15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

A Preliminary Study on Extracting Objects in Sketches

Bo Huang Jiansheng Chen
Department of Electrical Engineering, Tsinghua University
Beijing, China
huangb14@mails.tsinghua.edu.cn jschenthu@mail.tsinghua.edu.cn

Abstract

Humans have the incredible ability to interpret complex sketch images, but it remains a challenging task for computers to do the same thing. Researchers have made great progress on nature image detection and recognition, but little research has been done on object detection in sketch images. In this paper, we demonstrate that object extraction can be possibly conducted using a Single-Shot Multibox Detector(SSD) framework, without the guidance of segmentation information or user interaction. We train and test our model with a synthetic dataset based on TU-Berlin sketch dataset. Experiments on the synthetic dataset show reasonable object detection and recognition results in sketch images.

1 Introduction

Through human civilization, sketches have been used as a universal way of communication. It is simple but effective. People can easily understand each other by sketches even if they speak completely different languages. With the popularization of touch screens in the last decade, the sketch is becoming a natural way for man-machine interaction, and researches on the sketch image processing and understanding are getting increasing attention. A wide range of applications have been explored, including sketch recognition [1] [7] [5] [11] , sketch-based image retrieval [2] [3], sketch-based 3D model retrieval [10] [8], and sketch-based image generation [6].

Compared to natural images, sketch images are significantly different in many aspects. First, sketches are highly abstract and mostly symbolic. For example, a human figure can be simply painted as a ”stick-man”. Second, sketch images usually contain much more variations. For example, the same object can be drawn in many different ways under different circumstances. What is more, different from natural images, variations in sketches may even violate geometric and physical principles. Third, sketches images are generally sparse. Visual clues such as texture and color are usually missing in sketches. All these lead to high intra-class variations and inter-class ambiguities, making analyzing sketch images an extremely challenging visual task.

There exists a number of previous research works in sketch recognition which tries to identify predefined glyphs in narrow domains. To achieve such a task, Eitz et al. proposed to extract hand-crafted geometric features and used SVM as classifiers [1]. To further improve the classification accuracy, Li et al. introduced the multiple kernel learning in [5]. Schneider et al. assumed a stochastic distribution based on which specific fisher vectors can be generated to improve the discriminative power of sketch image features [7]. More recently, Yu et al. used a Convolutional Neural Network (CNN) for representation learning and a joint Bayesian fusion method for classification [11]. All these methods assume that there is only one object in the sketch to be recognized.

Several interesting approaches have also been proposed on sketch-based image retrieval [2][3]. A common assumption in these approaches is that, in some well-designed feature space, sketched objects are similar to their real-world counterparts. These methods do not limit the number of objects in the sketch. The problem is that these methods do not care what the sketch is, only what it looks like. In other words, it is the geometric similarity rather than object category that are considered in these methods. For example, if the input sketch is an alarm clock, the retrieval results may be considered appropriate as long they contains round objects.

Another related research topic is the sketch-based 3D model retrieval which might be more difficult than the 2D image retrieval considering that its effectiveness relies more heavily on the object segmentation accuracies Existing methods on this topic usually avoid explicit object segmentation in sketches. For example, Xu et al. assumed that the input sketch had been pre-segmented, say, by interactive operations [10]. Wang et al. even simply assumed that there was only one object in the input sketch [8] .

In this work, we study the problem of automatically extracting objects from complex sketch images. By ”complex” we mean that there might be multiple objects in one sketch image, and by ”automatically” we indicate that no segmentation information or user interaction are available. If two objects in a sketch are close to, or even overlap with each other, it is generally very difficult to tell which pixel belongs to which

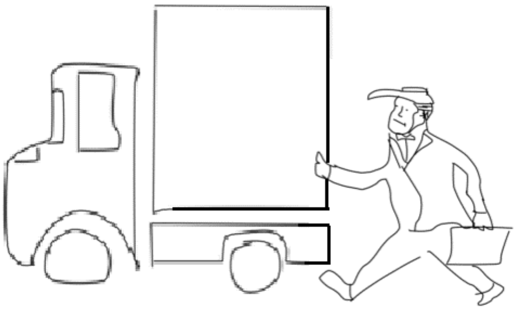


Figure 1. An example showing the difficulty in sketch object extraction.

object due to the lack of visual cues like color or texture. A typical example is shown in Fig. 1, in which a man is standing next to a truck. It can be observed that part of the human body overlaps with the compartment that is disconnected from the other parts of the truck. Therefore, it is hard to tell wether the compartment is part of the man or part of the truck using only low-level features like distances or connectivity.

