Signal Processing: Image Communication 70 (2019) 145–156

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Object instance detection with pruned Alexnet and extended training data

Rui Wang[a], Jingwen Xu[a,*], Tony X. Han[b]

[a] Key Laboratory of Precision Opto-mechatronics Technology, Ministry of Education, School of Instrumentation Science and Opto-electronics Engineering, Beihang University, China
[b] Jingchi.ai, China

ARTICLE INFO

Keywords:
Object instance detection
Pruned Alexnet
Binarized normed gradient
Data extension

ABSTRACT

Object instance detection has garnered much concern in many practical applications, especially in the field of intelligent service robot. Imagine robots working in real scenes, one may expect the instance detection system to be light-weighted to enable mobile or embedded system deployment. Focusing on reconstructing a smaller learning network from a noted deep model,we have pruned Alexnet to a compressed model with fewer parameters but equivalent accuracy, denoted as BING-Pruned Alexnet(B-PA). Our method first utilizes BING(Binarized Normed Gradient) to compute bounding boxes, then builds a pruned network for recognition by reducing neurons and cutting fully connected layers on the classic architecture Alexnet. Since the training samples for instance detection are limited and of small variation, we extend the training data by combining data augmentation with synthetic generation. In the end, our B-PA network occupies only 5MB, which is 50 times smaller than the original Alexnet, but can still achieve Alexnet-level accuracy when recognizing on GMU Kitchen dataset. Numerical experiments are conducted to compare our algorithm with the state-of-art instance detection algorithms on a self-made BHID database and two public database i.e.,WRGB-D dataset and GMU Kitchen dataset, which demonstrate that B-PA reduces the storage requirements of neural networks substantially while preserving generalization performance on object instance detection.

1. 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 field of humanoid robotics. Imagining using an object detection system for everyday indoor environment like your home 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. Besides, how to detect instance objects 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, especially the existence of various state-of-the-art deep models which can extract robust features to adapt to complex detecting environments, 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, the deep neural network detection system need amount of annotations to train it. Too well known, to generate thousands of diverse training images with varied backgrounds and viewpoints is necessary for training a robust deep learning model. Traditionally, such a mammoth work for creating realistic training images with bounding box labels must be taken by vision researchers via realistic shooting images in random background. Nevertheless collecting and annotating scenes in such way is time-consuming and costly.

Recently, two successful research directions are catching much attention. One is the network model compression [1–10], which does the research to reduce computational cost and file volume of network models without loss of accuracy. Hence the compression network with smaller architecture can fit in applications for mobile and embedded system. For example, SqueezeNet model [3] matches AlexNet [11]-level accuracy on ImageNet with 50× fewer parameters. The GoogLeNet-v1 [10] model has only 53 MB of parameters, and it matches VGG [12]-level (533 MB) accuracy on ImageNet; Another one is data extension, which refers to automatically generating new annotated training samples by means of augmentation or synthesis. Data augmentation [13–15] such as color jittering, random scaling, shifting, and etc., are frequently-used schemes. As for synthesis, in order to reduce reliance on manual annotation, researchers either use synthetically rendered scenes and objects [16–18] or synthetically superimpose object masks into scene images [19,20]. As report goes [19,20], the approach of synthetically

[*] Corresponding author.
E-mail addresses: wangr@buaa.edu.cn (R. Wang), xujingwen9402@buaa.edu.cn (J. Xu), tony.han@jingchi.ai (T.X. Han).

https://doi.org/10.1016/j.image.2018.09.013
Received 20 March 2018; Received in revised form 2 August 2018; Accepted 27 September 2018
Available online xxxx

placing object masks in scenes can reduce the dependence on graphics renderings that ensures realism.

Inspired from these achievements in network model compression and data extension, an efficient network architecture called B-PA (BING [21] +Pruned Alexnet [11]) together with our training data extension strategy is put forward in this paper. First, we use the efficient region proposal method Binarized Normed Gradient(BING) to generate candidate regions. Then, we design a compressed classification network i.e. Pruned Alexnet to screen foreground regions and assign them to their categories. Finally, all the foreground bounding boxes with high confidence in each category are kept and clustered as the final detections. Like other CNN network, our B-PA depends on large annotated datasets, which are expensive to be acquired and processed, Therefore, an effective data extension strategy to generate training data is proposed. The "backbone" neuron network in our architecture is PA(Pruned Alexnet), which can achieve Alexnet level accuracy on object instance database with about 5 MB memory of size (50 times smaller than the original Alexnet [11]). Benefiting from the small model and abundant training data, our B-PA can achieve accurate detection results and consume small amounts of computing resources at the same time.

