Complexity and accuracy analysis of common artificial neural networks on pedestrian detection

Jiatu Wu[1]

[1]School of Data and Computer Science, Sun Yat-sen University (SYSU), Guangzhou, 510006, P. R. China

Abstract: With the development of computer version, deep learning and artificial neural networks approaches like SPP-net, Faster-RCNN and YOLO are proposed. This paper compares them in terms of efficiency and effectiveness. By analyzing the network architecture, SPP-net is more complex than Faster-RCNN and YOLO. By analyzing the experiments, SPP-net and Faster-RCNN are more accurate than YOLO in static detection while opposite in real-time system. Therefore, in real-time pedestrian detection situation, YOLO can perform better. In static pedestrian detection situation, Faster-RCNN or SPP-net can perform better.

1 INTRODUCTION

In recent years, deep learning and artificial neural networks were applied to computer version areas such as object detection, target segmentation, object recognition, etc. Studies in area of object detection has last a long period and particularly pedestrian detection was concentrated [1-4]. Pedestrian detection can be affected by several factors, such as different body shapes under different camera angles and different clothing of the same person.

In the object detection field, the traditional method uses a framework named sliding window. This framework resolves a specific picture into millions of sub windows of different sizes. It uses a classifier for each window to determine whether it contains a target object. Traditional methods aim at classifying different categories of objects and generally designing different features and classification algorithms. As for pedestrian detection, Histogram of Oriented Gradient (HOG) [5] method was proposed, adopting the linear Support Vector Machine (SVM) as a baseline classifier throughout the detection process. In 2012, Krizhevsky et al. [6] relighted people's attention in Convolutional Neural Networks (CNNs). CNN once became less prevalent in the 1990s. But Krizhevsky's work [6] showed significant improvement of the accuracy in object detection and classification on the dataset ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [7, 8]. When investigation goes further, a neural network architecture, Regions with CNN features (RCNN) [9] was proposed combining the object detection and classification. A series of neural network architecture was proposed by improving the accuracy and training speed, such as Spatial Pyramid Pooling in deep convolutional networks (SPP-net) [10] and faster-RCNN [11], which are mainly based on the main idea of RCNN. And the method, You Only Look Once (YOLO) provided a much more straightforward way for object detection using only one modified CNN [12]. All these methods or architectures have drawbacks, advantages and perform differently in different datasets or application scenarios. The RCNN based architectures (SPP-net, faster-RCNN) reached a higher mean Average Precision (mAP) of 78.3% and 70.0% in VOC07 dataset respectively but rather large scale of training time [10-12]. YOLO, on the contrary, was 10 mAP less accurate than them while FPS was much more accurate [12].

In this paper, we will compare the performances of SPP-net, faster-RCNN and YOLO in pedestrian detection. We will divide this into three main aspects, accuracy, training time and performances in real-time task. Section two describes related works. And following parts will discuss the comparison.

2 RELATED WORK

2.1. RCNN and SPP-net

RCNN uses region proposal method selective search [13] to propose independently candidate regions and extracts feature-vectors via CNN from each candidate regions. At the end of the CNN, a fully connected network is attached served as linear SVM to classify the object in the candidate regions [9]. Selective search sets initial regions using image segmentation based on graph and repeats greedy algorithm to group regions. Similarities between all neighbour regions are calculated and the most similar two regions are grouped. Similarities between the grouped region and neighbour regions are calculated [13]. Process of grouping similar regions runs until they reach one region covering all in an image. Drawback of RCNN is to take much time under slow object detection.

SPP-net can take input of various size by setting final layer as SPP layer that is a pooling layer using Bags of Word (BoW). SPP divides image into several bins as level

* Corresponding author: wujt9@mail2.sysu.edu.cn

of resolution, pools each bin as max value and makes feature vector of fixed size by assembling them [10].

2.2. Fast-RCNN and Faster-RCNN

Fast-RCNN takes image and objects and gets CNN feature maps of image. Then, Region of Interest (RoI) pooling layer extracts feature vector of fixed size about each object by sequencing parts of feature on feature maps through Fully Connected (FC) layer. Outputs are probability of estimation through Softmax and position of bounding box [15].

