Surprisingly Easy Network Compression and Data Extension for Object Instance Detection

Rui Wang*, Jingwen Xu*, Tony X Han†
*Key Laboratory of Precision Opto-mechatronics Technology, Ministry of Education
School of Instrumentation Science and Opto-electronics Engineering,Beihang University,Beijing,China
†Jingchi.ai ,Beijing, China
Email: wangrbuaa.edu.cn, xujingwen9402@buaa.edu.cn,tony.han@jingchi.ai

***Abstract*—To detect instances in unstructured environment with mobile system, we develop a light weight but accurate learning model denoted as B-PA(BING Pruned Alexnet). Our method first utilizes BING(Binarized Normed Gradient) to compute bounding boxes, then builds a compressed network for recognition by pruning neurons and cutting fully connected layers on the original noted Alexnet. Addressing the problem that the training samples for instance detection are limited and of small variation, we extend the training data by combining data augmentation with synthetic generation. Our B-PA model takes only 5.3MB, which is 50 times smaller but with equivalent or even higher accuracy than the original Alexnet. Experiment results demonstrate that our method outperforms the state-of-art instance detection algorithms on WRGB-D Dataset and GMU Kitchen Dataset.**

***Index Terms*—Object Instance Detection , Pruned Alexnet , Binarised Normed Gradient , Data Extension; Synthetic generation**

I. Introduction

Object instance detection refers to recognizing and locating some specific objects in an image or video. It is a core functionality in many applications of computer vision, especially in the humanoid robotics. Imagine using an object detection system for an everyday indoor environment like your family or office. We do need such system to not only recognize different kinds of objects, e.g. can versus box, but also have a keen eye on specific instances, e.g. soda can versus coffee can. Thus, how to detect the instance in unstructured environments with complicated issues such as noise, occlusion, random variation in illumination, scales and viewpoints is a big challenge. With the amazing progress that has been made in visual recognition by various deep networks, which can extract robust features thus to adapt to the complex detecting environment, one may expect to easily take an existing neural network model and deploy it for such instance detection setting.

However, those state-of-the-art neural networks typically have up to millions of parameters, they are generally both computationally and memory intensive, making them difficult to deploy on embedded systems with limited hardware resources and power budgets. Furthermore, large amount of annotated images with rich variations are needed to train a deep network for detecting in unstructured environment. Traditionally, such a mammoth work must be taken via realistic shooting images in random background. Nevertheless collecting and annotating scenes in such way is expensive and time-consuming.

This work was supported by a grant from the National Natural Science Foundation of China (61673039) .

To address these problems, we refer to the study of two directions. One is network compression, which does the research to reduce computational cost and file volume of the network model without sacrificing accuracy. For example, SqueezeNet [1] matches AlexNet [2]-level accuracy on ImageNet with $50\times$ fewer parameters. GoogLeNet-v1 [3] has only 53MB of parameters, and it matches VGG [4]-level(533MB) accuracy on ImageNet; Another one is data extension, which relates to automatically generating new annotated training samples by means of augmentation or synthesis. Data augmentation such as color jittering and random scaling are frequently-used schemes. Data synthesis involves either synthetically rendering scenes and objects with CAD or superimposing object masks into scene images [5].

Inspired by aforementioned achievements in network compression and data augmentation, an efficient network architecture called B-PA (BING [6] +Pruned Alexnet [2]) together with our training data extension strategy is put forward in this paper. This novel method contributes to the solution for detecting specific instance object by the following means:

1. An compressed network architecture B-PA, which reaches AlexNet-level accuracy on object instance detection task with $50\times$ fewer parameters is introduced. This compression is carried out by preserving 75% neurons on each convolution layer and removing the first two fully connected layers in original Alexnet. Additionally, with the efficient region proposal technique BING [6], we are able to narrow down the target search space, thus to reduce computation load further.

2. An effective training data extension strategy, incorporating data augmentation with the synthesis method of synthetically superimposing object masks into the scene images, is employed in our work, which enables researchers to obtain abundant training data with minimal effort.

II. Data Extension and Network Compression

The flowchart of our proposed method is shown as Fig. 1. In the offline learning stage, we first extended the images containing the target instance object by data synthesis and augmentation then utilized all these augmented images along with their annotation information to train the region proposal

model BING(binarised normed gradients [6]) and the deep neuron network Pruned Alexnet. Once finishing offline training, a discriminative model called B-PA is obtained. In the online detection stage, given a test image, proposals generated by BING from the test image are fed into the Pruned Alexnet and assigned to regions with different category labels. Finally, all the proposals in the same category with high confidence are combined together, thus the location of the object can eventually be obtained.

