2017 17th International Conference on Control, Automation and Systems (ICCAS 2017)
Oct. 18-21, 2017, Ramada Plaza, Jeju, Korea

Data Augmentation using Synthesized Images for Object Detection

HyunJun Jo [1], Yong-Ho Na [2] and Jae-Bok Song[3*]

[1] School of Mechanical Engineering, Korea University,
Seoul, 02841, Korea (jhj0630@korea.ac.kr)
[2] Department of Mechatronics, Korea University,
Seoul, 02841, Korea (rna4352@korea.ac.kr)
[3] School of Mechanical Engineering, Korea University,
Seoul, 02841, Korea (jbsong@korea.ac.kr) * Corresponding author

Abstract: Recently deep learning-based research has been conducted in various fields. Deep learning algorithms require vast amounts of data for good performance. Therefore, collecting such a huge amount of high-quality data is crucial to the deep learning-based methods. Data collection is simple but very time-consuming. To cope with this difficulty, in this study we propose a method to generate a dataset by synthesizing the images of background and object. Various images can be generated through post-processes such as adding noise and changing brightness to the images of objects obtained from different viewpoints. Furthermore, we do not need to manually annotate the dataset for object detection because we can calculate the parameters of the bounding boxes from the location and size of object images during the synthesis process. Faster R-CNN, one of the deep learning algorithms for object recognition, was used to verify the proposed method. The performance based on the dataset generated by the proposed method is comparable to that based on the real dataset.

Keywords: data augmentation, synthesized images, deep learning, object detection

1. INTRODUCTION

Recent deep learning-based object recognition shows excellent performance [1, 2] under the condition that a large amount of data is provided for training. Therefore, a public dataset, such as ImageNet, is often used to satisfy this condition. Data augmentation is sometimes used to increase the size of dataset [3]. However, in situations where it is difficult to use public dataset, data must be collected directly. Data collection itself is not difficult, but it is time-consuming because of the need to gather a lot of images and manually annotate the positions of the objects for each image. To deal with this problem, simulators or computer graphics have been used to collect images [4]. However, despite the use of computer graphics annotating the bounding boxes often must be manually handled.

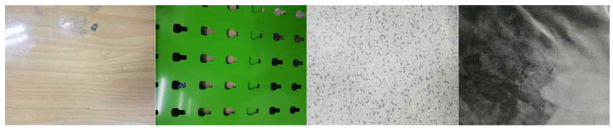
In this research, we propose a method to synthesize images to construct the dataset for efficient object recognition. First, background images and object images are acquired separately. The object images are arbitrarily resized, rotated, and then placed at an arbitrary position in the background. The images in various environments can be collected through post-processing such as adding arbitrary noise to the generated images. Moreover, since it is possible to know the position where the object images are placed in the background image and the size of the object images during this process, annotations can be automatically conducted from the information on the bounding boxes.

The rest of this paper is as follows. Section 2 discusses the method for synthesizing images. The use of synthesized images is verified through a Faster R-CNN algorithm in Section 3. Finally, conclusions are drawn in Section 4.

2. IMAGE SYNTHESIS AND AUTOMATIC ANNOTATION

2.1 Image synthesis

In this study, we provide a method to synthesize images and generate a dataset more efficiently than traditional methods. First, background images and object images are separately captured. In this research, 40 background images were prepared as shown in Fig. 1(a). In addition, as shown in Fig. 1(b), each object provides six images taken from different angles. Since six different objects are used as shown in Fig. 2, the total of 36 object images are prepared. Two of 36 object images were chosen arbitrarily and they were modified with arbitrary sizes and angles. They were then placed at arbitrary positions on the randomly chosen background images.



(a) examples of backgrounds

(b) examples of object images

Fig. 1 Examples of (a) backgrounds and (b) object images.

978-89-93215-13-7/17/$31.00 ©ICROS

Fig. 2 Six target objects.

Three types of post-processing were conducted on the images obtained through the above process: brightness change, noise addition, and image blurring. A salt & pepper noise and Poisson noise were added to the original image and a 3x3 mask was used for image blurring. Through this process, we were able to obtain the diversity similar to that obtained in the real environment. The effect of each post-processing is shown in Fig. 3.