But Fast-RCNN also suffers a bottleneck of speed in processing region proposal. Faster-RCNN aims at handling this problem by bringing up a Region Proposal Network (RPN) [11]. In RPN, the proposed regions are fixed. The input picture is divided into $n \times n$ fixed regions and each of them is given 9 different ratio and scale proposals. The output of the network is to determine whether the proposal is background or foreground and a correction of the alignment position. The remaining part of Faster-RCNN is using the outputs of RPN to give a more meticulous classification and bonding boxes regression [11].

2.3. YOLO

Strictly speaking, YOLO is not a single net but rather a way of building networks for detection purposes. R. Joseph et al. use common CNN which is pre-trained on classification tasks and then add some fully-connected layers on top, which are trained on detection task. In the paper [12], R. Joseph et al. aim to predict bounding boxes for each object on an image and corresponding class labels for them. To accomplish that they divide image on regular 7×7 grids and for each grid cell they predict: object bounding boxes sizes and locations relative to the centre of the cell as well as probabilities scores per class. Loss function includes both b-boxes term and probabilistic term, so classification and detection tasks are being resolved simultaneously. Thanks to applying CNN to the whole image rather than to its parts, such nets make their predictions using global context with less mistakes. Also, these nets show relatively high performance in real-time systems.

3 COMPLEXITY ANALYSIS

3.1. SPP-net

SPP-net architecture uses selective search [13] to propose 2,000 candidate windows for an input image (noted as P) in any size $(w \times h)$. The main idea of selective search algorithm is assuming that there are n pre-segmented regions on the image, noted as $R = \{R_1, R_2, \cdots, R_n\}$, calculating the similarity between each region and its neighbor region (the adjacent region). Therefore, we will obtain a $n \times n$ similarity matrix (similarity between the same regions as well as between a region and the non-adjacent region can be set to NaN). Then, we will find the two regions corresponding to the maximum similarity value from the matrix and combine these two areas into one. Through this procedure, there are $n-1$ areas left in the image. And next, the above process will be repeated (only the new similarity between the new areas and its neighboring areas are needed to be calculated, and the others are not needed to be calculated again). The total number of regions will be reduced by one as the repeating increases for each time, knowing that all the last regions have been merged into the same region (that is, this process was performed $n-1$ times, and the total number of regions finally became 1).:

The input image P is also put into a ZF-5 [16] to extract the $256 \times n \times n$ feature maps [10, 17], where n depends on the size of the input image. Using the following equation, we can map the candidate regions of the original image to the feature map:

$$(x, y) = (S \times x', S \times y') \quad (1)$$

where (x', y') represents the coordinate point on the feature map, (x, y)represents the coordinate point on the original input image, and S is the product of all strides (including every pooling and convolution layer) in ZF-5 whose value is 16.

In each candidate region on the feature map, a four-level spatial pyramid ($1 \times 1, 2 \times 2, 3 \times 3, 6 \times 6$, totally 50 bins) to pool the features. This generates a 12,800-d (256×50) representation for each candidate region on the feature map.

3.2. Faster-RCNN

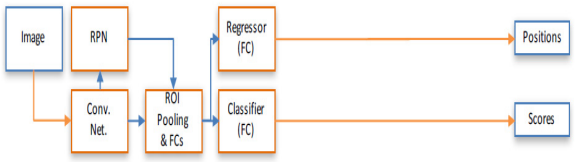

Fig. 1 The architecture of Faster-RCNN

The conventional Faster-RCNN is illustrated in Fig. 1. It is composed of 5 main parts: a deep fully convolutional network, region proposal network, ROI pooling and fully connected networks, bounding box regressor, and classifier. For consistency, the deep fully convolutional network is ZF-5. The input image P is also put into a ZF-5 [16] and extracted the $256 \times \mathrm{n} \times \mathrm{n}$ feature map [10], which is the input of RPN network and RoI pooling layer. In RPN network, for each point on feature map, there are k anchors (or candidate windows) with different scales and ratios and there will be $n \times n \times k$ candidate windows in total. In practice, $n \times n \times k$ candidate windows are ranked according to the score, the highest part is selected, and then 2,000 candidate windows are obtained through Non-Maximum Suppression [11] (noted that the complexity is $O(N^2/2)$). This is consistent with SPP-net mentioned above.

