

TRANSPARENT OBJECT DETECTION USING REGIONS WITH CONVOLUTIONAL NEURAL NETWORK

¹ Po-Jen Lai (賴柏任), ² Chiou-Shann Fuh (傅楸善)

¹ Dept. of Electrical Engineering,
National Taiwan University, Taiwan

² Dept. of Computer Science and Information Engineering,
National Taiwan University, Taiwan

E-mail: berlin579@gmail.com, fuh@csie.ntu.edu.tw

ABSTRACT

In this paper, we aim to develop a transparent object detection algorithm which can detect the location of transparent objects in color image. Due to the special characteristics of transparent objects, very few vision methods exist to identify them. To achieve our goal, we try to use deep learning based method to recognize the transparent object in color image. Experimental results show that our algorithm can achieve good transparent object retrieval results.

Keywords *Transparent Object Detection; Convolutional Neural Network; Deep Learning*

1. INTRODUCTION

Transparent objects are very common in our environment, from our home and restaurants to laboratories. However, transparent materials are difficult to detect due to the appearances of transparent objects change over different backgrounds, their edges are implicit and contain strong highlights. In this paper, we explore algorithms for better transparent object detection.

As for the outline of this paper, Section II describes the previous work have been done and briefly introduces of method used in this paper. Section III describes the proposed algorithm on better transparent object recognition. Section IV then proposes our improvement for better region proposal for detection. Finally, in section V, we evaluate the proposed approach and investigate performance in various scenes containing transparent objects.

2. RELATED WORK

Transparent objects are hard to detect, so the research are not popular until 2003. Osadchy et al. [7]

used specular highlights as a positive source of information to recognize shiny objects. But the process required a bright light source. McHenry et al. [8] proposed several features and characteristic of transparent object such as color similarity, blurring, overlay consistency, texture distortion and highlights. The method is effective to distinguish transparent objects. Although it successfully segment transparent object, the algorithms only adapted with non-cluttered scene without pose estimation. Fritz et al. [1] use an additive model of latent features to learned transparent local patch appearance. It successfully detects transparency in varying backgrounds too.

Phillips et al. [9] provide a new idea to detect semi-transparent objects by utilizing inverse perspective mapping. This method needs to capture more than one view and assumes that object is on a plane. The largest error of pose estimate was about 10.4 mm. There still have much more improvement for pose estimation.

For pose estimation, Lysenkov et al. [10] detect transparent object by using Kinect sensor, while unknown depth information (shown as black area in depth image) is considered as transparent object. It proposed an algorithm to calculate poses of transparent objects. The improvement in [11] makes their algorithm be able to deal with overlapped instances and cluttered transparent objects.

In recent years, deep neural network has been very popular recently and has been shown powerful in object recognition tasks. So we want to try using this model to find how to recognize transparent objects.

[1] has already shown that there exists some common visual words which belong to transparent objects, thus suggested that we might be able to use caffe [2], a popular tool for deep neural network for recognizing transparent objects. However, to detect the location of transparent objects, merely using object classification module is not enough. To identify the locations of transparent objects in the image, we take

advantage of the technique called Regions with Convolutional Neural Network (R-CNN) [3], Fig.1 shows the results illustrating what R-CNN can do.

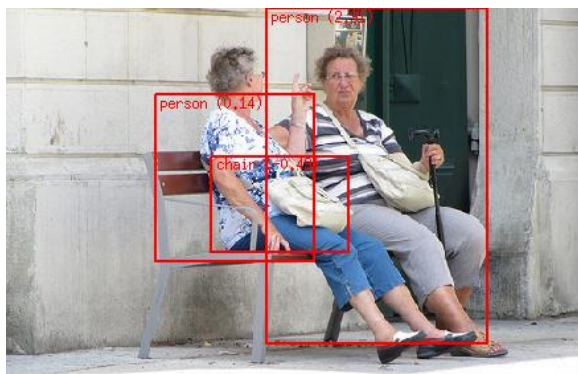


Fig. 1: An example of results of R-CNN.

