGD as optimizer and a batch size of 100. We trained the CNN for 3000 epochs and it took about 10 hours using GPU. Fig. 4 shows the convergence of accuracy.

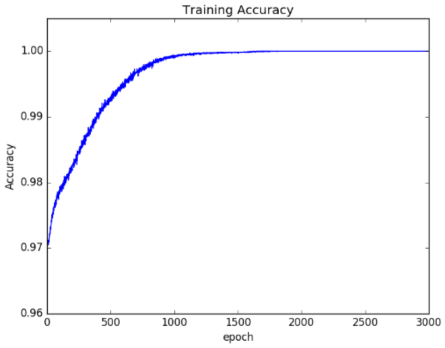


Figure 4. Convergence of accuracy

4.3 Training BING

We used 3324 of ground truth vehicle images as positive samples. It took about 22 seconds for training.

4.3 Definition of quantitative measures

In region proposals by BING, detection rate(DR) are defined as follows:

$$\mathrm{DR} = \frac{\text{number of covered vehicles}}{\text{number of vehicles}} \times 100\% \quad (1)$$

A vehicle is covered if IoU of its bounding box and a region proposal is over 0.3.

In test results, precision rate (PR) and recall rate (RR) are defined as follows:

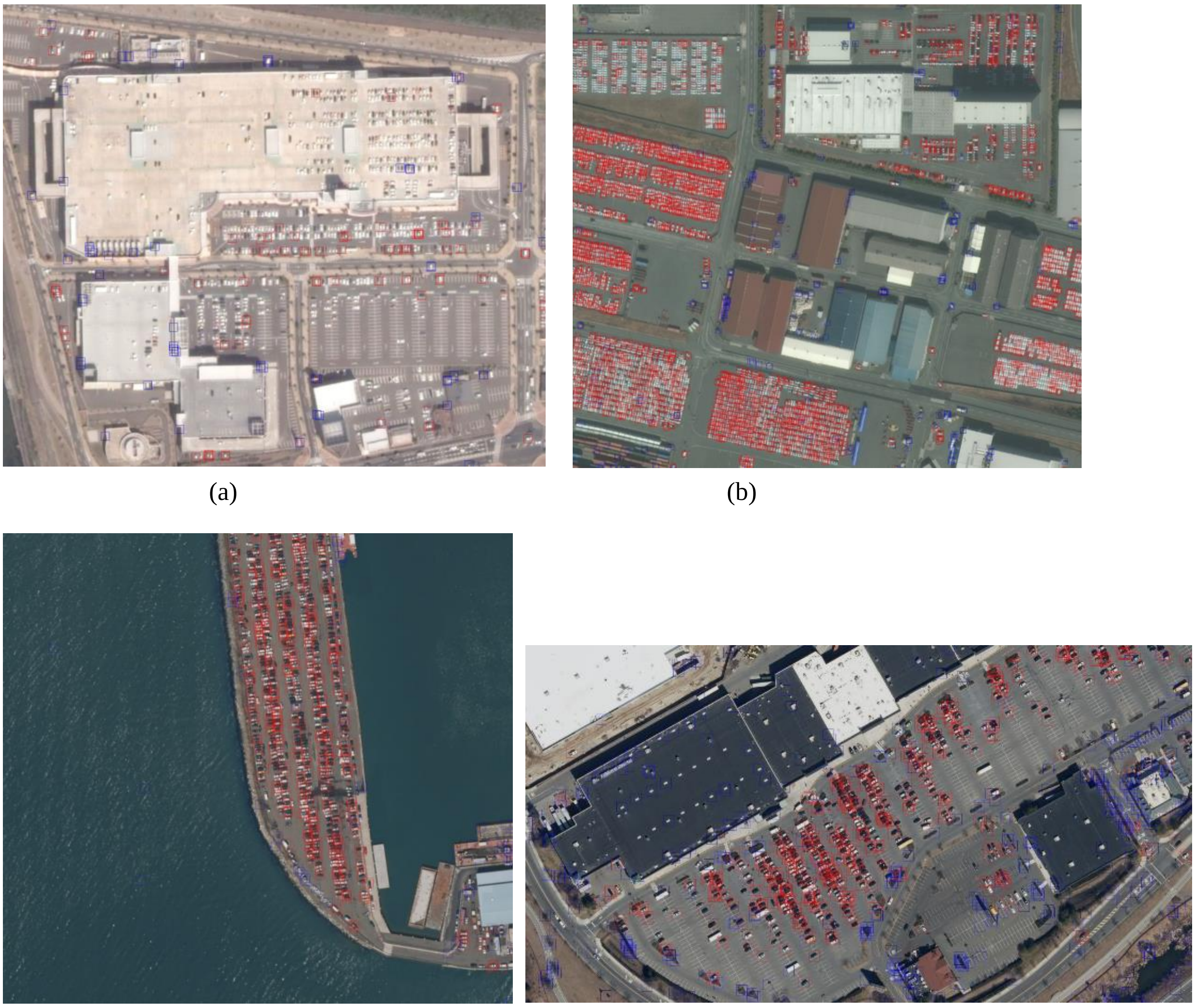
$$\mathrm{PR} = \frac{\text{number of detected vehicles}}{\text{number of detected objects}} \times 100\% \quad (2)$$

