FAST AIRCRAFT DETECTION IN SATELLITE IMAGES BASED ON CONVOLUTIONAL NEURAL NETWORKS

Hui Wu, Hui Zhang, Jinfang Zhang, Fanjiang Xu

Institute of Software Chinese Academy of Sciences, China
{wuhui13, zhanghui, jinfang, fanjiang}@iscas.ac.cn

ABSTRACT

Aircraft detection in satellite images is generally difficult due to the variations of aircraft type, pose, size and complex background. In this paper, we propose a new aircraft detection framework based on objectiveness detection techniques (e.g., BING) and Convolutional Neural Networks (CNN). The advantages are two folds. On one hand, we first introduce the CNN for aircraft detection, as CNN can learn rich features from the raw data automatically and has yielded a state-of-the-art performance in many object detection tasks. On the other hand, the use of candidate object regions proposed by BING achieves a high object detection rate and saves time simultaneously. Experimental results show that the proposed method is fast and effective to detect aircrafts in complex airport scenes. We also construct a dataset for aircraft detection obtained from Google Earth.

Index Terms**—** Deep learning, aircraft detection, convolutional neural networks, BING, objectness

1. INTRODUCTION

Detecting aircrafts in high-resolution satellite images is essential for military use. Although it has been studied for years and many works have been done [1-3], it is still difficult to find a generic location method and classifier to detect aircrafts in complex airport scenes.

Liu et al. [1] proposed a coarse-to-fine shape method based on edge computing to recognize aircrafts. Sun et al. [2] applied the key-points and spatial sparse coding bag-of-words model to detect aircrafts. Li et al. [3] detected aircrafts based on visual saliency computation and symmetry detection. All of the above methods are effective in their scenes, but they are based on manually engineered features.

To solve this problem, deep learning is widely used for learning intrinsic features automatically in the latest years, which has shown outstanding performance on object classification and detection [4]. Chen et al. [5] applied Deep Belief Nets with the object locating method to detect aircrafts. Their method was unsupervised and could detect tiny blurred aircrafts correctly in difficult airport images. However, their locating method was based on two certain scale sliding windows and required to compute gradients on multiple images, which made it rather time-consuming.

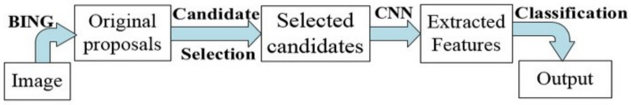


Fig. 1. Overview of the proposed target detection system.

In order to reduce candidate object windows, many objectiveness detection methods have been proposed in recent years, such as objectness [6], selective search [7], constrained parametric min-cuts [8], and binarized normed gradients (BING) [9]. Among those methods, BING [9] proposed a very fast objectness score based on image gradients while maintaining a high detection rate.

The CNN [10], which was first proposed by Yann et al, now is a state-of-the-art deep learning technique in image recognition area [11, 12]. In this paper, we propose a new aircraft detection framework based on BING and CNN as shown in Fig. 1. First, we obtain a set of original object proposals from test images by BING technique. Then, we provide a method to select candidate object proposals based on the assumption that most aircrafts in Satellite images are square and small. Finally, we feed all selected candidates to CNN for feature extraction and aircraft detection. Experimental results demonstrate that our framework can improve the detection performance and reduce the computational time simultaneously.

Our contributions are three folds. 1) We propose a new aircraft detection framework based on BING and CNN. We introduce CNN for aircraft detection for the first time, as it can learn rich features from the raw data automatically. 2) We propose a candidate selecting method based on BING to reduce the number of the proposals and save computational time. 3) We build a dataset for aircraft detection experiments by using the airport images obtained from the Google Earth. Now the manually labelled dataset is made available at https://github.com/wuhuiIOS/AircraftsDataset.

The rest of the paper is organized as follows. In section 2, we describe the overall detection framework in details. Our

Foundation item: Supported by the National Natural Science Foundation of China (NSFC No. U1435220, 41401409).

experiments and results are presented in section 3. Section 4 is the conclusion part.