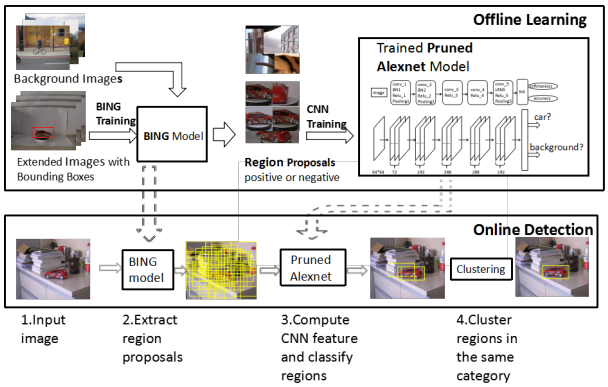


Fig. 1. Flowchart of object instance detection using B-PA

A. Data Extension Strategy

Abundant training data is quite important for improving the performance of deep networks, thus we introduced a simple way to generate rich training data with minimal effort. It includes two kinds of enhancement strategies: data synthesis and data augmentation, as shown in Fig. 2.

1) Data Synthesis: Our data synthesis strategy can be roughly summerized as extracting masks of objects, then blending them with random backgrounds. However, naively pasting object masks on scenes creates subtle artifacts, including edge discontinuity and global inconsistency. Some existing works, like [5], try hard to minimize both these artifacts. Rather, our key insight is that only eliminating edge discontinuity and ensuring patch-level realism can provide enough training information for region proposal based detector. Therefore, we focus mainly on edge smoothing. When pasting the masks of instance object on random backgrounds, we extract contours of each instance object, then traverse every contour point (x, y) for mean filtering in its 3×3 neighbourhood $N_{(x,y)}$. Then, the filtered value$\bar{f}(x, y)$ of a contour pixel is calculated as "(1)".

$$\bar{f}(x,y) = \frac{1}{3 \times 3} \sum_{(s,t) \in N_{(x,y)}} f(s,t) \tag{1}$$

In this way, only the values of contour pixels are blurred, which can mitigate boundary artifacts and maintain image clarity.

2) Data Augmentation: Apart from superimposing the objects on random background, we also adopted data augmentation methods, such as color jittering and geometric alteration, to enrich variation of training data.

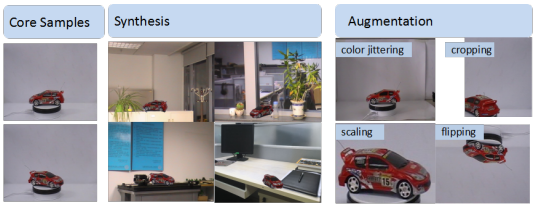


Fig. 2. Examples of our extended training data.

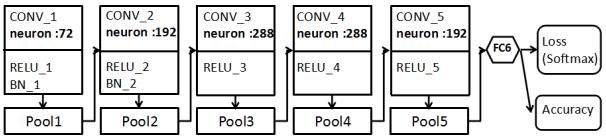


Fig. 3. Overall structure of Pruned Alexnet.

B. BING-Pruned Alexnet

When designing the detection architecture for instance objects, our chief concern is to decrease deep model size and guarantee detection accuracy simultaneously. To achieve this, we first utilized BING [6] to capture potential object locations while reducing the searching space as far as possible, then designed a compressed CNN model,Pruned Alexnet, for recognition. The compact architecture consisting of 7 trainable layers: 5 convolutional layers, 1 fully connected layer and 1 softmax layer shown in Fig. 3. Our pruning process is a two stage pipeline to remove the fully connected layers, and to cut down neuron number on each layer. The details of these two steps will be elaborated in the following part.

1) Removal of Fully connected layers: The fully connected layers (FCs) function as joint and transfixion to bridge the convolutional layers with neural network classifiers. However, the FCs have up to millions of parameters which account for 80% in the network. With this in mind, we propose to remove two FCs and preserve the final one so as to alleviate parameters redundancy and keep the bridge function at the same time.

However, when two FCs are removed, the dropout method disappeared at the same time, which may render serious overfitting problem. From [7], the batch normalization(BN) [7] can successfully resolve the overfitting problem. Therefore, we propose to use BN method to play the role of dropout. According to reference [7], the basic principle of batch normalization is illustrated as follows:

$$\overline{X}_{norm}^{(k)} = \frac{X^{(k)} - E[X^{(k)}]}{\sqrt{Var[X^{(k)}] + \varepsilon}} \tag{2}$$

where $\overline{X}_{norm}^{(k)}$ is the k^{th} normalized output of the convolution layers, $E[X^{(k)}]$ is the expectation over the batch input samples, and $Var[X^{(k)}]$ is the variance of the batch input samples, ε is a micro-constant.Since BN can additionally functions as normalizers, we replace the original local response normalization layer (LRN) [2] with BN layer.

