Reducing external information in crowdsourcing for salient object labeling

Abstract

This paper probes into the factors which influence crowdsourcing workers' efficiency on salient object labeling. Previous image labeling done by crowdsoucing workers usually took much time on filtering bad workers or reviewing. We think researchers are more willing to collect high-quality data with less time and cost. In that case, we designed the experiment by publishing 3 groups of pictures with different levels of opacity on Amazon Mechanical Turk. Through the data retrieved from AMT we concluded that overlay with 80% opacity can improve the efficiency of crowdsourcing workers.

Introduction

In the computer vision research history, datasets have played a significant role. Researchers apply dataset to train machine to better understand pictures. A high-quality dataset can be build by object labeling. Now, lots of object labeling work comes from crowdsourcing platforms such as Amazon Mechanical Turk. The problem is that to label a great number of objects may take workers too much time and the accuracy is not always satisfying. So, we want to find a method to improve the efficiency of the labeling procedure.

There are some famous previous work about image annotation and attribute labeling. LabelMe[8] lets people outline any object in the images and then label them. Peekaboom[10] is a web-based game involving two players to help computers locate objects in images. The basic rule is that one player reveals part of the image to the other player so that the other player can guess the associated word and enter it. MSCOCO[7] labels objects by selecting categories icons. From these work, we can see that the most common labeling task is to show an image on the screen and let workers label or describe it by choosing or typing categories. The other common thing among these works are they all used overlays during their labeling work. LabelMe add overlays in different color on the objects being labeled. Peekaboom add overlays on some part of the objects to increase the games difficulty. Though there are some previous works using overlays in object labeling work. We want to use overlays differently. We add them outside the objects to stress the most significant part.

Images contain lots of content. Most of the content are not necessary for workers to do labeling work and they can even affect workers' judgment. So, we assume that low accuracy may be caused by external information and try to improve the efficiency by reducing external information.

According to the above, we propose an idea to improve the previous solutions. Our pipeline is to add a grey overlay on the unselected part outside the bounding box to remove external information and then let workers label the object they can see in the image. We hope the labelling work can be more efficient and accurate according to this change.

Figure 1: Using overlays to highlight the object needed to be labeled

Related Work

Crowdsourcing object annotation collection Nowadays connecting people online to complete multiple annotation tasks is a common choice for researchers who aim to build a large image dataset. LabelMe[8] is an open annotation tool, where each worker can upload their own images and label objects in the images. However, after some users work, images are full of segmentations and annotations, which may confuse new users. The annotation pipeline of MSCOCO[7] is much more different from LabelMe. It requires workers first label the categories present in the image, then locate and mark all instances of the labeled categories, and finally segment each object instance. Each image was labeled by 8 workers to control the quality during second step. Other workflows of object detection[6][9] include quality verification task and coverage verification task to improve the quality of annotations and save money. Unlike prior work, we add overlay on images to improve the efficiency of crowdsourcing workers and we dont need too many workers annotate the same image or verify other peoples work.

Salient object annotation Some companies are now developing assistive technologies for visually impaired people such as Facebook and Microsoft. This brings another kind of annotation: salient object annotation. Usually the pipeline is that first detect prominent objects as many as you can, then add captions or descriptions to them and finally use these data to train algorithm. Most of the work[1][5][3] focuses on improving algorithm and doesnt pay too much attention to the efficiency of annotation. However, our work is different. We think from the perspective of workers and implemented 11 groups in a experiment to find the best solution for annotation.

Object overlay Some systems apply overlay to images to clearly show objects. LabelMe[8] adds different color to overlays to represent different objects. In Peek and Boom[10], Boom gets an image along with a word related to it, and must reveal parts of the image for Peek to guess the correct word. To help Peek, Boom can ping parts of the image by right-clicking on them. CUB-200-2011[11] adds overlays to external part when workers draw bounding boxes around birds.

Information overload Nowadays people are exposed to massive information from many areas everyday. This information may influence people's judgments and even lead to damage to health. Bawden[2] points out "information overload" is usually taken to represent a state of affairs where an individuals efficiency in using information in their work is hampered by the amount of relevant, and potentially useful, information available to them. Various psychological conditions are associated with this, such as attention deficit trait[4], a distractability and impatience due to too much mental stimulus.