$$\mathrm{RR} = \frac{\text{number of detected vehicles}}{\text{number of vehicles}} \times 100\% \quad (3)$$

In SHDNN, a window is exact only if the distance between the window center and a vehicle center is less than 0.45 times of the window size. In BING-CNN, a window is exact only if IoU of the window and a ground truth bounding box is larger than 0.3. A vehicle is exactly located if it has at least one exact window.

4.4 Test by the SHDNN

Fig. 5 and Table 1 show the results. Mean execution time was about four minutes. In Fig. 5 (a), only about ten percent of vehicles were detected correctly. Especially on the roof few vehicles were detected. It might have been caused by difference of color between train images and the test image. In Fig. 5 (b), 70 percent of vehicles are detected. In Fig. 5 (c), about 80 percent of vehicles are detected. In Fig. 5 (d), there are many false alarms while about 80 percent of vehicles are detected. Many areas of which edge features are similar to ones of vehicles seem to have been misclassified. As a whole, RRs in 0.5-meter resolution images tend to be higher than ones in one-meter resolution images.



(a) (b)

(c) (d)

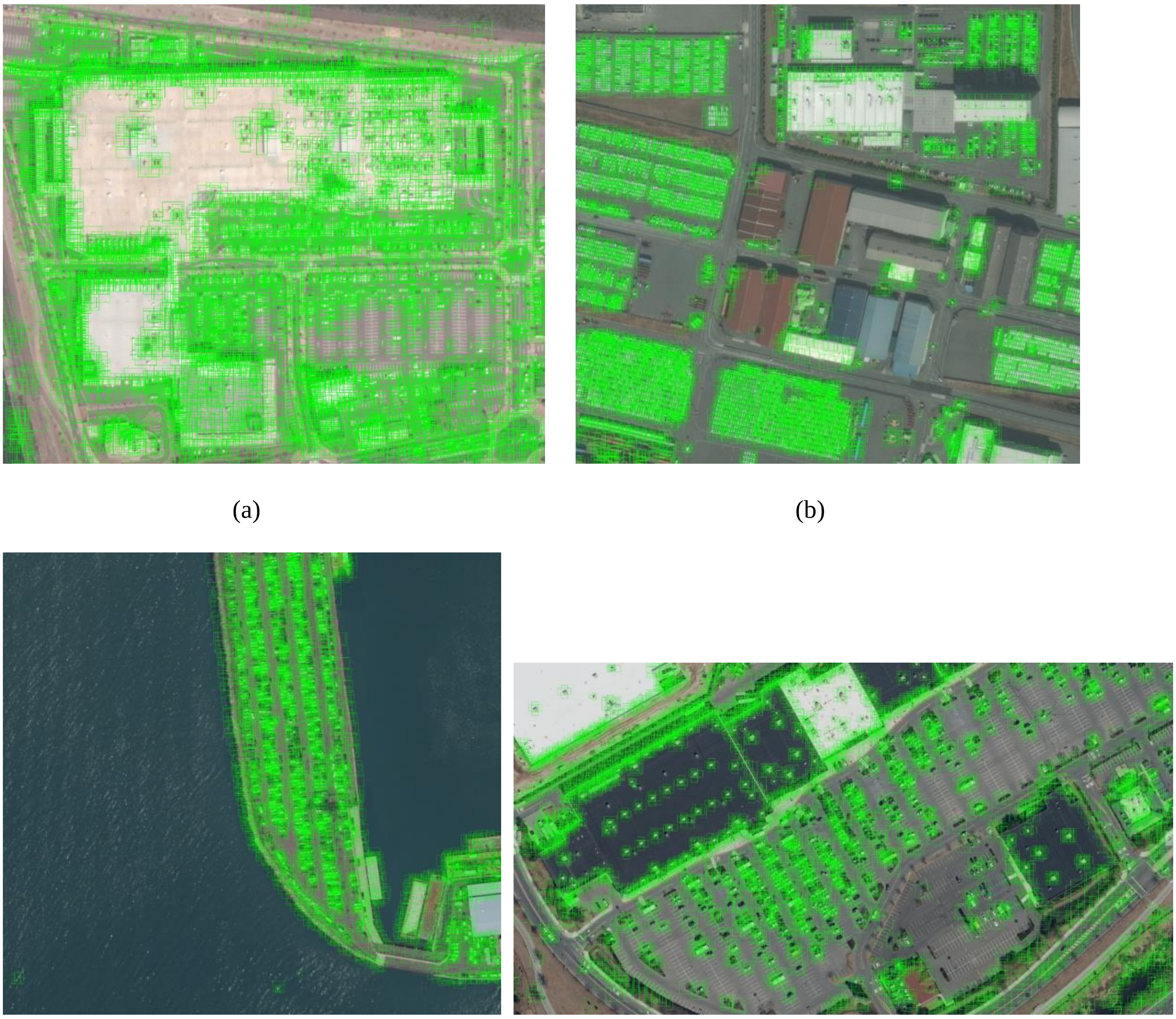
Figure 5. Results of the SHDNN. Red rectangles indicate true positives and blue ones indicate false positives.