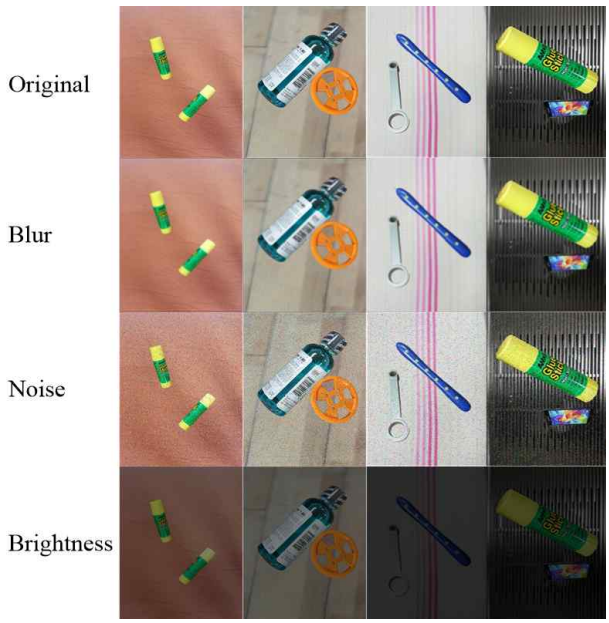


Fig. 3 Effects of each post-process.

2.2 Automatic annotation

To utilize the generated images for deep learning, two pieces of information are necessary. One is the type of the object on the image, and the other is the set of parameters of the bounding box for each object which displays the position of the object. The information on the bounding box and object type can be obtained automatically during image synthesis. First, the object type can be known through the management of the image file. The image files of the object are randomly selected, but the object type can be identified since the selected file is known. The bounding box can be can be described in two ways. One way is to use (x_{min}, y_{min}) and (x_{max}, y_{max}) which correspond to the x and y coordinates of the left-top point A and the right-bottom point B of the bounding box in Fig. 4. The other way is to use (x_c, y_c), w_{obj}, and h_{obj} which correspond to the x and y coordinates of the center, width and height of the bounding box, respectively. Both ways can be used, but the former is adopted in this study. When annotating a real image manually, a bounding box is drawn on the desired object. The annotation file is then created by storing the four variables and the object type together.

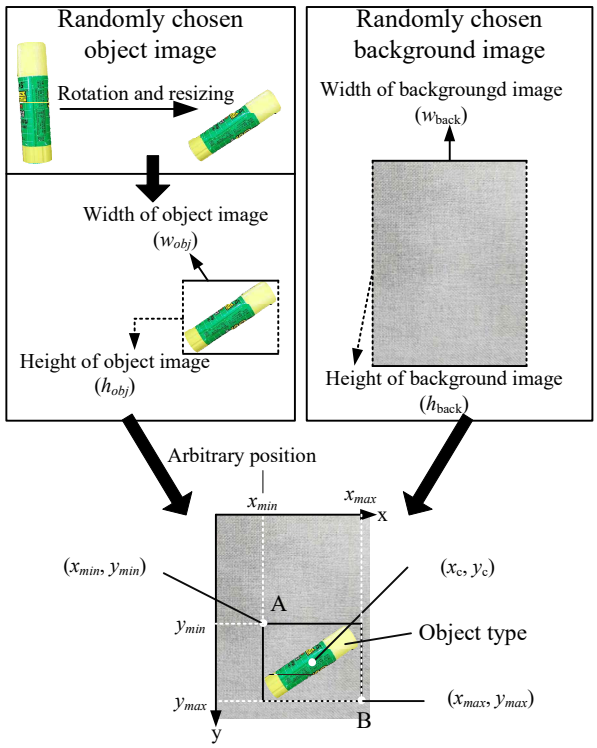


Fig. 4 Progress of image synthesis and automatic annotation.