2) Reduction on neuron number: One key property of a network architecture is its ability to produce a good representation of data. Redundant features not merely take up plenty

of computing resources, but also cause the network to be interfered with insignificant details, thus, the more feature maps are not the better. When applying deep learning network on instance detection task, it does not need to recognize thousands of objects (like in Imagenet [2]). Besides, instance objects have more explicit appearance compared with a category of objects, thus an instance detection network will need fewer feature maps to describe targets. Therefore, the pre-pruning method is adopted here to reduce the neuron number. To trade off between accuracy and model size, we pruned the neurons on each layer in original Alexnet at the ratio of 75%. Finally, we obtained a concise instance detection model which occupies only 5.3M but achieves Alexnet-level accuracy when recognizing on GMU Kitchen dataset [8].

III. Experiments

In this section, our proposed learning model B-PA is trained under Caffe framework with NVIDIA-GTX-1080 and is evaluated on two challenging datasets: Washington RGB-D dataset [9] , and GMU Kitchen Dataset [8].

A. Training Data Setup

1) WRGB-D Object Dataset.: Based on the provided RGB images of four target instances(soda can, cap, cereal box, flashlight) in WRGB-D Object Dataset [9], we adopted data augmentation methods to enrich the sizes and lightning conditions. We also utilized four scene images from WRGB-D Scenes v2 dataset [9] for background blending. Finally, about 4200 synthetic images are generated for each instance using all modes of data extension described in Sec II-A.

2) GMU Kitchen Training Set.: GMU Kitchen Dataset [8] contains thousands of annotated images taken in 9 comlicated scenes.The train-test split follows the division in [8], in which six scenes are used for training and three are used for testing. We call these training images realistic data. In order to enrich the variation of our training data, another 4200 synthetic images are generated for each instance using all modes of data extension in Sec II-A.Noticeable,when we generate synthetic data,the instances images are from BigBird Dataset [10], while the background images are selected from WRGB-V2 [9]. And we call these images extended data. In this experiment, we utilized real data, real data+extended data,these two kinds of data to train different detection network.

B. Detection Evaluation

Detection using our trained B-PA was implemented on the test split of WRGB-D Dataset (Scenes v1) ,and GMU Kitchen Dataset, which contains thousands of images taken in common indoor environments, like laboratory areas, kitchen rooms, and office workspace. The overall detection results are illustrated in Fig. 4. It can be seen that our detection model can achieve satisfying results in cluttered scenes, including size, illumination, viewpoints changes as well as occlusion.

C. Comparison with state-of-art Methods

1) Comparison with Traditional Methods: To further elaborate the superiority of our method, we compare our algorithm with two traditional detection methods: HOG+ SVM [9] and B-CST [11]. Here, we report precision-recall curves and average precision for evaluation, in which average precision is an approximation of the area under the precision-recall curve. It shows in Fig. 5 that the area under the red curve(representing our B-PA method) is much larger than the two other curves, which means that our approach can reduce false positives significantly, meanwhile, keeping a relative high detection rate compared with the other two methods.

2) Comparison with Deep Detection Models: For comparison with other deep detection models, we provide the performance of our proposed B-PA, Faster RCNN [12], and SSD [13] model trained solely on the real data in GMU. Also, we did extra experiments using both real and extended data (our extended data or synthetic data from [5])to train the deep models. Table.I shows the evaluation results. As far as the detection accuracy is concerned, our B-PA algorithm is superior to SSD [13], but it is inferior to Faster RCNN [12], when solely trained with real data. When we add our extended data in the training set, our B-PA model can achieve competitive result,80.55% mAP. However,it is still inferior to Faster RCNN trained with SP-BL-SS. This is because our B-PA uses the framework of Alexnet for classification,while Faster RCNN adopt the VGG16 network,which has stronger recognition ability. Still, these results can show that our proposed training data extension strategy provides complementary information to provide performance boost for detectors.

Noticeable,our model takes only 5.3 MB, while the other two deep learning models(Faster RCNN and SSD) both occupies over 500 MB of memory. Therefore, to trade off precision and size, our B-PA is a cost effective choice, especially when trying to apply deep detection network on mobile device.

IV. Conclusion

In this paper, we focus on designing a concise instance detection model that has very few parameters but high accuracy. Our whole framework provides a combination of the region proposal technique BING and a pruned Alexnet. Moreover, to address the problem of over-fitting, a novel data extension strategy is developed. We showed that only preserving one fully connected layer and 75% neurons on each layer of original Alexnet is adequate to represent objects when detecting on instance dataset. Besides, patch-level realism is sufficient for training region-proposal based object detectors. Our method, which incorporates network compression with data extension, provides a simple way to deploy an existing notable neural network model on mobile applications.