In this paper, we study the possibility of object detection in sketch images using the Single-Shot Multibox Detector(SSD) framework [9]. Our work has four key features that distinguish it from existing works. First, we do not use any hand-crafted features. Second, we do not limit the number of objects in sketches. Third, we do not limit the scale variation of objects. Finally, no segmentation information or user interaction are needed. The rest of this paper is organized as follows. Section 2 gives the detailed introduction of our model structure. Section 3 introduces the dataset and illustrates experiment results. Section 4 concludes our work and discusses possible extensions.

2 Methodology

We adopt the Single-Shot Multibox Detector (SSD) framework proposed in [9] in our system. SSD is an object detection framework based on a feed-forward CNN, it discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales. At prediction time, SSD produces adjustments to each default box to match the object shape, and generates scores for the presence of each object category in the box. SSD is quite simple compared to previous object detection frameworks likek Faster R-CNN, as it eliminates proposal generation and subsequent pixel or feature resampling stages and combines all computation in a single network. By doing so, SSD enables the simultaneous learning of the object shape regression and the object category classification. It has been verified to be both fast and accurate, and allows end-to-end training. The overall architecture of our proposal is shown in Fig. 2.

The SSD framework usually contains two major components, a base network and a detection network. The base network is pre-trained on the image classification task and is used as feature extractor in SSD. Wei et al chose VGGnet[4] as their base network. However, VGGnet is primarily designed for natural images, and contains over 130 million parameters, making it really hard to converge on sketch images. In this paper, we use the Sketch-A-Net [11] as the base network. We convert all the fully-connected layers fc6, fc7, fc8 to convolutional layers, and further remove all the dropout layers in the Sketch-A-Net considering that their regularization functions are no longer need. Compared to CNNs designed for natural images, the Sketch-A-Net structure is optimized for sketches. For example, Sketch-A-Net uses a large first layer filter size of 15x15, which helps to capture more structured context considering the high sparsity in sketches.

The detection network are mainly convolutional layers added to the base network to produce feature maps and predict detection results. To detect objects on multiple scales, the original SSD network uses feature maps from not only the low-level layers like conv5_3, but also high level layers like fc7. In this paper, we use a slightly different network structure. We do not use low-level feature maps, because their receptive fields are rather limited, they can only capture strokes and conjunctions. Only high-level feature maps provides global information are used for object detection. The layers we eventually use for predictions are relu7, relu8, relu9_2, relu10_2 and relu11_2. By discarding high dimensional low-level feature maps, we also accelerate the training process by approximately 3 times. Details of our model structure are demonstrated in Table 1 in which the upper part shows the base network structure while the lower part shows the detection network struture.

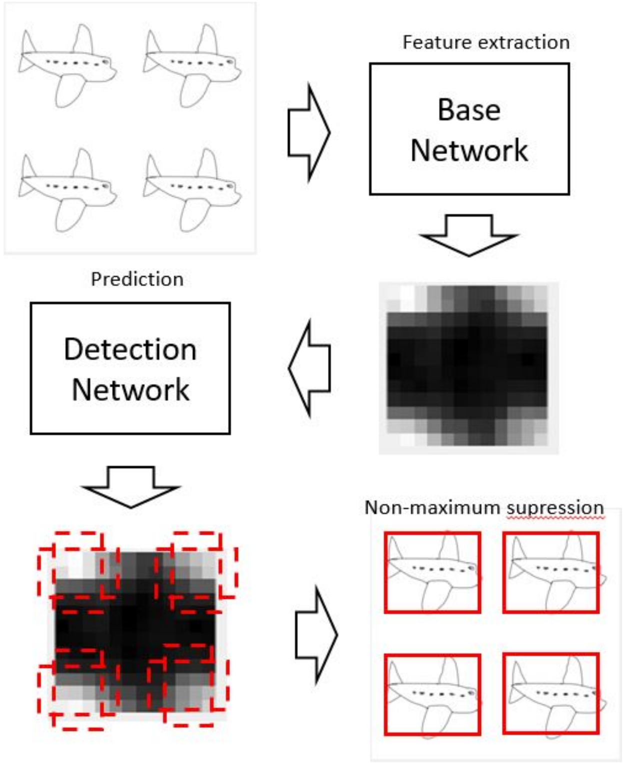


Figure 2. Overall flowchart of sketch object detection and recognition.

3 Experiments

There has not been a sketch-oriented object detection dataset yet. However, unlike natural images which requires mountains of work on data annotation, it is rather easy for sketch images to build a object detection dataset. We synthesized a dataset based on TU-Berlin sketch dataset [1], which is by far the largest and most commonly used free-hand sketch recognition dataset. We generated 20000 images as our training set, and 2000 images as validation set. Each image is of size 512x512, and contains 3 random selected object instances. Objects in the same sketch are randomly placed with two geometric constraints. First, any two objects must be far enough to avoid too much overlap. We simply require the overlapping area of two objects to be less than 20% of the total object area. Second, the bounding box of each object must not exceed the image boundary. Because as we will discuss later, we

Table 1. Detailed network structure in this paper.