Next, the information of the bounding box can be known via the size of the object image and the position where it is placed. The 4 parameters x_{min}, y_{min}, x_{max}, and y_{max} can be obtained by

$$x_{\min} = (w_{back} - w_{obj}) \times r_1 \quad (1)$$

$$y_{\min} = (h_{back} - h_{obj}) \times r_2 \quad (2)$$

$$x_{\max} = x_{\min} + w_{obj} \quad (3)$$

$$y_{\max} = y_{\min} + h_{obj} \quad (4)$$

where w_{back}, w_{obj}, h_{back}, and h_{obj} are shown in Fig. 4, and r_1 and r_2 are the arbitrary numbers between 0 and 1. Therefore, the object types and parameters related to the bounding boxes are obtained during the image synthesis, thus leading to automatic annotation.

3. EXPERIMENTS

As mentioned above, 40 background images and 36 object images were used in this study. These images created 25,000 images that were employed as a training dataset. To verify the usefulness of the synthesized

dataset, another training dataset was composed of 13,000 images taken in the real environment. Examples of these two datasets are shown in Fig. 5. In addition, a test dataset was composed of 1,300 real images which were not included in the training dataset.

(a) Examples of synthesized images

(b) Examples of real images

Fig. 5 Examples of (a) synthesized images and (b) real images.

Faster R-CNN (regions with convolutional neural network), one of object detection algorithms, is employed for the experiment in this study. This algorithm can recognize the type and position of objects on the image. It has a structure in which a region proposal network (RPN) and a classifier are connected in parallel at the end of a CNN structure such as ZF-Net [5] or VGG-Net [6]. Fig. 6 shows the simplified structure of Faster R-CNN. RPN detects the position of objects, and the classifier uses the results of RPN and CNN together to determine the type and position of objects. Faster R-CNN is trained by the synthesized dataset and the real dataset, respectively. The result of training Faster R-CNN using these two training datasets is shown in Fig. 7. Fig. 8(a) and 8(b) are the results of applying Faster R-CNN trained by the synthesized dataset and the real dataset to an image in the test dataset, respectively.

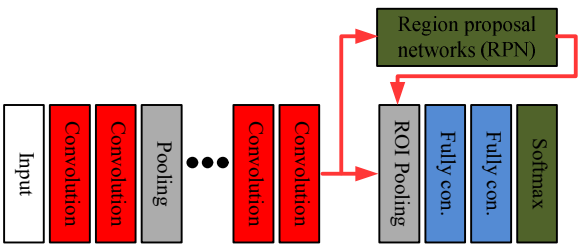


Fig. 6 Simple diagram of Faster R-CNN structure.

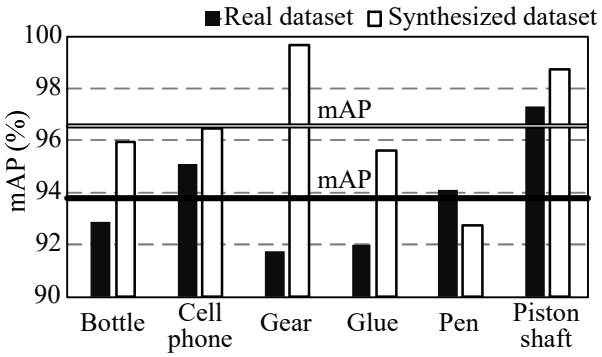
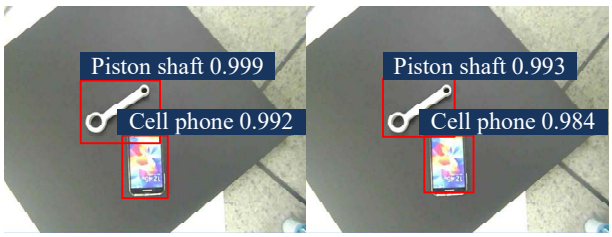


Fig. 7 Performance comparison of faster R-CNN trained by synthesized dataset and real dataset.



(a) Result with synthesized dataset

(b) Result with real dataset

Fig. 8 Example of the same image using Faster R-CNN trained by (a) synthesized dataset, and (b) real dataset.