Table 1. Results of the SHDNN

Test image	**exec time (sec)**	**PR(%)**	**RR(%)**	**ground truth vehicles**	**detected vehicles**	**detected objects**
(a)	111.214	75.93	12.31	820	101	133
(b)	331.691	92.58	69.42	4226	2934	3169
(c)	283.637	94.52	79.28	806	639	676
(d)	200.022	41.43	81.56	510	416	1004

4.5 Test by the BING-CNN

Fig.6 shows the region proposals by BING, and Fig. 7 and Table 2 show the results. We extracted 10,000 region proposals in each image by BING. It took less one second for each image. And we classified region proposals by the CNN. Mean execution time was about six seconds. In Fig.6 (a) and (b), DRs were lower than in Fig.6 (c) and (d). In Fig.7 (a), the result is similar to the one of the SHDNN while there are more false alarms. And in Fig.7 (b), many windows were judged as false positives while they covered vehicles because of low IoU. These seems to have been caused by the mismatch between window sizes of region proposals which is fixed by parameter of BING and the image resolution. In Fig.7 (c), while about 80 percent of vehicles were detected there are more false alarms than the result of the SHDNN. In Fig.7 (d), while 70 percent of vehicles were detected there are many false alarms like the result of the SHDNN.



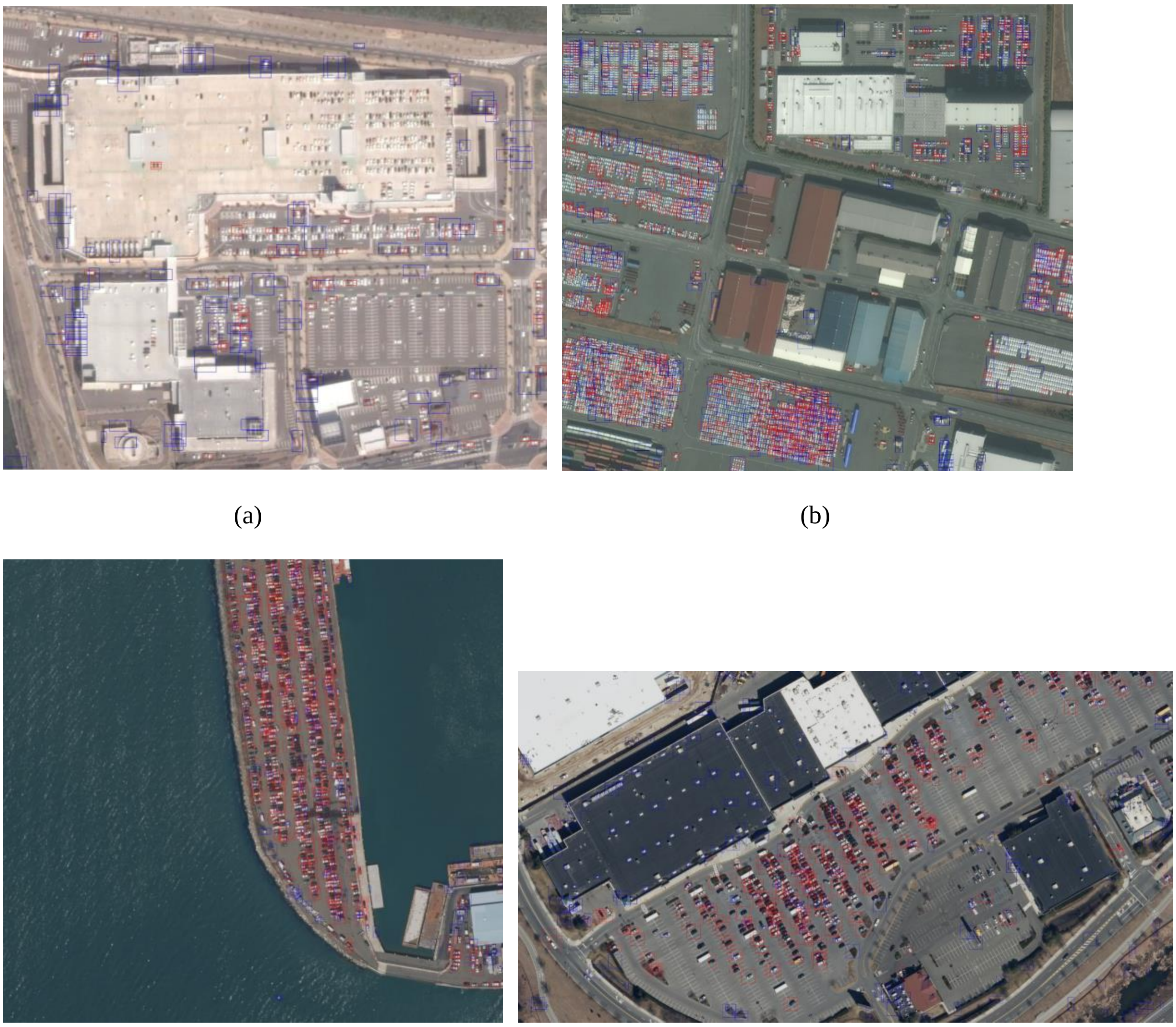
(a)

(b)

(c)

(d)

Figure 6. Green rectangles indicate region proposals by BING.



(a) (b)

(c) (d)

Figure 7. Results of the BING-CNN

Table 2. Results of the BING-CNN

Test image	prediction time(sec)	DR(%)	PR(%)	RR(%)	ground truth vehicles	detected vehicles	detected objects
(a)	3.628	79.26	48.02	10.36	820	85	177
(b)	12.369	70.18	99.22	39.44	4226	1667	1680
