2018 IEEE International Conference on Big Data and Smart Computing

Image Preprocessing for Efficient Training of YOLO Deep Learning Networks

Hyeok-June Jeong, Kyeong-Sik Park, Young-Guk Ha
Computer Science & Engineering, Konkuk University
120 Neungdong-ro, Gwangjin-gu, Seoul 05029. KOREA

Abstract- **Every artificial intelligence system needs big data for training. In particular, artificial intelligence for object detection requires a lot of images for training. It is possible to obtain training images from internet by web crawler. In most cases, however, images collected by the crawler are often not refined. In other words, it can't be used as training data in any artificial intelligence platform without proper pre-processing. This paper describes a pre-processing system for images collected by a crawler for YOLO training. And it will proved that this system is capable of efficient training.**

I. Introduction

Recently, Artificial Intelligence (AI) has been attracting attention in many fields. Especially, performance of object detection based on AI is superior (fast, accurate and flexible) to any existing technologies such as template matching, SURF.

However, it is difficult to use AI-based technologies because, there are many difficulties in understanding techniques or implementing system. The biggest challenge using AI, however, is that it needs training big data. In general, AI technologies have a significant association with the objects to be recognized and the training data. That is, in order to recognize or detect an object from a video stream, image data should be used as training data. Furthermore, it requires very well-refined image big data, for good training.

Image Big data can be obtained in many ways, but it is most economical to collect it on the Internet as a crawler. It is easy to get a lot of data, but the problem is images on the Internet are not normalized so it can't be used for training.

This paper proposes a system for preprocessing images collected by web crawler for YOLO training

II. Related work

Now the computer can emulate human intelligent behavior, because artificial intelligence mimics the neural network structure of a living organism. Generally artificial neural network is implemented Deep Neural Network (DNN). Likes human, it can perform intellectual activities such as judgment and recognition through training [1].

Convolution neural network (CNN) is the most powerful and effective DNN to recognize objects from images. CNN have a reputation in Images Net Challenge which distinguish 1000 categories from one million images [2]. But CNN can recognize an object in a single image. In other words, CNN can't detect different objects in a single image.

Object detection is a need to not only recognize whether specific object is present or not, but also determine the exact position of the target region and separated by bounding box of multiple objects in the image. Reference [3] refers Regions with CNN features which uses a selective search method to detect objects in the image in 2000 candidate regions. At the time this study was published, it was very innovative in object detection, but there are performance problems due to many operations.

In recent years, there have been a lot of studies to make object detection fast and accurate. Single Shot MultiBox Detector [4] attempts to detect objects by applying feature maps based on CNN. The feature points are formed in a grid on the image and the anchor box which centered on each feature grid cell determines the presence of an objects. SSD is faster than Fast R-CNN, but its accuracy is lower because limited number of the anchors of a grid cells act as constraints.

A new algorithm called You Only Look Once [5] is similar to SSD. The difference from SSD is that it removes FC and changes to fully-convolutional model and the application of hard negative, that is, the object does not learn a position that can never exist. This attempt could improve accuracy and speed then SSD. In other words, YOLO is the best object detector.

However, in order to get good performance with YOLO, it should be trained with images that are appropriate for its characteristics. This paper describes a system that pre-processing train images collected by web crawler according to YOLO characteristics. This system will make it easier to collect YOLO training data and maximize its performance.

III. System design

A. Characteristics of YOLO

YOLO is the fastest object detector with the accuracy of R-CNN. However, in order to obtain the object detection accuracy, training images must reflect the characteristics of YOLO.

YOLO performs CNN-based object recognition in a box called an anchor which is centered on the 13x13 grid cell in the image. That is, the size is reduced to 416x416 regardless of the size of the original image to be trained or recognized. That means if there is a large difference in the ratio of width to height of training images, serious distortion occurs on objects in the resizing process.