Layer	Channel	Size	Stride	Pad
conv1	64	15x15	3	0
relu1	-	-	-	-
maxpool1	-	3x3	2	0
conv2	128	5x5	1	0
relu2	-	-	-	-
maxpool2		3x3	2	0
conv3	256	3x3	1	1
relu3	-	-	-	-
conv4	256	3x3	1	1
relu4	-	-	-	-
conv5	256	3x3	1	1
relu5	-	-	-	-
maxpool5	-	3x3	2	0
conv6	512	7x7	1	0
relu6	-	-	-	-
conv7	512	1x1	1	0
relu7	-	-	-	-
conv8	250	1x1	1	0
relu8	-	-	-	-
conv9_1	128	1x1	1	0
relu9_1	-	-	-	-
conv9_2	256	3x3	1	1
relu9_2	-	-	-	-
conv10_1	128	1x1	1	0
relu10_1	-	-	-	-
conv10_2	256	3x3	1	1
relu10_	-	-	-	-
conv11_1	128	1x1	1	0
relu11_1	-	-	-	-
conv11_2	256	3x3	1	1
relu11_2	-	-	-	-

should avoid dealing with incomplete sketches.

We basically use the same training technique as Wei et al described in [9], with two minor changes: higher matching threshold, harder negative mining. During training, SSD needs to decide which default box corresponds to a ground truth detection, and train the network accordingly. Wei et al. [9] choose to match any default box with jaccard overlap with the ground truth box higher than a threshold of 0.5, which might be a good choice for natural images, but not for sketches. The problem is that sketches are usually highly abstract and sparse, and it will be really hard to recognize if an object is incomplete. For example, even if a natural image only captures a small part of a banana, it can still be correctly recognized with high probability according to its color and textural details. However, if such a situation happens in a sketch image, the banana might be represented by just a few meaningless curved lines which could belong to any other objects containing curved boundaries. These incomplete matched boxes will bring heavy noise to the network, making it really hard to converge. In our experiment, we set the threshold to 0.8.

As could be expected, after the matching step, most of the default boxes are negative. And due to the sparse nature of sketches we discussed above, a lot of ”positive” boxes are quite noisy. So it’s really important to keep a proper ratio between the negative and positive boxes. In our experiment, we set the ratio to 1 instead of 3.

During training, we use a batch-size of 32, and a initial learning rate of 0.001. The learning rate is reduced by 0.8 after every 20 epochs. As for model parameters, we freeze the first seven convolutional layers of the base network during the whole process, because it’s pretty much well-trained already. As a matter of fact, the whole network would quickly become saturated if we don’t freeze the base network.

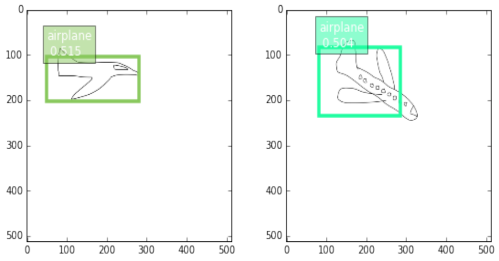
Sample results of our work are demonstrated in Fig. 3 - Fig. 5. Predictions are shown in colored rectangles. Different colors represents different classes, and scores on the upper-left shows how confident we are about the prediction. Fig. 3 shows that a single object could be easily captured by our model, even when the scale of objects are changing. Fig. 4 shows that objects can still be captured when there is more than one target to detect. Fig. 5 shows that the model can response to objects from different categories at the same time. Fig. 5(c) and Fig. 5(c) further shows that objects can still be successfully detected even if they obviously overlap with each other.

Although the result seems promising, it’s far from perfection. As can be shown in Fig. 4 and Fig. 5, the bounding boxes produced are not as accurate as we expected, and it is far worse than that on natural image datasets like Pascal VOC. Again, we blame the sparse nature of sketch images. In a natural image, most regions have color and textures, together they provide valuable information about how the network should converge. However, in a sketch image, most of the pixels are 0, which provides no information about classification nor shape regression. Even if one default box is ”lucky” enough to capture a part of an object, due to the large intra-class variation of sketch images, it is still hard to tell ”what” or ”where” the object is.

Another phenomenon we discovered is that the mAP metric drops rapidly when the number of categories increases. Which might mean that object detection in sketch images can not easily scale up to more classes. The details are shown in Table 2. In our experiments, the network convergence becomes really slow as soon as the number of classes reaches 20.