Our work more closely relates to CUB-200-2011 because we also add overlays to external information. However, this process aims to improve the efficiency of annotation instead of detection. We also try different opacities of overlay to figure out the best solution.

Methods

Workflow

Our system begins with collecting images. Each image includes at least one salient object and it can be recognized by people.

The first task is to detect the location of the object in the image using bounding boxes. Once the correct and accurate bounding box is drawn, the starting point, width and height can be acquired and stored in a XML file. This information can be used to pre-process images. The procedure of preprocessing includes drawing bounding boxes, adding overlays with different transparency outside object on images. These images which have been pre-processed will be shown on the AMT interface. Three groups of workers are required to label objects in different kinds of images.

Image processing

Image processing work consists of two parts. The first part is loading bounding boxes information on original images. Each salient object need to be covered with one tight bounding box. (see Figure-2). The second part is adding corresponding overlays on original images by using the coordinate value acquired from bounding boxes (see Figure-2). The opacity value of the overlays will be set as 80% and 100%. Figure-1c is an example of overlay with 80% opacity. We use light grey color because it does not appear on the image in large area. And grey causes little distraction.

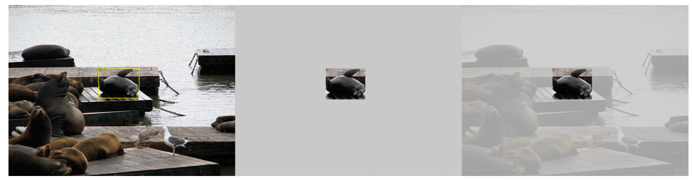

Figure 2: Bounding box; Overlays with 100% opacity; Overlays with 80% opacity

Image labelling

In image labeling task, one image processed before will show on AMT interface each time. Workers need to read the instruction and follow it to finish the labeling task. An example and four tips are also provided under the instruction to assist worker. Workers need to input the name of each object in text field. They can view following images by clicking the next button or click the previous button to review and change their answer. If workers have any feedback, they can type in comment area. Submit button will not show on the bottom of the page until they finish all the five images. The user interface shows in Figure-3. All tasks use the same user interface with different image combinations.

Experimental Design

The system is deployed on Amazon Mechanical Turk(AMT).

Dataset

We downloaded images from ImageNet dataset. ImageNet provides images and annotations of object bounding box for over 3000 synsets. We selected five categories: bear, sea lion, police dog, bedroom and office. We picked these categories because they each represent common problems in image annotation: bear is a large-sized natural object; sea lion and police dog are both medium-sized or small-sized natural objects; bedroom supplies are large-sized handmade objects; office supplies are medium-sized or small-sized objects.

The image annotations are saved in XML file. We parsed the annotations to grab width, height and starting point of the bounding box and use those annotations to add bounding boxes and overlays on images.