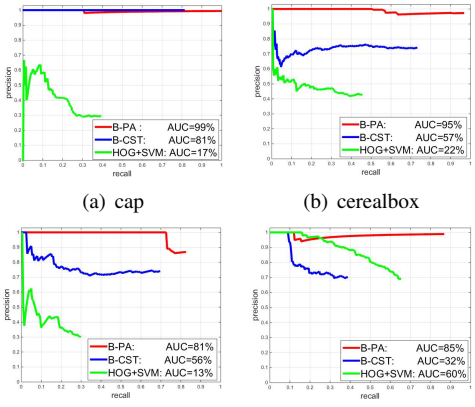
Acknowledgement

The authors thank the anonymous reviewers for their assistance. This work was supported by a grant from the National Natural Science Foundation of China(61673039).

Fig. 4. Detection samples of B-PA on two Dataset.Top: Detection on WRGB-D Dataset;Bottom :Detection on GMU Kitchen Dataset;Noticeable,in the detection on GMU Kitchen dataset,the upper line shows detection results with B-PA trained on real data; lower line shows detection with B-PA trained on both with realistic data and our extended data. The red bounding boxes represent false detection, and yellow boxes are correct detection.

TABLE I
COMPARATIVE RESULTS ON GMU KITCHEN DATASET

Detection Model	Pop secret	Mahatma rice	Red bull	Nature v2	Hunt sauce	Honey bunches	mAP
Real Data							
Faster RCNN [12]	86.4%	74.7%	54.6%	85.9%	81.8%	**91.9%**	78.0%
SSD [13]	64.8%	62.9%	27.7%	70.3%	64.5%	81.8%	59.1%
B-PA	80.2%	76.4%	30.5%	87.2%	80.4%	60.93%	64.8%
Real Data+ Extended Data							
Model:Faster RCNN [12] Data:SP-BL-SS [5]+real	**93.6%**	81.9 %	54.1%	88.6%	85.5%	91.4%	**82.5%**
Model:SSD [13] Data:SP-BL-SS [5]+real	85.2%	67.5%	37.6%	78.9%	74.2%	85.1%	71.4%
Model:B-PA Data:Extended data+Real	90.2%	**83.2%**	**54.9%**	**91.0%**	**87.7%**	76.3%	80.55%



(a) cap (b) cerealbox

(c) flashlight (d) sodacan

Fig. 5. Precision-recall curves comparing performance with traditional learning method: HOG+SVM [9], BING+ Color-Shape- Texture+ Cascade Classifier [11] and our proposed method B-PA on WRGB-D scene dataset.

REFERENCES

[1] F. Iandola, M. Moskewicz, and et al., "Squeezenet:alexnet-level accuracy with 50x fewer parameters and 1mb model size," *ICLR*, pp. 234–778, 2017.

[2] A. Krizhesky and G. H. I Sutskever, "Imagenet classification with deep convolutional neural networks." *International Conference on Neural Information Processing Systems.*, vol. 25, no. 2, 2012.

[3] C. Szegedy and et al., "Going deeper with convolutions." *Computer Vision and Pattern Recognition.*, pp. 1097–1105, 2014.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition." *Computer Vision and Pattern Recognition.*, 2015.

[5] G. Georgakis, A. Mousavian, and et al., "Synthesizing training data for object detection in indoor scenes." *arXiv:1702.07836*, 2017.

[6] M.-M. Cheng, Z. Zhang, and et al., "Bing: Binarized normed gradients for objectness estimation at 300fps." *Computer Vision and Pattern Recognition*, pp. 3286–3293, 2014.

[7] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Computer Science.*, 2015.

[8] G. Georgakis, A. M. Reza, and et al., "Multiview rgb-d dataset for object instance detection." *2016 Fourth International Conference in 3D vision*, pp. 426–434, 2016.

[9] K. Lai, L. Bo, and X. R. et al., "A large-scale hierarchical multi-view rgb-d object detection." *IEEE International Conference on Robotics and Automation*, vol. 47, pp. 1817–1824, 2011.

[10] A. Singh, J. Sha, and et al., "Bigbird: A large-scale 3d database of object instances." *In Robotics and Automation (ICRA), 2014 IEEE International Conference*, pp. 509–516, 2014.

[11] W. Rui and L. Ying, "Real-time 3d object detection in unstructured environments," *International Conference on Information and Systems*, 2016.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks." *NIPS*, vol. 39, no. 6, 2015.

[13] W. Liu, "Ssd: Single shot multibox detector.springer international publishing," *ECCV*, pp. 21–37, 2016.