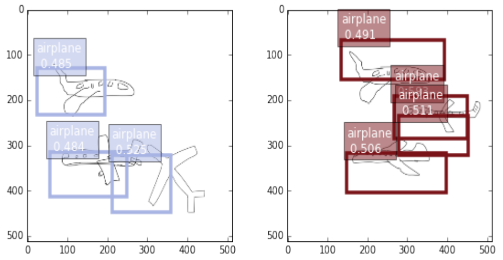
4 Conclusion

In this paper we study the possibility of using a Single-Shot Multibox Detector model to detect objects in sketch images. We demonstrate that although SSD has been proved to be an effective general detection



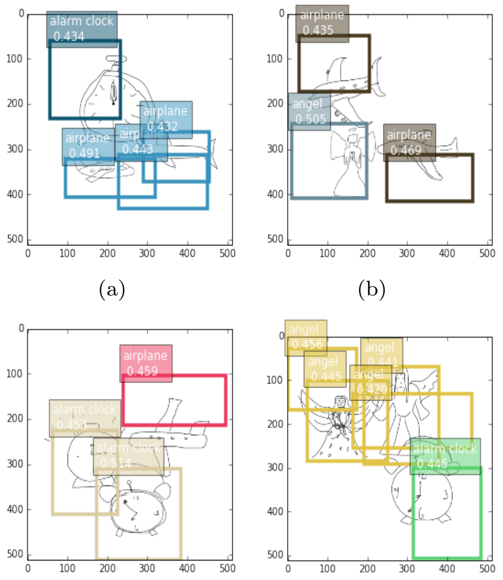
(a) (b)

Figure 3. Object detection results when there is only one object in the sketch.



(a) (b)

Figure 4. Detection when there is multiple object of the same class.



(a) (b)

(c) (d)

Figure 5. Detection when there is multiple object of different classes.

framework for natural images, its application on sketch images is not straight forward. This is largely due to the extreme sparsity and inter-class variations in sketch images.

However, through some preliminary experiments, we reveal that extracting objects from complex sketch images using deep learning framework is possible after careful tuning of the network architecture as well as training settings. Experimental results has shown that successful object detection and recognition can be realized even for complex sketch images containing multiple and even overlapping objects.

Compared to prior works on sketch images, our model does not rely on hand-crafted features, and does not limit the number of objects. It is a meaningful step towards understanding complex sketch images. We believe our work opens many possibilities for further exploration. Following this work, more applications, such as sketch captioning and retrieval, could be explored.

Table 2. mAP drops rapidly as number of classes increases. mAPs are all measured after 5 epochs.

number of object classes	mAPl
1	0.5992
3	0.4813
5	0.4182
10	0.3754

5 Acknowledgment

This work was supported by the National Natural Science Foundation of China (61673234), and the Tsinghua University Initiative Scientific Research Program (20131089382).

References

[1] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects. *ACM Transactions on Graphics*, 31(4):44, 2012.

[2] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE Trans. on Visualization and Computer Graphics*, 17(11):1624–1636, 2011.

[3] Rui Hu and John Collomosse. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *Computer Vision and Image Understanding*, 117(7):790–806, 2013.

[4] Andrew Zisserman Karen Simonyan. Very deep convolutional networks for large-scale image recognition. 2014.

[5] Yi Li, Timothy M Hospedales, Yizhe Song, and Shaogang Gong. Free-hand sketch recognition by multikernel feature learning . *Computer Vision and Image Understanding*, 137:1–11, 2015.

[6] Yonggang Qi, Jun Guo, Yizhe Song, Tao Xiang, Honggang Zhang, and Zhenghua Tan. Im2sketch: Sketch generation by unconflicted perceptual grouping. *Neurocomputing*, 165:338–349, 2015.

[7] Rosalia G Schneider and Tinne Tuytelaars. Sketch classification and classification-driven analysis using fisher vectors. *ACM Trans. on Graphics*, 33(6):174, 2014.

[8] Fang Wang, Le Kang, and Yi Li. Sketch-based 3d shape retrieval using convolutional neural networks. pages 1875–1883, 2015.

[9] Dumitru Erhan Christian Szegedy S Reed Chengyang Fu Alexander C Berg Wei Liu, Dragomir Anguelov. Ssd: Single shot multibox detector. 2016.

[10] Kun Xu, Kang Chen, Hongbo Fu, Weilun Sun, and Shimin Hu. Sketch2scene: sketch-based co-retrieval and co-placement of 3d models. *ACM Transactions on Graphics*, 32(4):123, 2013.

[11] Qian Yu, Yongxin Yang, Yizhe Song, Tao Xiang, and Timothy M Hospedales. Sketch-a-net that beats humans. 2015.