2375-9356/18/$31.00 ©2018 IEEE
DOI 10.1109/BigComp.2018.00113

IEEE computer society

If there are difference of the ratio between training image and recognition image YOLO shows poor performance. In this case, distortion of the shape of the object also occurs. That is, a good performance can be expected only if the learning image and the recognition image have the same ratio.

It is necessary to understand the characteristics of the anchor for good accuracy. Anchors are defined as a fixed ratio and it used in the training and detection process. Anchors must be defined heuristically, it is usually calculated in K-means by looking at the shape of objects in the training images. In other words, it is important to make similar ratio of area occupied by each objects throughout the image in training and detecting.

In summary, for good accuracy of YOLO, training images should be pre-process as follow.

1. All the size of the training image and the detection image should be same.

2. The proportion of the area occupied by the object in the image must be similar in the training image and the detection image.

First, the region of the objects should be correctly annotated in each crawled images then regenerate training images based on the above principle. This system call Object Crop and Relocation manager and following sections describe these system in detail.

B. Object Crop and Relocation manger

This system is an image pre-processor for YOLO training and all training images are is collected by web crawler. That is, all the objects are extracted from crawled images and relocated in new images which is similar to the environment should be detected. The system architecture is shown in Figure 1 below.

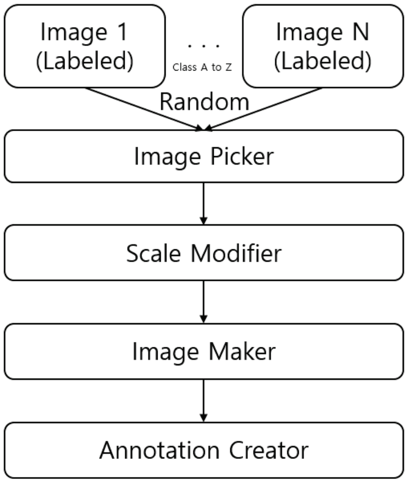

Fig. 1. Object Crop and Relocation manger architecture

Every crawled images have one or more specific objects and all objects in images are annotated with the size, the location and its class for example, cone, car, truck, pedestrian and so on. Annotation is a really important task but the annotation process is not the scope of this system. That means, this system can be operate after it is done.

1. *Image picker*: This subsystem randomly picks out objects from a prepared crawled image set. First it randomly selects the class of the object, and picks up an image within the class then crops the annotated region.

2. *Scale Modifier*: This subsystem reduces the size of objects cropped in the Image picker to an appropriate size. Usually crawled images are often large in size because they are mostly single object images. Appropriate size range is determined by considering the size of the target appeared to recognize.

3. *Image Maker*: In this step, the modified object image is pasted into base images. Base images with similar backgrounds and sizes. The base image is composed of the same size and similar background to the image to be detected. In one base image, one or more objects are placed at random locations.

4. *Annotation Creator*: This subsystem annotates the position and size of newly placed objects in the base image. These annotations can be calculated based on the tasks applied in steps 1 to 3.

The following figure 2 shows the process of creating a learning image in the system.

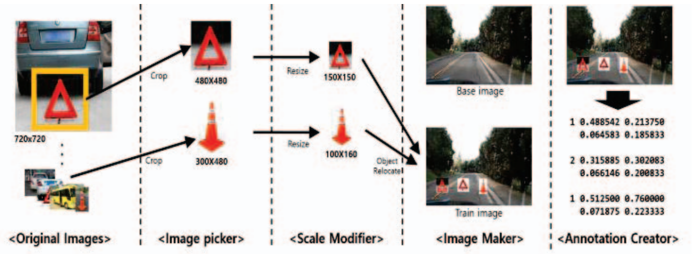

Fig. 2. Image preprocessing process

The Image picker crops object in the annotated area from the original image set. Then reduce the size of the object in the Scale Modifier. The reason for reducing size is to match the size of the object to be detected and trained.

The Image Maker picks a base image and relocates the objects at random locations. The important thing is that each object should not be overlapped.

Finally, the Annotation Creator recalculates the location, size, and class of relocated objects to create annotations.