2. TARGET DETECTION WITH BING-CNN

Our target detection system is shown in Fig. 1. First, we use BING technique to generate a set of original object proposals. Then we propose a method to select candidates from the original proposals produced by BING [9]. Finally, these selected candidates are fed to Convolutional Neural Network for feature extraction and prediction.

Details of the BING-CNN based target detection system are introduced as follows.

2.1. Potential object proposal by BING technique

Traditionally, sliding windows are widely used for target detection. It is time-consuming and contrast to mechanisms in the human vision system. Recently, Cheng et al. [9] proposed BING technique to produce object regions by using objectness scores.

Objects with well-defined closed boundaries share strong correlation after resizing their corresponding image windows to a small fixed size (e.g. 8×8) in the normed gradient space. The method proposed by [9] learns a generic objectness measure of image windows in a two-stage cascaded SVM framework.

In order to find objects within an image, we scan over a predefined quantized window sizes (scales and aspect ratios). Each window is scored with a single linear model w ($w \in R^{64}$), which is learnt by a linear SVM in stage Ⅰ,

$$s_l = \langle w, BING_l \rangle, \quad (1)$$

$$l = (i, x, y), \quad (2)$$

where s_l, $BING_l$, l, i and (x,y) are filter score, binarized normed gradients, location, size and position of a window respectively. Therefore, the objectness score is defined as:

$$o_l = v_i \cdot s_l + t_i, \quad (3)$$

where $v_i, t_i \in \mathbf{R}$ are coefficient and a bias terms respectively for each quantized size i. They are separately learnt by a linear SVM in stage Ⅱ. A window is more likely to contain an object when o_l is high.

The BING technique involves many parameters such as the quantized window size, and the final number of proposed bounding boxes per size (scale and aspect ratio). Such parameter values are determined by applications.

BING technique has a good generalization ability in those categories that are not used for training two stages cascaded SVM. In this paper, we use the trained model as [9] for obtaining a set of object proposals from the airport scene images at the testing stage. We set the number of proposed bounding boxes per size to 300, as the sizes of our testing images are much bigger than those of training images in [9].

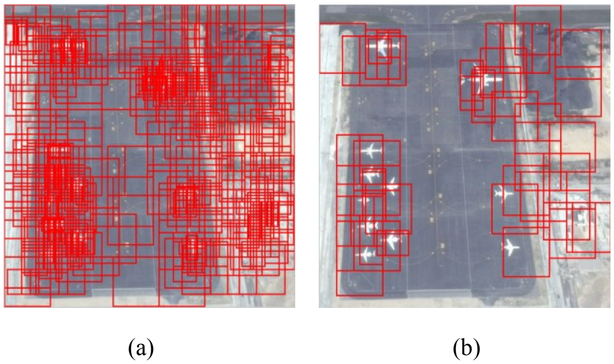


Fig. 2. (a) Original bounding boxes generated by BING. (b)Remaining bounding boxes selected by the proposed candidate-selection method.

2.2. Candidate selection

Although the number of original object windows proposed by BING is very small compared with the common sliding window paradigm, it is still very large for CNN. To tackle this problem, we propose a candidate-selection method to select candidates from the proposals extracted by BING.

BING proposes many different sizes of bounding boxes, but the majority of aircrafts in Satellite images are square and small. So we empirically select candidates by the following rules.

Rule 1: We filter out those proposals with high height/width (or width/height) ratios,

$$Ratio = \max(w,h) / \min(w,h), \quad (4)$$

$$Ratio > T_{ratio}, \quad (5)$$

where T_{ratio} is a threshold with a constant value.

Rule 2: We select proposals whose areas are within a certain range ($[T_{\min}, T_{\max}]$),

$$Area = w \times h, \quad (6)$$

$$T_{\min} \le Area \le T_{\max}, \quad (7)$$

Rule 3: We pick out proposals with higher objectness scores generated by BING,

$$o_l > o_t, \quad (8)$$

where o_t is a threshold with a constant value.