3. IMPLEMENTATION

As a brief introduction, R-CNN is a state-of-the-art detector that classifies region proposals by a fine tuned Caffe model. The main idea of R-CNN is to do structured learning with lots of training examples so that in testing stage, we can assume that image is x , and the region is y , and score each hypothesis y to select the highest one as the output. With the training images, the convolutional neural network can learn what a correct region shown in Fig. 2. In this paper, training data used is the same as in [3]. The CNN is pre-trained by the dataset of ILSVRC 2012. And since beaker is a class of ILSVRC dataset, we use it to detect transparent object.



Fig. 2: An training example of R-CNN.

The first problem in the testing stage is to select the candidate regions for scoring, the techniques used in R-CNN is selective search [4]. [4] addresses the problem of generating possible object locations for use in object recognition. They introduce selective search which combines the strength of both an exhaustive search and segmentation. Like segmentation, they use the image structure to guide the sampling process. Like exhaustive search, the paper aim to capture all possible object locations. By using selective search, we don't need to exhaustively enumerate all possible regions for scoring.

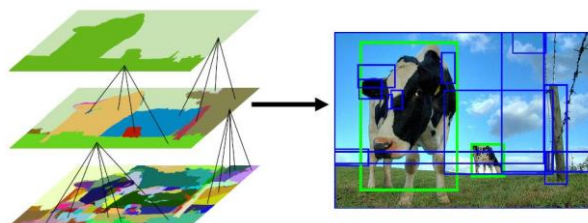


Fig. 3: An illustration of selective search.

Recognizing transparent objects is quite specific. However, training a convolutional neural network needs a lot of training images. For example, in [5] they use 1.2 million training images. For this problem, we can reference the technique used in [3]. They use R-CNN to recognize the images in PASCAL VOC, but they use ImageNet to pre-train the CNN and fine-tune the network by data of PASCAL VOC. And they found that the performance is better than training the network only with data in ImageNet. As a result, we would like to use the same training method and assess the result.

As for the neural network part, we would like to use the structure in [5], as shown in Fig 4. By adding SVMs in the final layer, we can get the R-CNN structure.

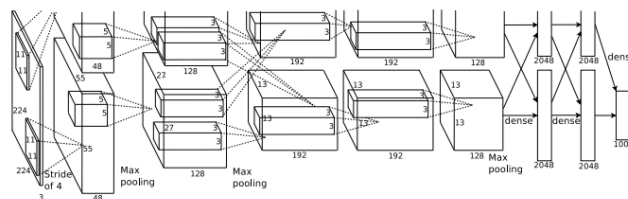


Fig. 4: The structure of used neural network.

4. IMPROVEMENT FOR SELECTIVE SEARCH

Selective search generates many candidate regions, but there are lots of candidates that are not transparent. These candidates might lower the speed dramatically. As a result, we want to use visual cues of transparent object to rule out regions that does not contain highlight.

Here we use highlight and the color of transparent object. First, we use highlight because every transparent object contains highlight in a scene with light source. Second, one of the important features of transparent object is that the color tends to be similar on both side of the edge if the background of transparent object is not cluttered. Because the object is transparent, it presents the color of background, which is similar to the color around object.

As for the highlight, in general, transparent object reflects light such that it produces multiple highlights on its surface. Various empirical models can be used to count the local highlight points on a surface, such as Phong model [6], which is commonly used in 3D computer graphics. To find the highlight, a way is to build hypothesized 3D shape and search through a large set of candidate image highlights [7]. By binary threshold, one can detect highlight regions of transparent

object, but threshold value is hard to be determined. We then used the method described in [8] as it is an efficient but useful method for searching highlight.

The first step is to threshold image by value from 0 to 255, and estimate number of perimeter pixels for each image after threshold. The slope of the curve would significantly decrease at the value close to 255. To find the critical value of threshold, we produce first order polynomial to fit the straight line of perimeter curve as:

$$P = aT + b \quad (1)$$

Variable T represents the threshold value. We fit the perimeter from threshold 255 to 0, iteratively. For each fit, we estimate the mean square error and plot another curve of error. Finally, we compare the slope of error curve from threshold 255 to 0. When there is a significant increase of slope, the value is the proper value of threshold. As shown in the Fig. 5, only highlights remain in image after threshold.