According to the experimental results, the resulting mAPs of the two cases trained by the synthesized and the real dataset are 96.5% and 93.9%, respectively. The result based on the synthesized dataset showed a 2.6% higher mAP than that based on the real dataset. However, as shown in Fig. 8, there is no big difference when the results of synthetic and real datasets are applied to a real image. Both models trained by the synthesized dataset and the real dataset successfully found a piston shaft and a cell phone. The location and size of their bounding boxes are also similar. The class probability of the piston shaft predicted by both models were 0.999 and 0.993, respectively. In the case of a cell phone, the predicted class probabilities of both models were 0.992 and 0.984, respectively. The closer to 1 the class probability is, the stronger the confidence of the class is. In summary, all results, such as mAP, size and position of the bounding boxes, and class probability were similar in both cases. This is because the amount of synthetic dataset is twice as much as that of the real dataset although the synthetic data does not reflect the real environment perfectly (e.g., perspective).

The synthesis method also reflects the environments that were not taken into consideration when collecting real images. For example, if a dataset is collected manually, the dataset may have a lot of data at a certain brightness, which results in the biased dataset. However, if images are synthesized under arbitrary brightness, the dataset is less likely to be biased. In summary, the synthesized dataset has advantages in quantity and variety, but it does not reflect real environments well.

On the other hand, it takes a long time to construct a dataset consisting of only real images, but the synthesis method could generate the same amount of data in a very short time. Consider the time required to create or capture the image. To include diversity in real images, the number of images that can be taken in a similar environment is limited to 20. Under this condition, we could obtain about 600 images an hour. In other words, it takes 6 seconds to gather one real image. On the contrary, it takes only 0.08 seconds to create one image in image synthesis. Next, if one manually annotates the images that contain two objects, he or she can process approximately 250 images an hour. In other words, it takes 14.4 seconds per image on average. However, using the proposed method, the annotation takes about 0.01 seconds per image under the same conditions. Consequently, it takes 20.4 seconds per image to manually annotate a real image, but it takes only 0.09 seconds per image to automatically annotate a synthesized image. This means that the synthesized dataset can be created 226.7 times faster than the real dataset, if the dataset is composed of the same number of images. Table 1 summarizes the experimental results.

Table 1 Time spent to construct datasets.

	Time to annotate an image	Time to create or take an image	Time consumed per image
Synthesized dataset	0.01 s	0.08 s	0.09 s
Real dataset	14.4 s	6 s	20.4 s
Ratio (real/syn.)	1440x	75x	226.7x

4. CONCLUSION

In this study, we propose a method to efficiently generate a large amount of synthesized data based on a small number of object images and background images. The training result using the synthesized data showed a 2.6% higher mAP than that using the real data. However, the overall performance is similar because there is not much difference between mAPs, position and size of a bounding box, and class probability. This is because synthesized datasets have strengths in terms of quantity and environment compared to real datasets, but they cannot reflect the real environment fully. On the other hand, synthesized datasets can be produced 226.7 times faster than real datasets. Conversely, a synthesized dataset can get 226.7 times more data than the real dataset for the same time. This can mitigate the lack of data in object detection algorithms based on deep learning algorithms.

ACKNOWLEDGMENT

This research was supported by the MOTIE under the Industrial Foundation Technology Development Program supervised by the KEIT (No. 10067441)

REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, pp. 91-99, 2015.

[2] Y. Li, K. He, and J. Sun. "R-fcn: Object detection via region-based fully convolutional networks," *Advances in Neural Information Processing Systems*, pp. 379-387, 2016.

[3] A. Krizhevsky, I. Sutskever, and H. Geroffrey, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems,* pp. 1097-1105, 2012.

[4] B. Sapp, A. Saxena, and A. Y. Ng, "A Fast Data Collection and Augmentation Procedure for Object Recognition," *Association for the Advancement of Artificial Intelligence,* pp. 1402-1408, 2008.

[5] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolution Networks," *European conference on computer vision,* pp. 818-833, 2014.

[6] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," International Conference on Learning Representations, 2015.