Thus, a much smaller number of candidate proposals are selected based on the rules above. Fig. 2 shows the process of candidate-selection method on an airport scene.

Computational time. We compare the performance of our candidate-selection method based on BING with object location algorithm in [5] on 26 images in our test dataset. More details about the test images can be found in the experiment part.

In our dataset, our method is able to efficiently generate almost 2000~3000 object proposals per image at milliseconds. The object location algorithm [5] needs dozens of seconds to process one image in average. All the experiment is done on the same desktop with an Intel i7-870TM CPU.

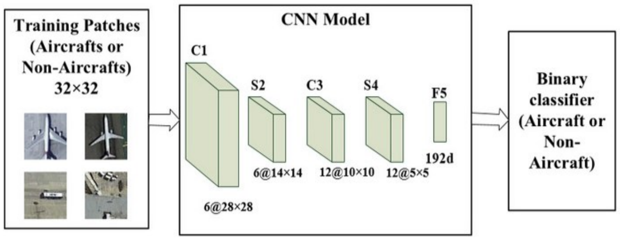


Fig. 3. Architecture and parameters of our CNN model

2.3. CNN architecture and training

A typical CNN [10] stacks convolution and pooling (or subsampling) layers in alternation into a multilayer architecture, followed by fully connected layers.

Each convolutional layer of CNN has many filters and generates different feature maps using sliding filters (or kernels) on local receptive field in the maps of the previous layer or input. The size of filters can be considered as $n \times n$ (n is smaller than the input size). Weights are shared in convolutional layers. The pooling layer shrinks the representation of the convolutional layer over rectangular regions by a constant factor. There are several pooling types, such as max-pooling and average pooling.

The overall architecture is shown in Fig. 3. We adopt the CNN containing five layers. The architecture is: C-S-C-S-F, where C, S and F represent convolutional layer, subsampling layer and full connected layer respectively. The input is a gray image of size 32 by 32 pixels and is passed on to a convolutional layer (C1) with 6 filters. The convolutional filter sizes of C1 and C3 both are 5×5. The subsampling field sizes of S2 and S4 both are 2×2. The dimension of the full connected layer is 300. The output of the last fully-connected layer is fed to a sigmoid function which produces a distribution over the class labels.

In this paper, we use small patches (aircrafts or non-aircrafts) to train our CNN by the back-propagation algorithm on CPU. Initial weights are set by random values, and initial biases are set to zero. We set learning rate to 0.5, batch size to 100.

2.4. Target detection and final output

After candidates selecting, all the object proposals are normalized to size 32×32. Then we feed them into CNN for feature extraction and classification. We don't adopt multi-scale analysis so that the image can be processed quickly.

During the detection phase, we apply a standard non-maximum suppression technique to fuse the multiple overlapping detections for the same target.

3. EXPERIMENTS

3.1. Data set

Train dataset. We collect 500 positive patches and 5000 negative patches from the Google Earth. The positive samples are image patches with an aircraft. The negative samples are randomly sampled patches of the airfield background. The original patch size is 64×64 pixels. To improve the robustness, we enlarge the positive samples by rotating each small aircraft patch 4 times, 90 degrees per time. The extended training dataset has 2000 positive samples.

Test dataset. We evaluate our aircraft detection system on 26 test images. The image sizes vary from 565×369 to 1484×865 pixels. There are 453 aircrafts in the test dataset.