Fig. 5: Highlight extraction of image.

Next, to check if the colors on both sides are similar, we calculate the histogram of hue in HSV color space on both sides. After retrieving transparent candidate as shown in Fig 6(b), we can calculate the pixels around the candidate, the result is shown in Fig 6(c). Then we can calculate the hue histogram of pixels inside transparent candidate and pixels around the candidate. Fig 6(d) shows the histogram of the pixels inside candidate and pixels around candidate. To compare the similarity of the two histograms, we view these two histograms as vectors. By comparing the Euclidean distance, we are able to compare the similarity, if the distance is short, the colors on both sides are similar.

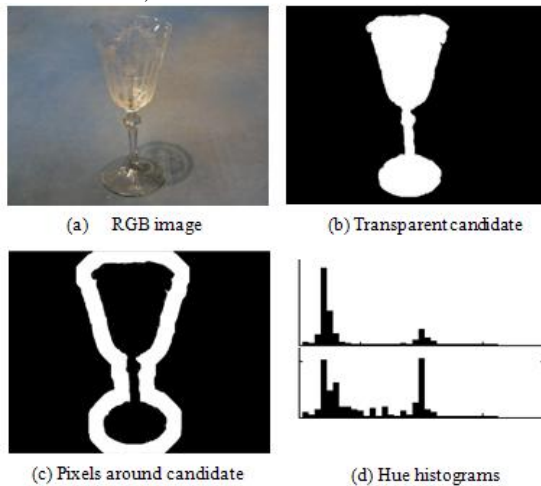


Fig. 6: The result of computing color similarity.

5. EXPERIMENT

5.1. Selective search

To test if selective search can generate regions containing transparent object, we use the test data used in [1] as input. There are 14 images in total. Three of them are shown in Fig. 7.

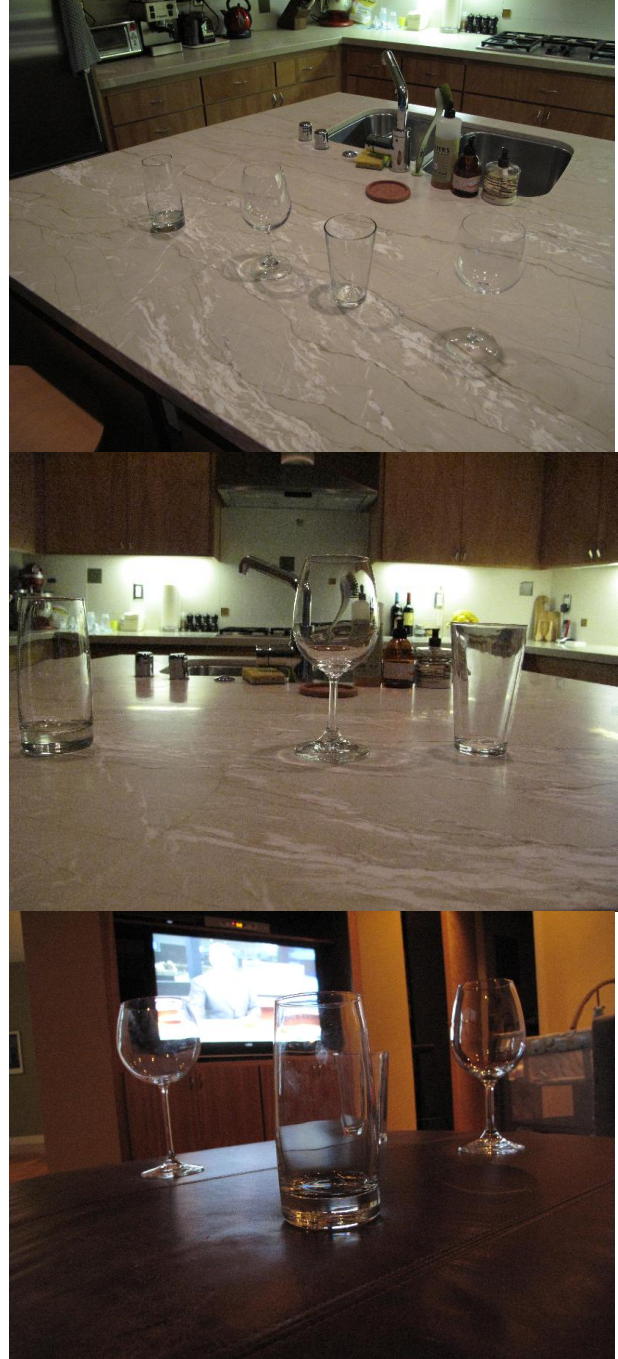


Fig. 7: The images used for testing.