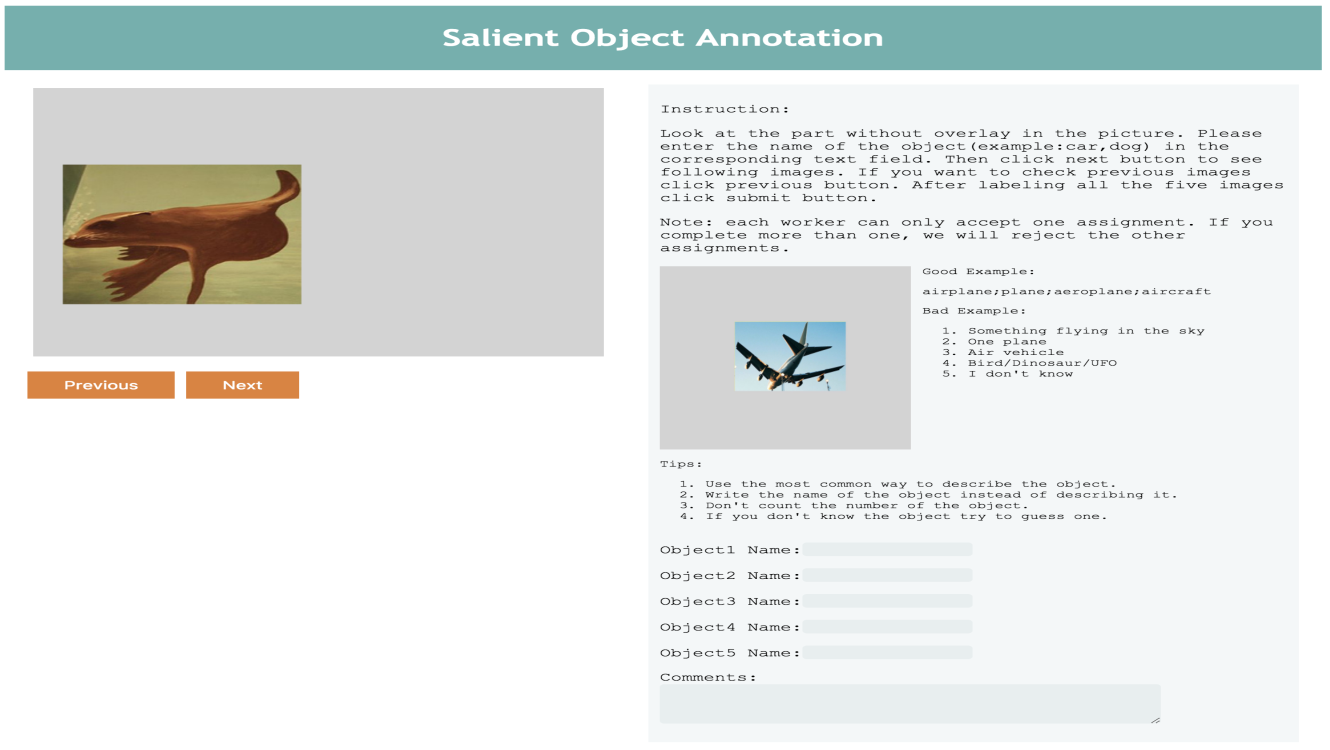

Figure 3: Our online, interactive object labeling user interface.

Experiment procedure

We selected 20 images from each category. The experiment includes 3 groups. Group 1 is control group containing 100 different images with bounding boxes. Group 2 are test groups containing 100 different images with overlays covered. The opacity of the overlays in group 2 are 80% opacity. And Group 3 have the overlays without transparency. We published 200 assignments each group. In each assignment, workers need to annotate 5 images.

We request that the approval rate of previous AMT work of each worker should over 90%. Each worker can only accept one assignment in each group. If they complete more than one assignment, we will reject all the others.

In the end, we evaluate the overall quality and cost time of each group.

Experimental Results

Measuring Accuracy and Cost Time of each Experiment After retrieving results from Amazon Mechanical Turk, we rejected duplicated assignments completed by the same worker and only kept their first assignments. Overall, we collected 43 assignments for control group, 47 for group 2 and 32 for group 3. Then we analyzed average cost time and accuracy of each group.

Figure-4 shows average cost time and accuracy of each group. Adding overlay with 80% opacity increases accuracy by 2.13% and reduces cost time by 23.9% compared to control group. However, overlay with 100% opacity decreases accuracy by 1.39% and increases cost time by 12.71%. From this result, we can conclude that overlay with 80% opacity is the best solution among the three groups, which is in line with our hypothesis. 20% transparency can not only make workers focus on the object required to be labelled but also provide context information for them. Though 0 transparency can make worker more concentrate on the objects, it removes background information, which might be very important for some images.

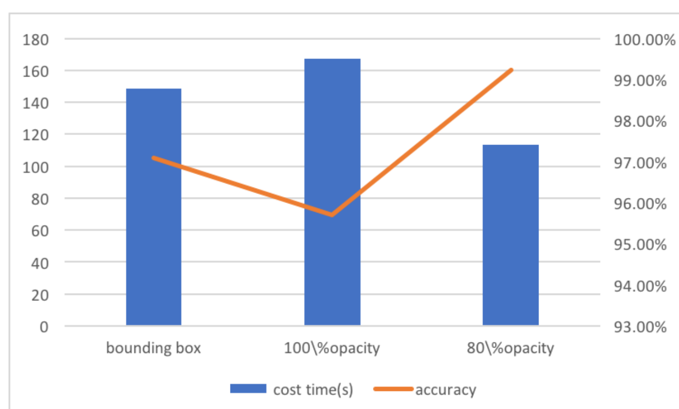

Figure 4: Accuracy and Time Cost of Each Group

Demonstrate Variation among Different Types of Categories In experimental design session we divided all the categories into 4 types: large-sized natural object[A]; medium-sized and small-sized natural object[B]; large-sized handmade object[C]; medium-sized and small-sized objectD. So in this section we discuss the accuracy of each type respectively.