(c)	4.076	100	62.11	83.99	806	677	1090
(d)	3.334	99.21	49.24	70.39	510	359	729

5. CONCLUSION

In this experiment, while the SHDNN achieved better performance than the BING-CNN, the BING-CNN was much faster than the SHDNN. And there are some problems: The results of the image whose color is different from training images were bad. Areas which has similar edge features to vehicles were misclassified. Quantized window sizes of BING seem not to fit the one-meter resolution images. Solutions of them are follows: We need to use more training data which has different features. We need to use more sophisticated architecture of CNN which has better capability of classification. We need to optimize the parameter of BING. And to improve the performance of BING-CNN more, we need to train CNN with training data which consists of region proposals by BING.

We tested two methods and evaluated the performance, and it turned out the performance will be improved by solving some problems. We will work on that.

References

Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, Philip Torr, 2014. BING: Binarized Normed Gradients for Objectness Estimation at 300fps. IEEE CVPR.

Xueyun Chen, Shiming Xiang, Cheng-Lin Liu, and Chun-Hong Pan, 2014. Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks. IEEE Geoscience and Remote Sensing Letters, Vol. 11, Issue 10, pp. 1797-1801.

Shenquan Qu, Ying Wang, Gaofeng Meng, and Chunhong Pan, 2016. Vehicle Detection in Satellite Images by Incorporating Objectness and Convolutional Neural Network. Journal of Industrial and Intelligent Information, Vol. 4, No. 2, pp. 158-162.