By repeating this process, the crawled images can be reprocessed in a form suitable for YOLO training, and data of various cases can be easily obtained. In addition, since images can be generated, it is possible to acquire more images for training.

IV. Implementation and Result

The proposed system works on a server with Xeon E5, DDR4-128GB, GTX1080 4way for YOLO training. The operating system is Ubuntu 14.04, a study was done in CUDA, cuDNN, YOLO v2 platform.

In this study about 190,000 object was generated from with 6 classes and 25,000 images were used as training images.

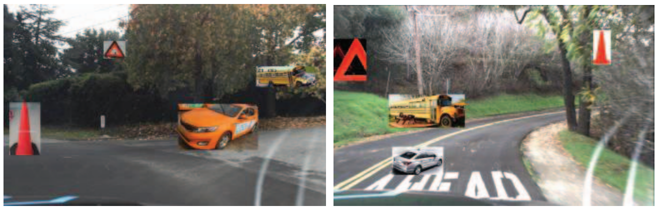

Fig. 3. Example of generated training image

Figure 3 shows the image generated by the system. As previously suggested, the object relocated into the base image from the crawled image. These images were repeatedly generated and YOLO was learned as follow.

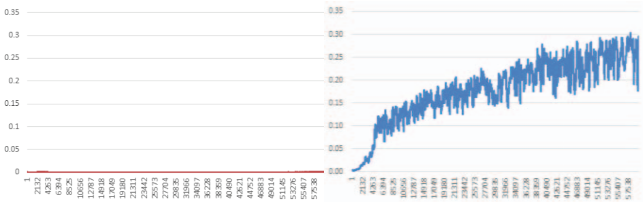

Fig. 4. Obj graph of YOLO training log. Only use original crawled image (left), Image created by the proposed system(right) Up to 60000 iteration, the higher the better

Figure 4 shows the Shows Obj changes according to YOLO training. The value of Obj indicates the probability of detecting an object in the image. This means that the proposed method is effective.

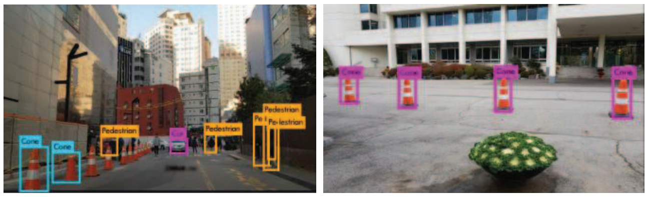

Fig. 5. Object detection results

The above figure shows the result of object detection by the trained network in the proposed method. This result shows that performance of object detection and objects categorized in images with cones, people, and cars. If the YOLO training is performed from the original crawled image, it can’t detect any thing. In other words, proposed system makes acquire the training images very easily and improves detection performance.

V. Conclusion

This paper described a system that pre-processing train images collected by web crawler Optimized for YOLO. Before designing the system, two key factors have been mentioned. First, all the size of the training image and the detection image should be same, second, the proportion of the area occupied by the object in the image must be similar in the training image and the detection image.

This system is designed 4-Steps to generate the appropriate training image, Image picker, Scale Modifier, Image Maker and Annotation Creator. This system makes the crawled images to suitable for YOLO training.

The result was successful. The YOLO training log value was also good. The object detection results were also very satisfactory. In other words, the research was successful.

However, just six classes used in this study. The next study will be performed with more classes. It is need to develop the research in a more mathematical method than random based resizing or placing.

Acknowledgment

This work was supported by the ICT R&D program of MSIP/IITP (2016-0-00004, Development of Driving Computing System Supporting Real-time Sensor Fusion Processing for Self-Driving Car).

References

[1] Demuth, Howard B., et al. Neural network design. Martin Hagan, 2014.

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012..

[3] Demuth, Howard B., et al. Neural network design. Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." arXiv preprint arXiv:1606.00915 (2016).Martin Hagan, 2014.

[4] Liu, Wei, et al. "Ssd: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016.

[5] Re dmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.