Using the candidate windows and the feature map, RoI pooling layer divides the varied size candidate windows into an $H \times W$ grid of sub-windows then max-pooling the values in each sub-window into the corresponding output grid cell. The complexity of this process is $O(1)$. The RoI layer is simply the special-case of the spatial pyramid

pooling layer used in SPP-net [15] in which there is only one pyramid level. Therefore, we obtain the uniform dimension feature vectors for the FC of regressor and classifier.

3.3. YOLO

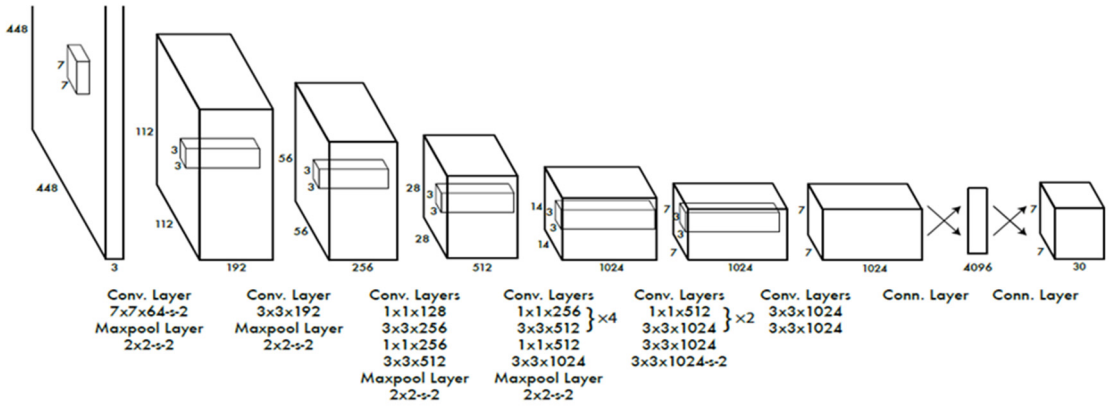

Fig. 2 YOLO architecture

YOLO architecture is inspired by GooLeNet model for image classification [18] as showed in Fig. 2. This network has 24 convolutional layers followed by 2 fully connected layers, which is almost 5 times larger than ZF-5 used in SPP-net and Faster-RCNN architecture.

First, the image is resized to $L \times L$ (448×448 in practice) size and is divided into $S \times S$ grids ($S = 7$, so each 64×64 sub-image belongs to one grid). Each grid proposes B bounding boxes ($B = 2$ in practice). Each bounding box corresponds to 4 coordinates $(x_{center}, y_{center}, w, h)$ and 1 confidence value. x_{center} and y_{center} is the center coordinate of the bounding box while w and h is weight and height of the bounding box respectively. The confidence value is computed via equation as follows:

$$\text{Confidence} = \Pr(\text{Object}) \times \text{IOU}_{\text{pre}}^{\text{truth}} \quad (2)$$

If there is a ground true box in the grid cell, $\Pr(\text{Object})$ is 1, otherwise it is 0, while $\text{IOU}_{\text{pre}}^{\text{truth}}$ is the IOU value between the predicted bounding box and the actual ground truth box.

In addition, each grid also generates C conditional class probabilities $\Pr(\text{Class}_i|\text{Object})$ ($C = 20$ in practice). In this way, an image eventually produces $S \times S \times (\text{B}\times5+\text{C})$ outputs (7*7*(2*5+20)=1,470 outputs in practice). At test time YOLO multiplies the conditional class probabilities and the individual box confidence predictions,

$$\Pr(\text{Class}_i|\text{Object}) \times \Pr(\text{Object}) \times \text{IOU}_{\text{pre}}^{\text{truth}} = \Pr(\text{Class}_i) \times \text{IOU}_{\text{pre}}^{\text{truth}} \quad (3)$$

which gives us class-specific confidence scores for each bounding box.