As shown in Fig. 8, some proposals (in red rectangle) are good, but there are still many regions that are not transparent. As a result, we develop some algorithms

described in section 4 to prune some regions that are not transparent.

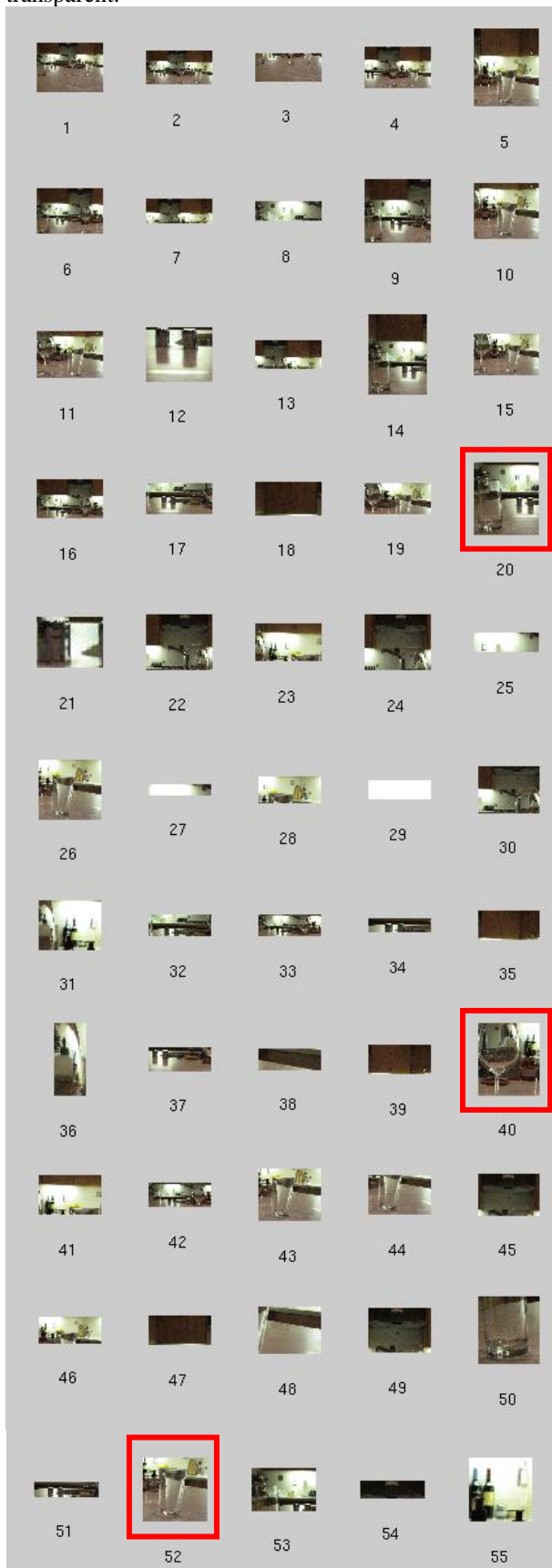




Fig. 8: The result of selective search on one image.

5.2. Transparent Object Detection

To test if our algorithm can detect transparent objects in color image, we use the 14 images as input and one of the results is shown in Fig. 10. As can be seen in Fig. 9, the red rectangle contains a transparent object and the label is beaker. Although there is a blue rectangle recognized, the label is axe, so it is not related to transparent object. The result shows that R-CNN can be used to recognize transparent objects in color image.



Fig. 9: The testing result.