The novel method described in our paper contributes to the solution for detecting specific instance objects by the following means:

1. An compression network architecture B-PA, which reaches AlexNet-level accuracy on object instance detection task with 50× fewer parameters is introduced. This compression is carried out by preserving 75% neuron in each convolution layer and removing the first two fully connected layers in original Alexnet. Additionally, with the efficient region proposal technique BING [21], we are able to not only compute the bounding boxes but also narrow down the target search space, thus to reduce computation load further.

2. An effective training data extension strategy, which incorporates augmentation with the synthesis method of superimposing object masks into scene images, is employed in our work. It is proved to be a good solution to the contradiction that training a robust deep model depends on large annotated datasets while they are expensive to be acquired by manual annotation in realistic scene.

In the reminder of this paper, Section 2 discuss some related works on instances detection and network compression. Section 3 presents our approach in detail. Comparisons between our approach and the state-of-art ones are shown in Section 4 with some discussion and analysis, and concluding remarks are given in Section 5.

2. Related works

2.1. Object instance detection

The overarching goal of instance detection is to design a detecting system that can work well in unstructured environments and cope with complicated issues, such like noise and occlusion in real scenes. To address these problems, a fairly straightforward approach is to build a 3D model of each object and then fit the 3D model to the scene [22,23]. However, the process required to create these models is slow and often requires a specialized setup to obtain both RGB images and depth information. Other methods, which depend solely on 2D RGB images, can roughly be divided into two directions: template matching approach and learning based approach.

The basic template matching algorithm calculates a distortion function that measures the degree of similarity between the template and the sub-images at different positions of the image, then, locates the template into the image by taking the position of the maximum correlation or minimum distortion. It has received considerable attention in the research of template matching for designing robust templates against scaling, rotation and illumination changes, such as SIFT [24], SURF [25]. However, these approaches do not work well for objects which are not 'feature-rich'. Although DOT (Dominant Orientation Templates) based method [26] can operate on untextured objects, it cannot work with viewpoint changes. In addition, the limitations of template matching approach become apparent as the number of objects within the database increases. The size of the feature database can become quite large.

Learning-based method can effectively reduce the computation cost during detection and find the objects with heavy noise or occlusion through large amounts of offline learning.

Early learning based approaches depend heavily on extracting handcrafted features and using them to train an object classifier. For example, Alykhan Tejaniet et al. [27] proposed Latent-Class Hough Forests for specific object detection. They adapted the LINEMOD [28] feature into a scale-invariant patch descriptor and integrate it into a regression forest, which achieved good performance in noisy background scenes and occluded foregrounds. Another efficient object instance detection method based on handcrafted features is elaborated in [29]. The authors propose a cascading model with multiple features, including color histogram, HOG and LBP descriptors for object instance detection. The cascading framework is composed of three classifiers, which are used to exclude background regions level by level. Outstanding as they have performed, there is plenty of room for improvement. One of the standing problems is that the handcrafted features may not be applicable for all possible instances and that one such detection system targets only at one instance object.

In recent years, the popularity of handcrafted feature based learning method seems to be overtaken by the convolutional neural network (CNN), a hierarchical structure that has been shown to outperform handcrafted features in instance detection task. Besides, the deep learning model can target at multiple objects simultaneously, which significantly increase the working efficiency of detectors. In [19], they automatically superimpose 2D images of objects into images of real environments, and then utilize these synthetic images along with real images to train Faster RCNN network [30] and SSD network [31] to classify each image region into foreground/background. Their method benefits from the strong generalization ability of deep CNN and can deal with most background clutter. More recently, deep learning based approaches in computer vision are further being adopted for the task of detection and pose estimation of specific objects [32,33].

2.2. Model compression

Existing famous deep neural networks typically contain hundreds of millions of parameters and are both computationally and memory intensive. For example, AlexNet [11] (240 MB) has 60 million parameters, VGG-16 [12] (520 MB) has 130 million parameters. In 2015, ResNet [34] was formed by cascading a large number of residual modules which can overcome the over-fitting problem, making the depth of the model well trained at layer 152 and win the first prize of ILSVRC-2015. The success of these famous models reflects that recent researches in deep convolutional neural networks (CNNs) focus mainly on designing accurate and robust neural networks with more parameters to digest the millions of training image data. With the development of intelligent robot technology and mobile driving technology, there has been several papers focusing on compressing network without affecting their accuracy. These researchers believe that there typically exist multiple CNN architectures that achieve almost the same accuracy level for a given object detection task. Specially, smaller CNN architectures have advantages of less communication overhead to export new models to clients and more feasible FPGA or embedded deployment. Thus the problem of identifying a CNN architecture with fewer parameters but equivalent accuracy compared to a well-known model becomes another hot topic. Several methods have been proposed to address this issue. We can roughly divide these works into three types: designing compact layers, quantizing parameters and network pruning.

Designing compact layers aims at building compact blocks at each layer to address efficient training in deep neural networks. By discovering a suitable low-rank approximation of the parameters, Denton