4 ACCURACY ANALYSIS

In this section, we use mAP to compare the detection accuracy of SPP-net, Faster-RCNN and YOLO. Under the Pascal VOC 2007 test data, the mAP of SPP-net (based on ZF-5) is 60.9% [10], which is slightly larger than Faster-RCNN (based on ZF-5)'s mAP (59.9%) [11]. Under the Pascal VOC 2012 test data, the mAP of Faster-RCNN is 70.4% [12], which is larger than YOLO's mAP (57.9%) [12]. But in real-time system on Pascal VOC 2007, YOLO gets a better accuracy and is faster than Faster-RCNN (63.4% mAP and 45 fps in YOLO and 62.1% mAP and 18 fps in Faster-RCNN).

Therefore, we can conclude that in detection accuracy, SPP-net performs better than Faster-RCNN and Faster-RCNN performs better than YOLO. But in real time situation, YOLO perform much more accurate and faster than Faster-RCNN and SPP-net.

5 CONCLUSION

In this paper, we discuss the complexity of SPP-net, Faster-RCNN, YOLO and conclude that under the same network architecture scale, SPP-net has a more complex calculating process than Faster-RCNN while Faster-RCNN is more complex than YOLO. When it comes to accuracy, SPP-net performs more accurately than Faster-RCNN and YOLO in static detection. But in real-time system, YOLO performs faster and more accurate than SPP-net and Faster-RCNN. There are a lot of investigations focus on Faster-RCNN and YOLO [19-22]. In political applications, pedestrian detection is mostly a real-time problem. Therefore, YOLO can get a better performance. But in the situation that plenty of objects in one image, Faster-RCNN can be better by sacrificing speed. Because Faster-RCNN can detect more accurately in complex scene. But with the development of hardware facilities, complexity is not the main concern and accuracy attracts the most attention in research.

REFERENCES

1. B.Y. Hyeon, K.K. Chang. A performance comparison of pedestrian detection using Faster RCNN and ACF. *IIAI-AAI* (2017).

2. D.M. Gavrila. Pedestrian detection from a moving vehicle. *ECCV* (2000).
3. R.H. Zhang, F.L. Li, X. Zhou, T.H. Jiang, F. You, J.M. Xu, S.Q. Yang. A pedestrian detection method under time-space data fusion based on laser and video information. Journal of Transportation Systems Engineering and Information Technology, v 15, n 3, p 49-55, June 1, 2015;
4. H.K. Kyaw. Hidden-layer ensemble fusion of MLP neural networks for pedestrian detection. Informatica. v 41, n 1, p 87-97, 2017
5. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*. (2005)
6. A. Krizhevsky, I. Sutskever, G. Hinton. ImageNet classification with deep convolutional neural networks. *NIPS* (2012).
7. J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, L. Fei-Fei. *ILSVRC* (2012).
8. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. *CVPR* (2009).
9. G. Ross, D. Jeff, D. Trevor, M. Jitendra. *CVPR* (2014).
10. K. He, X. Zhang, S. Ren, J. Sun. *ECCV* (2014).
11. R. S. Qing, H. K. Ming); G. Ross, S. Jian. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks (2015)
12. R. Joseph, D. Santosh, G. Ross, F. Ali. *CVPR* (2016).
13. J. Uijlings, K. Sande, T. Gevers, A. Smeulders. *IJCV* (2013).
14. R. Girshick, J. Donahue, T. Darrell, J. Malik. *TPAMI* (2016).
15. R. Girshick, *CVPR* (2015).
16. M. D. Zeiler, R. Fergus. *arXiv:1311.2901* (2013).
17. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel. *NC* (1989).
18. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich. *CoRR* (2014).
19. X.T. Zhao, W. Li, Y.F. Zhang, Gulliver T. Aaron, S. Chang, Z.Y. Feng. *VTC* (2017).
20. B. Hyeon, K.Chang, *IIAI* (2017).
21. R. Peiming, F. Wei, D. Soufiene. *ISCC* (2017).
22. P.Q. Wei, L. Wang, H.G. Yi, F. Min, X. Yuan, Y. Lei, H.X. Long, W. Xu, L.M. Xuan. *IHMSC* (2016).