Other results are shown in Fig. 10. In Fig. 10, we only show rectangles contains beaker. As can be seen, in most of the cases, transparent object can be detected in red rectangle.



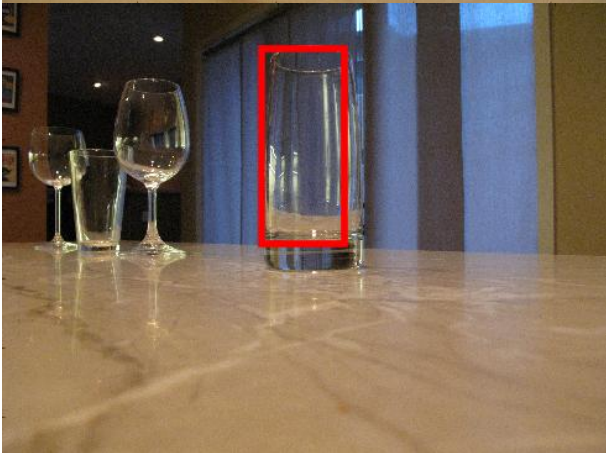
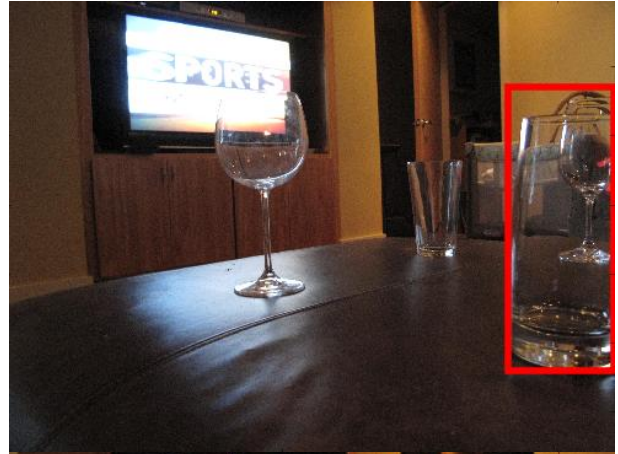


Fig. 10: The testing results.

7. CONCLUSION

In this paper, we use the R-CNN to detect transparent objects in color image. To improve the efficiency of region proposal algorithm, we use some characteristics of transparent objects to rule out some region proposals. Moreover, the result shows that this algorithm can be used for transparent object detection.

REFERENCES

- [1] M. Fritz, G. Bradski, S. Karayev, T. Darrell, and M. J. Black, "An additive latent feature model for transparent

- object recognition," in *Advances in Neural Information Processing Systems*, 2009, pp. 558-566.
- [2] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, et al., "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*, 2014, pp. 675-678.
 - [3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2014 IEEE Conference on, 2014, pp. 580-587.
 - [4] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, pp. 154-171, 2013.
 - [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.
 - [6] B. Tuong-Phong, "Illumination for computer-generated images," *University of Utah*, pp. 29-51, 1973.
 - [7] M. Osadchy, D. Jacobs, and R. Ramamoorthi, "Using specularities for recognition," in *Computer Vision*, 2003. *Proceedings. Ninth IEEE International Conference on*, 2003, pp. 1512-1519.
 - [8] K. McHenry, J. Ponce, and D. Forsyth, "Finding glass," in *Computer Vision and Pattern Recognition*, 2005. *CVPR 2005. IEEE Computer Society Conference on*, 2005, pp. 973-979.
 - [9] C. J. Phillips, K. G. Derpanis, and K. Daniilidis, "A novel stereoscopic cue for figure-ground segregation of semi-transparent objects," in *Computer Vision Workshops (ICCV Workshops)*, 2011 *IEEE International Conference on*, 2011, pp. 1100-1107.
 - [10] I. Lysenkov, V. Eruhimov, and G. Bradski, "Recognition and pose estimation of rigid transparent objects with a kinect sensor," *Robotics*, p. 273, 2013.
 - [11] I. Lysenkov and V. Rabaud, "Pose estimation of rigid transparent objects in transparent clutter," in *Robotics and Automation (ICRA)*, 2013 *IEEE International Conference on*, 2013, pp. 162-169.