3.2. Results

Detection is reported as positive if more than half of the ground truth bounding box has been detected. Some overlapped false alarms are fused into one alarm. False Alarm Rate (FAR), Recall Rate (RR) and Precision Rate (PR) are defined as:

$$FAR = \frac{number\ of\ False\ Alarms}{number\ of\ aircrafts} \tag{9}$$

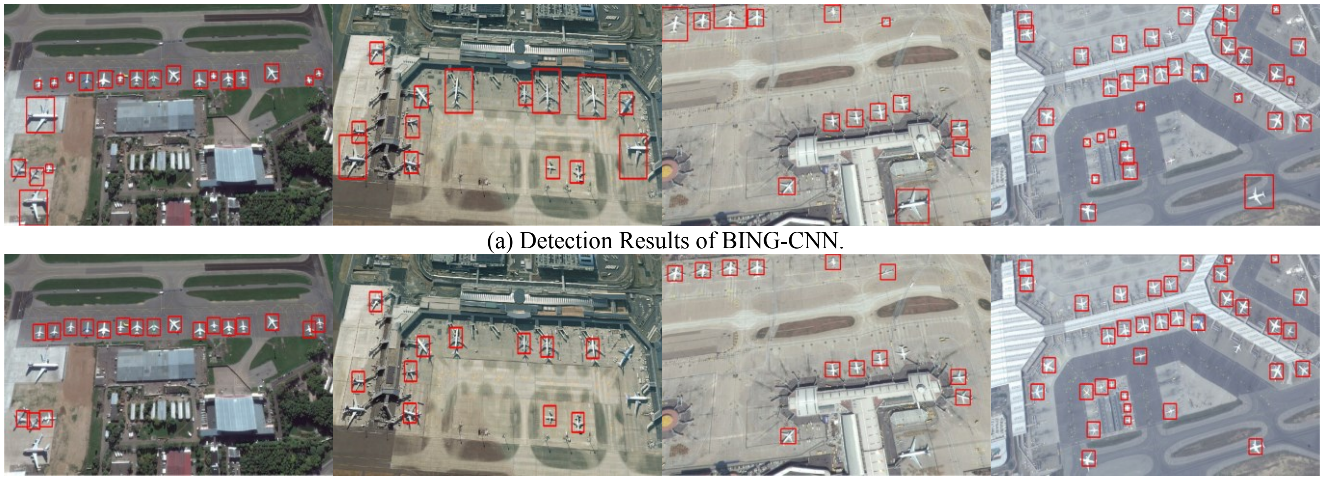
$$RR = \frac{number\ of\ detected\ aircrafts}{number\ of\ aircrafts} \tag{10}$$

$$PR = \frac{number\ of\ detected\ aircrafts}{number\ of\ detected\ \ objects} \tag{11}$$

3.2.1 Detection performance

Fig. 4 shows the detection results of two different methods on partial test images. In most cases, BING-CNN performs better. In this paper, we abbreviate object location method to Location for convenience. When the sizes of objects vary largely, Location-DBN [5] has high misdetection rate. Location-DBN [5] performs better than BING-CNN in rare cases (the last column of Fig. 4). BING-CNN missed two small aircrafts in the last column of Fig. 4(a), since the parts of the aircrafts' fuselages are ambiguous so that BING fails to generate proposals containing them.

Table 1 shows False Alarm Rates of different methods on all test images when given the Recall Rates (65%, 70%, 75%, 80% and 85%). We compute HOG features as [13]. BING-CNN outperforms HOG-SVM and Location-DBN [5]. Location-DBN [5] is better than HOG-SVM. The 3rd and the 4th columns of table 1 are the False Alarm Rates of DBN when using two and three scale sliding windows to locate objects respectively. The performance of Location-DBN [5] is improved explicitly when using three scales compared to only using two scales. When the sizes of objects vary largely, a few different and elaborate scales are needed to locate all objects. It indicates that the Location method in [5] is sensitive to the sizes of targets. BING is robust to the object's size changes.



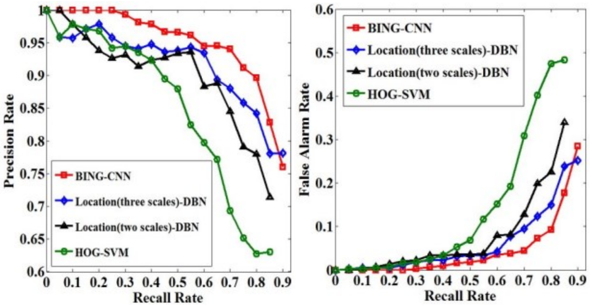
(a) Detection Results of BING-CNN.