From Figure-5 we can clearly conclude that the accuracy of large-sized objects no matter natural or handmade outnumbers that of small-sized or medium-sized; natural objects are more easily annotated by crowdsourcing worker compared to handmade objects. What's more, people tend to give a more detailed label when evaluate natural objects. For example most of them labeled with polar bear or grizzly instead of just bear.

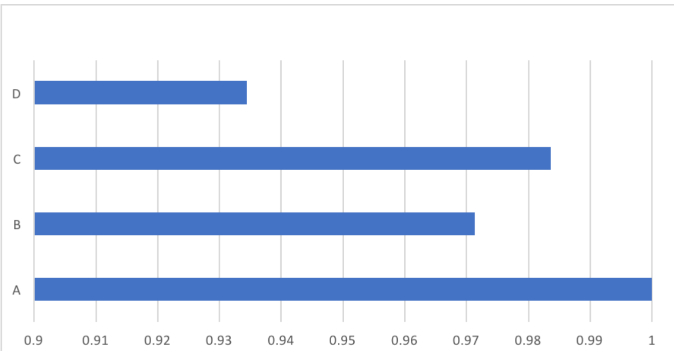

Figure 5: Accuracy of Each Type

Discussion Our discussion above proves that overlay with a certain level of opacity around salient object can increase the efficiency of crowdsoucing workers. Besides this solution can also benefit other areas of computer vision. If an image includes privacy information or even uncomfortable information such as violence or nude, we can apply overlay to reduce that kind of information.

Conclusion

Our work provides a promising solution to improve the efficiency of AMT workers for research in computer vision field. Building large image dataset is an extremely important step before algorithm training. Amazon Mechanical Turk is a common choice for a great deal of annotation tasks, however, the ability of each worker is unknown. To acquire good quality data, researchers have to filter bad workers or employ more workers to complete the same task and verify other workers job. Since adding overlay with 80% opacity can largely improve the efficiency of workers, researchers can collect high-quality data within less time and finally save their funding.

Although the result of our work proves our hypothesis, it has some limitations. First, we specified that each worker can only complete one assignment in the instruction of AMT interface, but the result shows that some workers still ignored it and this increased our workload. Second, with limited time we only designed three groups in the experiment. One future research direction is including more groups such as overlay with 20%, 40% and 60% opacity. Future work also includes how to add overlay to improve the efficiency of labeling multiple objects within one image.

References

[1] Radhakrishna Achanta et al. "Salient region detection and segmentation". In: *Computer Vision Systems* (2008), pp. 66–75.

[2] David Bawden and Lyn Robinson. "The dark side of information: overload, anxiety and other paradoxes and pathologies". In: *Journal of information science* 35.2 (2009), pp. 180–191.

[3] Ali Borji et al. "Salient object detection: A benchmark". In: *IEEE Transactions on Image Processing* 24.12 (2015), pp. 5706–5722.

[4] EM Hallowell. *Overloaded circuits: why smart people underperform in Harvard Business Review on Managing Yourself*. 2005.

[5] Huaizu Jiang et al. "Automatic salient object segmentation based on context and shape prior." In: *BMVC*. Vol. 6. 7. 2011, p. 9.

[6] Ranjay Krishna et al. "Visual genome: Connecting language and vision using crowdsourced dense image annotations". In: *International Journal of Computer Vision* 123.1 (2017), pp. 32–73.

[7] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[8] Bryan C Russell et al. "LabelMe: a database and web-based tool for image annotation". In: *International journal of computer vision* 77.1 (2008), pp. 157–173.

[9] Hao Su, Jia Deng, and Li Fei-Fei. "Crowdsourcing annotations for visual object detection". In: *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*. Vol. 1. 2. 2012.

[10] Luis Von Ahn, Ruoran Liu, and Manuel Blum. "Peekaboom: a game for locating objects in images". In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM. 2006, pp. 55–64.

[11] Catherine Wah et al. "The caltech-ucsd birds-200-2011 dataset". In: (2011).