(b) Detection Results of Location (two scales) -DBN [5].

Fig. 4. Illustrating aircraft detection results of two different methods. In most cases, BING-CNN performs better.

Table 1. False Alarm Rates of different methods

Given Recall Rates (RR)	False Alarm Rates (FAR)			
	HOG-SVM	**Location (two scales) -DBN [5]**	**Location (three scales) -DBN [5]**	**BING-CNN**
65%	19.21%	8.17%	7.73%	**3.75%**
70%	30.91%	12.80%	9.49%	**4.42%**
75%	40.18%	19.87%	12.36%	**7.28%**
80%	47.46%	22.52%	15.01%	**9.27%**
85%	48.34%	34.00%	23.84%	**17.66%**



(a) (b)

Fig. 5. (a) Precision-Recall Curves of different methods. (b) False Alarm-Recall Curves of different methods.

Table 2. Average detecting time on test images

Method	HOG -SVM	Location (two scales) -DBN [5]	Location (three scales) -DBN [5]	**BING -CNN**
Time	1.073s	171.254s	201.983s	6.414s

Fig. 5 shows Precision-Recall curves and False Alarm-Recall curves of different methods. The results show that the BING-CNN detector outperforms other two methods.

3.2.2 Detection time

Table 2 shows average detecting time per image of different methods. Generally speaking, our method enjoys relatively better time efficiency and detection performance among the given methods. The most computationally expensive part of [5] is locating objects. We only use one thresholding image when computing the detection time of Location-DBN [5]. We do not find a substantial difference in detection performance of DBN between using one and three thresholding images when locating objects.

All experiments are done on the same desktop with an Intel i7-870TM CPU. All methods are implemented by Matlab except that BING is implemented by C++.

4. CONCLUSION

In this paper, we propose a framework based on BING and CNN to fast detect aircrafts in Satellite Images. CNN can learn features from the raw images and is invariant to small rotation and shift. Our candidate selecting method based on BING technique produces a smaller number of candidates for prediction than sliding windows, which reduces the computational time of the detector. Experiments show that our method yields a good performance on aircraft detection. The proposed method is not limited to aircraft detection. In our future works, we will be devoted to improving the performance of the method and applying it to other target-detection researches.

5. REFERENCES

[1] G. Liu, X. Sun, K. Fu, and H. Wang, “Aircraft recognition in high-resolution satellite images using coarse-to-fine shape prior,” GRSS, vol. 10, no. 3, pp. 573-577, 2013.

[2] H. Sun, X. Sun, H. Wang, Y. Li, and X. Li, “Automatic target detection in high-resolution remote sensing images using spatial

sparse coding bag-of-words model," GRSS, vol. 9, no. 1, pp. 109-113, 2012.

[3] W. Li, S. Xiang, H. Wang, and C. Pan, "Robust airplane detection in satellite images,", in ICIP, 2011, pp. 2821-2824.

[4] Q.V. Le, "Building high-level features using large scale unsupervised learning." in ICML, 2012, pp. 8595-8598.

[5] X. Chen, S. Xiang, C.L. Liu, and C.H. Pan, "Aircraft detection by deep belief nets," in ACPR, 2013, pp. 54-58.

[6] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," IEEE TPAMI, vol. 34, no. 11, pp. 2189-2202, 2012.

[7] J.R. Uijlings, K.E. van de Sande, T. Gevers, and A.W. Smeulders, "Selective search for object recognition," in IJCV, vol. 104, no. 2, pp. 154-171, 2013.

[8] J. Carreira, and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation." in CVPR, 2010. pp. 3241-3248.

[9] M.M. Cheng, Z. Zhang, W.Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in CVPR, 2014, pp. 3286 - 3293.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," P IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[11] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," in NIPS, 2012, pp. 1097-1105.

[12] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in CVPR, 2013, pp. 1312-6229.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detector," in CVPR, vol. 1, pp. 888-893, 2005.